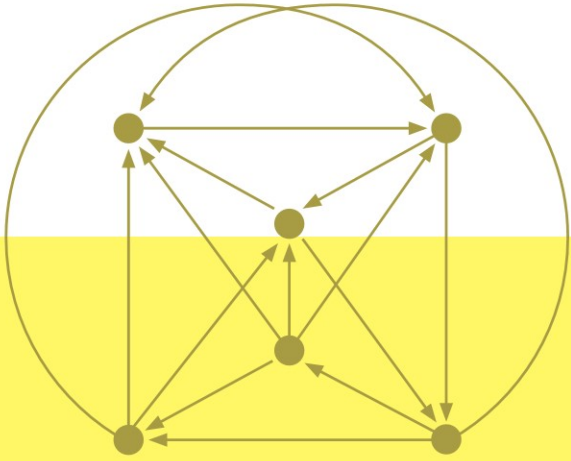
Jørgen Bang-Jensen
Gregory Z. Gutin

SPRINGER
MONOGRAPHS IN MATHEMATICS

Digraphs

Theory, Algorithms and Applications

Second Edition



Springer

S*pringer* **M***onographs* in **M***athematics*

For further volumes:
www.springer.com/series/3733

Jørgen Bang-Jensen • Gregory Z. Gutin

Digraphs

Theory, Algorithms and Applications

Second edition

Springer

Prof. Jørgen Bang-Jensen
University of Southern Denmark
Dept. Mathematics & Computer Science
Campusvej 55
5230 Odense
Denmark
jbj@imada.sdu.dk

Prof. Gregory Z. Gutin
Royal Holloway Univ. London
Dept. Computer Science
Egham Hill
Egham, Surrey
United Kingdom TW20 0EX
gutin@cs.rhul.ac.uk

ISSN 1439-7382
ISBN 978-1-84800-997-4 (hardcover) e-ISBN 978-1-84800-998-1
ISBN 978-0-85729-041-0 (softcover)
DOI 10.1007/978-1-84800-998-1
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2008939033

Mathematics Subject Classification (2000): 05C20, 05C38, 05C40, 05C45, 05C70, 05C85, 05C90, 05C99, 68R10, 68Q25, 68W05, 68W40, 90B06, 90B70, 90C35, 94C15

Cover design: deblik

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Lene and Irina

Preface to the Second Edition

The theory of graphs can be roughly partitioned into two branches: the areas of undirected graphs and directed graphs (digraphs). While there are many books on undirected graphs with new ones coming out regularly, the first edition of *Digraphs*, which was published in 2000, is the only modern book on graph theory covering more than a small fraction of the theory of directed graphs.

Since we wrote the first edition, the theory of directed graphs has continued to evolve at a high speed; many important results, including some of the conjectures from the first edition, have been proved and new methods were developed. Hence a new completely revised version became necessary. Instead of merely adding some new results and deleting a number of old ones, we took the opportunity to reorganize the book and increase the number of chapters from 12 to 18. This allows us to treat, in separate chapters, important topics such as branchings, feedback arc and vertex sets, connectivity augmentations, sparse subdigraphs with prescribed connectivity, applications, as well as packing, covering and decompositions of digraphs. We have added a large number of open problems to the second edition and the book now contains more than 150 open problems and conjectures, almost twice as many as the first edition. In order to avoid the book becoming unacceptably long, we had to remove a significant portion of material from the first edition. We have tried to do this as carefully as possible so that most of the information in the first edition is still available, along with a very large number of new results.

Even though this book should not be seen as an encyclopedia on directed graphs, we included as many important results as possible. The book contains a considerable number of proofs, illustrating various approaches and techniques used in digraph theory and algorithms.

One of the main features of this book is the strong emphasis on algorithms. This is something which is regrettably omitted in many books on graphs. Algorithms on (directed) graphs often play an important role in problems arising in several areas, including computer science and operations research. Secondly, many problems on (directed) graphs are inherently algorithmic. Hence, whenever possible we give constructive proofs of the results in the book. From these proofs one can very often extract an efficient algorithm for the problem studied. Even though we describe many algorithms, partly

due to space limitations, we do not supply all the details necessary in order to implement these algorithms. The latter (often highly nontrivial step) is a science in itself and we refer the reader to books on data structures and algorithms.

Another important feature is the large number of exercises which not only helps the reader to improve his or her understanding of the material, but also complements the results introduced in the text by covering even more material. Attempting these exercises will help the reader to master the subject and its main techniques.

Through its broad coverage and the exercises, stretching from easy to quite difficult, the book will be useful for courses on subjects such as (di)graph theory, combinatorial optimization and graph algorithms. Furthermore, it can be used for more focused courses on topics such as flows, cycles and connectivity. The book contains a large number of illustrations. This will help the reader to understand otherwise difficult concepts and proofs.

To facilitate the use of this book as a reference book and as a graduate textbook, we have added comprehensive symbol and subject indexes. It is our hope that the organization of the book, as well as detailed subject index, will help many readers to find what they are looking for without having to read through whole chapters. Due to our experience, we think that the book will be a useful teaching and reference resource for several decades to come.

Highlights

We cover the majority of important topics on digraphs ranging from quite elementary to very advanced ones. One of the main features of the second edition is the focus on open problems and the book contains more than 150 open problems or conjectures, thus making it a rich source for future research. By organizing the book so as to single out important areas, we hope to make it easy for the readers to find results and problems of their interest.

Below we give a brief outline of some of the main highlights of this book. Readers who are looking for more detailed information are advised to consult the list of contents or the subject index at the end of the book.

Chapter 1 contains most of the terminology and notation used in this book as well as several basic results. These are not only used frequently in other chapters, but also serve as illustrations of digraph concepts.

Chapter 2 is devoted to describing several important classes of directed graphs, such as line digraphs, the de Bruijn and Kautz digraphs, digraphs of bounded tree-width, digraphs of bounded directed widths, planar digraphs and generalizations of tournaments. We concentrate on characterization, recognition and decomposition of these classes. Many properties of these classes are studied in more detail in the rest of the book.

Chapters 3 and 4 cover distances and flows in networks. Although the basic concepts of these two topics are elementary, both theoretical and al-

gorithmic aspects of distances in digraphs as well as flows in networks are of great importance, due to their high applicability to other problems on digraphs and large number of practical applications, in particular, as a powerful modelling tool.

The main part of Chapter 3 is devoted to minimization and maximization of distance parameters in digraphs. In the self-contained Chapter 4, which may be used for a course on flows in networks, we cover basic topics on flows in networks, including a number of important applications to other (di)graph problems. Although there are several comprehensive books on flows, we believe that our fairly short and yet quite detailed account of the topic will give the majority of readers sufficient knowledge of the area. The reader who masters the techniques described in this chapter will be well equipped for solving many problems arising in practice.

Connectivity in (di)graphs is a very important topic. It contains numerous deep and beautiful results and has applications to other areas of graph theory and mathematics in general. It has various applications to other areas of research as well. We give a comprehensive account of connectivity topics and devote Chapters 5, 10, 12, 14 and parts of Chapter 11 to different aspects of connectivity.

Chapter 5 contains basic topics such as ear-decompositions, Menger's theorem, algorithms for determining the connectivity of a digraph as well as advanced topics such as properties of minimally k-(arc)-strong digraphs and critically k-strong digraphs.

Chapter 10 deals with problems concerning (arc-)disjoint linkings with prescribed initial and terminal vertices in digraphs. We prove that the 2-linkage problem is $\mathcal{NP}$-complete for arbitrary digraphs, but polynomially solvable for acyclic digraphs. Results on linkings in planar digraphs, eulerian digraphs as well as several generalizations of tournaments are discussed.

In Chapter 12 we study the problem of finding, in a k-(arc)-strong digraph, a small set of arcs (called a certificate) so that these arcs alone show that the digraph has the claimed connectivity. These problems are generally $\mathcal{NP}$-hard, so we give various approximation algorithms as well as polynomial algorithms for special classes of digraphs. We illustrate an application due to Cheriyan and Thurimella of Mader's results on minimally k-(arc)-strong digraphs to the problem of finding a small certificate for k-(arc)-strong connectivity. Finally, we also discuss recent results due to Gabow et al. on directed multigraphs.

In Chapter 14 we describe the splitting technique due to Mader and Lovász and illustrate its usefulness by giving an algorithm, due to Frank, for finding a minimum cardinality set of new arcs whose addition to a digraph D increases its arc-strong connectivity to a prescribed number. We also discuss a number of results related to increasing the connectivity by reversing arcs.

In Chapter 11 the famous theorem by Nash-Williams on orientations preserving a high degree of local arc-strong connectivity is described and the

weak version dealing with uniform arc-strong connectivities is proved using splitting techniques. We also discuss extensions of these results, including recent ones by Király and Szigeti. We give a proof a Jordán's result that every 18-connected graph has a 2-strong orientation. Submodular flows form a powerful generalization of circulations in networks. We introduce submodular flows and illustrate how to use this tool to obtain (algorithmic) proofs of many important results in graph theory. Finally we describe in detail an application, due to Frank, of submodular flows to the problem of orienting a mixed graph in order to maintain a prescribed degree of arc-strong connectivity.

In Chapter 6 we give a detailed account of results concerning the existence of hamiltonian paths and cycles in digraphs. Many results of this chapter deal with generalizations of tournaments. The reader will see that several of these much larger classes of digraphs share various nice properties with tournaments. In particular the hamiltonian path and cycle problems are polynomially solvable for most of these classes. The chapter illustrates various methods (such as the multi-insertion technique) for proving hamiltonicity.

In Chapter 7 we describe a number of interesting topics related to restricted hamiltonian paths and cycles. These include hamiltonian paths with prescribed end-vertices and orientations of hamiltonian paths and cycles in tournaments. We cover one of the main ingredients in the proof by Havet and Thomassé of Rosenfeld's conjecture on orientations of hamiltonian paths in tournaments.

Chapter 8 describes results on (generally) non-hamiltonian cycles in digraphs. We cover pancyclicity and the colour-coding technique by Alon, Yuster and Zwick and its application to yield polynomial algorithms for finding paths and cycles of 'logarithmic' length. We discuss the even cycle problem, including Thomassen's even cycle theorem. We also cover short cycles in multipartite tournaments, the girth of a digraph, chords of cycles and Ádám's conjecture. The chapter features various proof techniques including several algebraic, algorithmic, combinatorial and probabilistic methods.

Chapter 9 is devoted to branchings, a very important structure generalizing spanning trees in graphs. Branchings are not only of interest by themselves, they also play an important role in many proofs on digraphs. We prove Tutte's Matrix-Tree theorem on the number of distinct out-branchings in a digraph. We give a number of recent results on branchings with bounds on the degrees or extremal number of leaves. Edmonds' theorem on arc-disjoint branchings is proved and several applications of this important result are described. The problem of finding a minimum cost out-branching in a weighted digraph generalizes the minimum spanning tree problem. We describe algorithms for finding such a branching.

Chapter 13 covers a number of very important results related to packing, covering and decompositions of digraphs. We prove the Lucchesi-Younger theorem on arc-disjoint directed cuts, and give a number of results on arc-

disjoint hamiltonian paths and cycles. We discuss results on decomposing highly connected tournaments into many spanning strong subdigraphs, including a conjecture which generalizes the famous Kelly conjecture. We give a number of results on cycle factors with a prescribed number of cycles. Finally, we give a full proof, due to Bessy and Thomassé, of Gallai's conjecture on the minimum number of cycles needed to cover all vertices in a strong digraph.

Chapter 15 deals with another very important topic, namely, how to destroy all cycles in a digraph by removing as few vertices or arcs as possible. One of the main results in the chapter is that the feedback arc set problem is $\mathcal{NP}$-hard already for tournaments. This was conjectured by Bang-Jensen and Thomassen in 1992, but was proved only recently by three different sets of authors. Another major result is the proof by Chen, Liu, Lu, O'Sullivan and Razgon and that the feedback arc set problem and the feedback vertex set problems are both fixed-parameter tractable. We also include a scheme of a solution of Younger's conjecture, by Reed, Robertson, Seymour and Thomas, on the relation between the number of disjoint cycles and the size of a minimum feedback vertex set in a digraph.

Chapter 16 deals with edge-coloured graphs, a topic which has a strong relation to directed graphs. Alternating cycles in 2-edge-coloured graphs generalize the concept of cycles in digraphs. Certain results on cycles in bipartite digraphs, such as the characterization of hamiltonian bipartite tournaments, are special cases of results for edge-coloured complete graphs. There are useful implications in the other direction as well. In particular, using results on hamiltonian cycles in bipartite tournaments, we prove a characterization of those 2-edge-coloured complete graphs which have an alternating hamiltonian cycle. We briefly describe one important recent addition to the topic, i.e., a characterization by Feng, Giesen, Guo, Gutin, Jensen and Rafiey of edge-coloured complete multigraphs containing properly coloured Hamilton paths. This characterization was conjectured in the first edition.

Chapter 17 describes a number of different applications of directed and edge-coloured graphs. Through the topics treated we illustrate the diversity of the applications including some in quantum mechanics, bioinformatics, embedded computing and the traveling salesman problem.

Chapter 18 is included to make the book self-contained, by giving the reader a collection of the most relevant definitions and methods on algorithms and related areas.

Technical Remarks

We have tried to rank exercises according to their expected difficulty. Marks range from (−) to (++) in order of increasing difficulty. The majority of exercises have no mark, indicating that they are of moderate difficulty. An exercise marked (−) requires not much more than the understanding of the

main definitions and results. A (+) exercise requires a non-trivial idea, or involves substantial work. Finally, the few exercises which are ranked (++) require several deep ideas. Inevitably, this labelling is subjective and some readers may not agree with this ranking in certain cases. Some exercises have a header in boldface, which means that they cover an important or useful result not discussed in the text in detail.

We use the symbol □ to denote the end of a proof, or to indicate that either no proof will be given or the assertion is left as an exercise.

A few sections of the book require some basic knowledge of linear programming, while a few others require basic knowledge of probability theory.

We would be grateful to receive comments on the book. They may be sent to us by email to jbj@imada.sdu.dk or gutin@cs.rhul.ac.uk. We plan to have a web page containing information about misprints and other information about the book; see http://www.cs.rhul.ac.uk/books/dbook/

Acknowledgments

We wish to thank the following colleagues for helpful assistance and suggestions regarding various versions of the first edition: Noga Alon, Alex Berg, Thomas Böhme, Adrian Bondy, Jens Clausen, Samvel Darbinyan, Charles Delorme, Reinhard Diestel, Odile Favaron, Herbert Fleischner, András Frank, Yubao Guo, Vladimir Gurvich, Frédéric Havet, Jing Huang, Alice Hubenko, Bill Jackson, Tommy Jensen, Thor Johnson, Tibor Jordán, Ilia Krasikov, Hao Li, Martin Loebl, Gary MacGillivray, Wolfgang Mader, Crispin Nash-Williams, Jarik Nešetřil, Steven Noble, Erich Prisner, Gert Sabidussi, Lex Schrijver, Paul Seymour, Eng Guan Tay, Meike Tewes, Stéphan Thomassé, Carsten Thomassen, Bjarne Toft, Lutz Volkmann, Anders Yeo and Ke-Min Zhang.
For help on the second edition we wish to thank the following colleagues: Stephanne Bessy, Paulo Feofiloff, Frédéric Havet, Tibor Jordán, Eun Jung Kim, Paul Medvedev, Morten Hegner Nielsen, Anders Sune Pedersen, Zoltán Szigeti and Anders Yeo.

We are very grateful to Karen Borthwick of Springer-Verlag, London for her help and encouragement and to David Bahr who did excellent work copy-editing the book. We also thank the anonymous reviewers used by Springer for providing us with encouragement and very useful feedback.

Last, but most importantly, we wish to thank our families, in particular our wives Lene and Irina, without whose constant support we would never have succeeded in completing this project.

Odense, Denmark — Jørgen Bang-Jensen
London, UK — Gregory Gutin
September 2008

Contents

1. Basic Terminology, Notation and Results 1
1.1 Sets, Matrices and Vectors 1
1.2 Digraphs, Subdigraphs, Neighbours, Degrees 2
1.3 Isomorphism and Basic Operations on Digraphs 6
1.4 Walks, Trails, Paths, Cycles and Path-Cycle Subdigraphs ... 11
1.5 Strong and Unilateral Connectivity 15
1.6 Undirected Graphs, Biorientations and Orientations 18
1.7 Trees and Euler Trails in Digraphs 21
1.8 Mixed Graphs, Orientations of Digraphs, and Hypergraphs .. 24
1.9 Depth-First Search 26
1.10 Exercises 29

2. Classes of Digraphs 31
2.1 Acyclic Digraphs 32
2.2 Multipartite Digraphs and Extended Digraphs 34
2.3 Transitive Digraphs, Transitive Closures and Reductions 36
2.4 Line Digraphs 39
2.5 The de Bruijn and Kautz Digraphs 44
2.6 Series-Parallel Digraphs 47
2.7 Quasi-Transitive Digraphs 52
2.8 Path-Mergeable Digraphs 55
2.9 Locally In/Out-Semicomplete Digraphs 57
2.10 Locally Semicomplete Digraphs 59
2.10.1 Round Digraphs 60
2.10.2 Non-Strong Locally Semicomplete Digraphs 61
2.10.3 Strong Round Decomposable Locally Semicomplete Digraphs 63
2.10.4 Classification of Locally Semicomplete Digraphs 66
2.11 Totally Φ-Decomposable Digraphs 69
2.12 Planar Digraphs 71
2.13 Digraphs of Bounded Width 73
2.13.1 Digraphs of Bounded Tree-Width 74
2.13.2 Digraphs of Bounded Directed Widths 78
2.14 Other Families of Digraphs 80

2.14.1 Circulant Digraphs 80
2.14.2 Arc-Locally Semicomplete Digraphs 81
2.14.3 Intersection Digraphs 82
2.15 Exercises 84

3. Distances 87
3.1 Terminology and Notation on Distances 87
3.2 Structure of Shortest Paths 89
3.3 Algorithms for Finding Distances in Digraphs 91
3.3.1 Breadth-First Search (BFS) 92
3.3.2 Acyclic Digraphs 93
3.3.3 Dijkstra's Algorithm 94
3.3.4 The Bellman-Ford-Moore Algorithm 97
3.3.5 The Floyd-Warshall Algorithm 99
3.4 Inequalities on Diameter 100
3.5 Minimum Diameter of Orientations of Multigraphs 103
3.6 Minimum Diameter Orientations of Some Graphs and Digraphs 108
3.6.1 Generalizations of Tournaments 108
3.6.2 Extended Digraphs 111
3.6.3 Cartesian Products of Graphs 113
3.6.4 Chordal Graphs 114
3.7 Kings in Digraphs 115
3.7.1 2-Kings in Tournaments 115
3.7.2 Kings in Semicomplete Multipartite Digraphs 116
3.7.3 Kings in Generalizations of Tournaments 118
3.8 (k, l)-Kernels 119
3.8.1 Kernels 119
3.8.2 Quasi-Kernels 122
3.9 Exercises 123

4. Flows in Networks 127
4.1 Definitions and Basic Properties 127
4.1.1 Flows and Their Balance Vectors 128
4.1.2 The Residual Network 130
4.2 Reductions Among Different Flow Models 131
4.2.1 Eliminating Lower Bounds 131
4.2.2 Flows with One Source and One Sink 132
4.2.3 Circulations 133
4.2.4 Networks with Bounds and Costs on the Vertices 134
4.3 Flow Decompositions 136
4.4 Working with the Residual Network 137
4.5 The Maximum Flow Problem 140
4.5.1 The Ford-Fulkerson Algorithm 142
4.5.2 Maximum Flows and Linear Programming 145

4.6 Polynomial Algorithms for Finding a Maximum (s,t)-Flow . . 146
4.6.1 Augmenting Along Shortest Augmenting Paths 147
4.6.2 Maximal Flows in Layered Networks 148
4.6.3 The Push-Relabel Algorithm 149
4.7 Unit Capacity Networks and Simple Networks 153
4.7.1 Unit Capacity Networks 153
4.7.2 Simple Networks 155
4.8 Circulations and Feasible Flows 156
4.9 Minimum Value Feasible (s,t)-Flows 158
4.10 Minimum Cost Flows 160
4.10.1 Characterizing Minimum Cost Flows 162
4.10.2 Building up an Optimal Solution 166
4.10.3 The Assignment and the Transportation Problem . . . 169
4.11 Applications of Flows 170
4.11.1 Maximum Matchings in Bipartite Graphs 170
4.11.2 The Directed Chinese Postman Problem 174
4.11.3 Finding Subdigraphs with Prescribed Degrees 176
4.11.4 Path-Cycle Factors in Directed Multigraphs 177
4.12 Exercises 179

5. Connectivity of Digraphs 191
5.1 Additional Notation and Preliminaries 192
5.1.1 The Network Representation of a Directed Multigraph 194
5.2 Finding the Strong Components of a Digraph 195
5.3 Ear Decompositions 198
5.4 Menger's Theorem 201
5.5 Determining Arc- and Vertex-Strong Connectivity 204
5.6 Minimally k-(Arc)-Strong Directed Multigraphs 207
5.6.1 Minimally k-Arc-Strong Directed Multigraphs 207
5.6.2 Minimally k-Strong Digraphs 213
5.7 Critically k-Strong Digraphs 218
5.8 Connectivity Properties of Special Classes of Digraphs 220
5.9 Disjoint X-Paths in Digraphs 223
5.10 Exercises 223

6. Hamiltonian, Longest and Vertex-Cheapest Paths and Cycles 227
6.1 Complexity 228
6.2 Hamilton Paths and Cycles in Path-Mergeable Digraphs 230
6.3 Hamilton Paths and Cycles in Locally In-Semicomplete Digraphs 231
6.4 Hamilton Cycles and Paths in Degree-Constrained Digraphs . 233
6.4.1 Sufficient Conditions 233
6.4.2 The Multi-Insertion Technique 239

6.4.3 Proofs of Theorems 6.4.1 and 6.4.5 240
6.5 Longest Paths and Cycles in Degree-Constrained Oriented Graphs 243
6.6 Longest Paths and Cycles in Semicomplete Multipartite Digraphs 244
6.6.1 Basic Results 245
6.6.2 The Good Cycle Factor Theorem 247
6.6.3 Consequences of Lemma 6.6.12 250
6.6.4 Yeo's Irreducible Cycle Subdigraph Theorem and Its Applications 253
6.7 Hamilton Paths and Cycles in Quasi-Transitive Digraphs . . . 256
6.8 Vertex-Cheapest Paths and Cycles 260
6.8.1 Vertex-Cheapest Paths and Cycles in Quasi-Transitive Digraphs 260
6.8.2 Minimum Cost k-Path-Cycle Subdigraphs 261
6.8.3 Cheapest i-Path Subdigraphs in Quasi-Transitive Digraphs 263
6.8.4 Finding a Cheapest Cycle in a Quasi-Transitive Digraph 265
6.9 Hamilton Paths and Cycles in Various Classes of Digraphs . . 265
6.10 Exercises 271

7. Restricted Hamiltonian Paths and Cycles 275
7.1 Hamiltonian Paths with a Prescribed End-Vertex 275
7.2 Weakly Hamiltonian-Connected Digraphs 277
7.2.1 Results for Extended Tournaments 277
7.2.2 Results for Locally Semicomplete Digraphs 283
7.3 Hamiltonian-Connected Digraphs 286
7.4 Hamiltonian Cycles Containing or Avoiding Prescribed Arcs . 289
7.4.1 Hamiltonian Cycles Containing Prescribed Arcs 290
7.4.2 Avoiding Prescribed Arcs with a Hamiltonian Cycle . 292
7.4.3 Hamiltonian Cycles Avoiding Arcs in 2-Cycles 295
7.5 Arc-Traceable Digraphs 296
7.6 Oriented Hamiltonian Paths and Cycles 297
7.7 Exercises 303

8. Paths and Cycles of Prescribed Lengths 307
8.1 Pancyclicity of Digraphs 307
8.1.1 (Vertex-)Pancyclicity in Degree-Constrained Digraphs 308
8.1.2 Pancyclicity in Extended Semicomplete and Quasi-Transitive Digraphs 309
8.1.3 Pancyclic and Vertex-Pancyclic Locally Semicomplete Digraphs 312
8.1.4 Further Pancyclicity Results 315
8.1.5 Cycle Extendability in Digraphs 317

8.1.6 Arc-Pancyclicity . . . 318
8.2 Colour Coding: Efficient Algorithms for Paths and Cycles . . . 320
8.3 Cycles of Length k Modulo p . . . 324
8.3.1 Complexity of the Existence of Cycles of Length k Modulo p Problems . . . 324
8.3.2 Sufficient Conditions for the Existence of Cycles of Length k Modulo p . . . 326
8.4 Girth . . . 329
8.5 Short Cycles in Semicomplete Multipartite Digraphs . . . 332
8.6 Exercises . . . 336

9. Branchings . . . 339
9.1 Tutte's Matrix Tree Theorem . . . 339
9.2 Optimum Branchings . . . 342
9.2.1 Matroid Intersection Formulation . . . 343
9.2.2 A Simple Algorithm for Finding a Minimum Cost Out-Branching . . . 344
9.3 Arc-Disjoint Branchings . . . 345
9.4 Implications of Edmonds' Branching Theorem . . . 348
9.5 Out-Branchings with Degree Bounds . . . 351
9.6 Arc-Disjoint In- and Out-Branchings . . . 354
9.7 Out-Branchings with Extremal Number of Leaves . . . 358
9.7.1 Minimum Leaf Out-Branchings . . . 359
9.7.2 Maximum Leaf Out-Branchings . . . 361
9.8 The Source Location Problem . . . 363
9.9 Miscellaneous Topics . . . 365
9.9.1 Edge-Disjoint Mixed Branchings . . . 365
9.9.2 The Minimum Covering Out-Tree Problem . . . 366
9.9.3 Minimum Cost Arc-Disjoint Branchings with Bandwidth Constraints . . . 367
9.9.4 Out-Forests . . . 368
9.9.5 The Maximum Weight Out-Forest Problem . . . 368
9.9.6 Branchings and Edge-Disjoint Trees . . . 370
9.10 Exercises . . . 370

10. Linkages in Digraphs . . . 373
10.1 Additional Definitions and Preliminaries . . . 373
10.2 The Complexity of the k-Linkage Problem . . . 375
10.3 Sufficient Conditions for a Digraph to Be k-Linked . . . 379
10.4 The k-Linkage Problem for Acyclic Digraphs . . . 382
10.5 Linkages in (Generalizations of) Tournaments . . . 385
10.5.1 Sufficient Conditions in Terms of (Local-)Connectivity . . . 385
10.5.2 The 2-Linkage Problem for Semicomplete Digraphs . . . 389

10.5.3 The 2-Linkage Problem for Generalizations of Tournaments . . . 391
10.6 Linkages in Planar Digraphs . . . 394
10.7 Weak Linkages . . . 398
10.7.1 Weak Linkages in Acyclic Directed Multigraphs . . . 400
10.7.2 Weak Linkages in Eulerian Directed Multigraphs . . . 401
10.7.3 Weak Linkages in Tournaments and Generalizations of Tournaments . . . 407
10.8 Linkages in Digraphs with Large Minimum Out-Degree . . . 410
10.8.1 Subdivisions of Transitive Tournaments in Digraphs of Large Out-Degree . . . 411
10.9 Miscellaneous Topics . . . 412
10.9.1 Universal Arcs in 2-Cyclic Digraphs . . . 412
10.9.2 Integer Multicommodity Flows . . . 413
10.10 Exercises . . . 414

11. Orientations of Graphs and Digraphs . . . 417
11.1 Underlying Graphs of Various Classes of Digraphs . . . 417
11.1.1 Underlying Graphs of Transitive and Quasi-Transitive Digraphs . . . 418
11.1.2 Underlying Graphs of Locally Semicomplete Digraphs 421
11.1.3 Local Tournament Orientations of Proper Circular Arc Graphs . . . 423
11.1.4 Underlying Graphs of Locally In-Semicomplete Digraphs . . . 426
11.2 Orientations with No Even Cycles . . . 428
11.3 Colourings and Orientations of Graphs . . . 431
11.4 Orientations and Nowhere-Zero Integer Flows . . . 435
11.5 Orientations Achieving High Arc-Strong Connectivity . . . 441
11.5.1 k-Arc-Strong Orientations . . . 441
11.5.2 Well-Balanced and Best-Balanced Orientations . . . 443
11.5.3 Simultaneous Best-Balanced Orientations . . . 444
11.5.4 Best-Balanced Orientations of Eulerian Multigraphs . 445
11.6 k-Strong Orientations . . . 446
11.7 Orientations Respecting Degree Constraints . . . 448
11.7.1 Orientations with Prescribed Degree Sequences . . . 448
11.7.2 Restrictions on Subsets of Vertices . . . 452
11.8 Submodular Flows . . . 453
11.8.1 Submodular Flow Models . . . 454
11.8.2 Existence of Feasible Submodular Flows . . . 455
11.8.3 Minimum Cost Submodular Flows . . . 458
11.8.4 Applications of Submodular Flows . . . 459
11.9 Orientations of Mixed Multigraphs . . . 461
11.10 k-(Arc)-Strong Orientations of Digraphs . . . 466
11.11 Miscellaneous Topics . . . 470

11.11.1 Another Measure of Well-Balancedness 470
11.11.2 Orienting to Preserve Reachability for Prescribed Pairs 470
11.12 Exercises 472

12. Sparse Subdigraphs with Prescribed Connectivity 479
12.1 Minimum Strong Spanning Subdigraphs 480
12.1.1 Digraphs with High Minimum Degree 482
12.2 Polynomially Solvable Cases of the MSSS Problem 483
12.2.1 The MSSS Problem for Extended Semicomplete Digraphs 484
12.2.2 The MSSS Problem for Quasi-Transitive Digraphs . . . 485
12.3 Approximation Algorithms for the MSSS Problem 487
12.3.1 A Simple $\frac{7}{4}$-Approximation Algorithm 487
12.3.2 Better Approximation Algorithms 488
12.4 Small Certificates for k-(Arc)-Strong Connectivity 489
12.4.1 Small Certificates for k-Strong Connectivity 490
12.4.2 Small Certificates for k-Arc-Strong Connectivity 491
12.4.3 Certificates for Directed Multigraphs 494
12.5 Minimum Weight Strong Spanning Subdigraphs 497
12.6 Directed Steiner Problems 498
12.7 Miscellaneous Topics 501
12.7.1 The Directed Spanning Cactus Problem 501
12.7.2 An FTP Algorithm for the MSSS Problem 501
12.7.3 Minimum Cost Strong Subdigraphs 502
12.8 Exercises 503

13. Packings, Coverings and Decompositions 505
13.1 Packing Directed Cuts: The Lucchesi-Younger Theorem 505
13.2 Packing Dijoins: Woodall's Conjecture 511
13.3 Packing Cycles 512
13.4 Arc-Disjoint Hamiltonian Paths and Cycles 515
13.5 Path Factors 519
13.6 Cycle Factors with the Minimum Number of Cycles 521
13.7 Cycle Factors with a Fixed Number of Cycles 525
13.8 Cycle Subdigraphs Covering Specified Vertices 528
13.9 Proof of Gallai's Conjecture 529
13.10 Decomposing a Tournament into Strong Spanning Subdigraphs 536
13.11 The Directed Path-Partition Conjecture 542
13.12 Miscellaneous Topics 546
13.12.1 Maximum One-Way Cuts and Covering by One-Way Cuts 546
13.12.2 Acyclic Decompositions of Digraphs 548

13.12.3 Decomposing Tournaments into Strong Subtournaments . . . 548
13.12.4 Decomposing Digraphs under Degree Constraints . . . 549
13.13 Exercises . . . 550

14. Increasing Connectivity . . . 553
14.1 The Splitting Off Operation . . . 553
14.2 Increasing the Arc-Strong Connectivity Optimally . . . 557
14.3 Increasing the Vertex-Strong Connectivity Optimally . . . 562
14.3.1 One-Way Pairs . . . 563
14.3.2 Optimal k-Strong Augmentation . . . 565
14.3.3 Special Classes of Digraphs . . . 566
14.4 Arc Reversals and Vertex-Strong Connectivity . . . 568
14.5 Arc-Reversals and Arc-Strong Connectivity . . . 570
14.5.1 Determining $r_k^{deg}(D)$ Efficiently . . . 571
14.5.2 Reversals of Arcs to Achieve High Arc-Strong Connectivity in Tournaments . . . 572
14.6 Increasing Connectivity by Deorienting Arcs . . . 573
14.7 Miscellaneous Topics . . . 576
14.7.1 Increasing Arc-Strong Connectivity of a Bipartite Digraph . . . 576
14.7.2 Augmenting Arc-Strong Connectivity in Directed Hypergraphs . . . 577
14.7.3 Weighted Versions of Local Arc-Connectivity Problems . . . 578
14.8 Exercises . . . 580

15. Feedback Sets and Vertex Orderings . . . 583
15.1 Two Conjectures on Feedback Arc Sets . . . 584
15.2 Optimal Orderings in Tournaments . . . 585
15.3 Complexity of the Feedback Set Problems . . . 586
15.3.1 $\mathcal{NP}$-Completeness Results . . . 587
15.3.2 FAS for Planar Digraphs . . . 590
15.3.3 Approximation Algorithms . . . 591
15.3.4 Fixed-Parameter Tractability Results . . . 593
15.4 Disjoint Cycles Versus Feedback Sets . . . 596
15.4.1 Relations Between Parameters ν_i and τ_i . . . 596
15.4.2 Solution of Younger's Conjecture . . . 598
15.5 Optimal Orderings and Seymour's Second Neighbourhood Conjecture . . . 600
15.6 Ádám's Conjecture . . . 603
15.7 Exercises . . . 605

16. Generalizations of Digraphs: Edge-Coloured Multigraphs . 607
16.1 Terminology, Notation and Initial Observations 608
16.2 Properly Coloured Euler Trails 610
16.3 Properly Coloured Cycles 613
16.4 Gadget Graphs and Shortest PC Cycles and (s,t)-Paths 617
16.4.1 P-Gadgets 617
16.4.2 P-Gadget Graphs 618
16.5 Long PC Cycles and Paths 621
16.6 Connectivity of Edge-Coloured Multigraphs 622
16.7 Alternating Cycles in 2-Edge-Coloured Bipartite Multigraphs 625
16.8 Paths and Cycles in 2-Edge-Coloured Complete Multigraphs 628
16.9 PC Paths and Cycles in c-Edge-Coloured Complete Graphs, $c \geq 3$ 635
16.10 Exercises 640

17. Applications of Digraphs and Edge-Coloured Graphs 643
17.1 A Digraph Model in Quantum Mechanics 643
17.1.1 Lower Bound for $\mu(n)$ 644
17.1.2 Families of Sets and $\mu(n)$ 644
17.1.3 Upper Bounds for $\mu(n)$ 646
17.1.4 When $\mu(n) > f(n)$ 647
17.1.5 Mediated Digraphs in Quantum Mechanics 647
17.2 Embedded Computing and Convex Sets in Acyclic Digraphs . 649
17.2.1 Embedded Computing Systems and Convex Sets 649
17.2.2 Bounds for the Number of Convex Sets 650
17.2.3 Algorithms for Generating Convex and Connected Convex Sets 652
17.3 When Greedy-Like Algorithms Fail 655
17.3.1 Greedy Algorithm 656
17.3.2 Max-Regret Algorithms 659
17.4 Domination Analysis of ATSP Heuristics 660
17.4.1 ATSP Heuristics with Factorial Domination Numbers 662
17.4.2 Upper Bounds on Domination Numbers 664
17.5 Solving the 2-Satisfiability Problem 666
17.6 Alternating Hamilton Cycles in Genetics 670
17.6.1 Proof of Theorem 17.6.1 672
17.6.2 Proof of Theorem 17.6.2 673
17.7 Gaussian Elimination 674
17.8 Markov Chains 677
17.9 List Edge-Colourings 679
17.10 Digraph Models of Bartering 683
17.11 PERT/CPM in Project Scheduling 685
17.12 Finite Automata 687
17.13 Puzzles and Digraphs 689
17.14 Gossip Problems 690

17.15 Deadlocks of Computer Processes 692
17.16 Exercises 694

18. Algorithms and Their Complexity 695
18.1 Combinatorial Algorithms 696
18.2 $\mathcal{NP}$-Complete and $\mathcal{NP}$-Hard Problems 700
18.3 The Satisfiability Problem 702
18.4 Fixed-Parameter Tractability and Intractability 703
18.5 Exponential Algorithms 705
18.6 Approximation Algorithms 706
18.7 Heuristics and Metaheuristics 707
18.8 Matroids 711
18.8.1 The Dual of a Matroid 714
18.8.2 The Greedy Algorithm for Matroids 714
18.8.3 Independence Oracles 715
18.8.4 Union of Matroids 715
18.8.5 Intersection of Two Matroids 716
18.8.6 Intersections of Three or More Matroids 717
18.9 Exercises 718

References 721

Symbol Index 761

Author Index 767

Subject Index 777

1. Basic Terminology, Notation and Results

In this chapter we will provide most of the terminology and notation used in this book. Various examples, figures and results should help the reader to better understand the notions introduced in the chapter. The results covered in this chapter constitute a collection of simple yet important facts on digraphs. Most of our terminology and notation are standard. Therefore, some readers may proceed to other chapters after a quick look through this chapter (unfamiliar terminology and notation can be clarified later by consulting the indices supplied at the end of this book).

In Section 1.1 we provide basic terminology and notation on sets and matrices. Digraphs, directed pseudographs, subdigraphs, weighted directed pseudographs, neighbourhoods, semi-degrees and other basic concepts of directed graph theory are introduced in Section 1.2. Isomorphism and basic operations on digraphs are considered in Section 1.3. In Section 1.4, we introduce walks, trails, paths and cycles, and study some properties of tournaments and acyclic digraphs. Basic notions and results on strong and unilateral connectivity are considered in Section 1.5. Undirected graphs, biorientations and orientations of graphs are considered in Section 1.6. In Section 1.7, we give characterizations of eulerian directed multigraphs and digraphs with out-branchings (in-branchings). Mixed graphs, orientations of digraphs, and hypergraphs are discussed in Section 1.8. Finally, in Section 1.9 we will introduce a simple, yet very important, technique in algorithmic graph theory called depth-first search (DFS).

1.1 Sets, Matrices and Vectors

For the sets of real numbers, rational numbers and integers we will use $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{Z}$, respectively. Also, let $\mathbb{Z}_+ = \{z \in \mathbb{Z} : \ z > 0\}$ and $\mathbb{Z}_0 = \{z \in \mathbb{Z} : \ z \geq 0\}$. The sets $\mathbb{R}_+$, $\mathbb{R}_0$, $\mathbb{Q}_+$ and $\mathbb{Q}_0$ are defined similarly. For an integer n, $[n]$ will denote the set $\{1, 2, \ldots, n\}$.

The main aim of this section is to establish some notation and terminology on finite sets used in this book. We assume that the reader is familiar with the following basic operations for a pair A, B of sets: the **intersection** $A \cap B$, the **union** $A \cup B$ (if $A \cap B = \emptyset$, then we will often write $A + B$ instead of

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_1,

$A \cup B$) and the **difference** $A \backslash B$ (often denoted by $A - B$). Sets A and B are **disjoint** if $A \cap B = \emptyset$.

Often we will not distinguish between a single element set (singleton) $\{x\}$ and the element x itself. For example, we may write $A \cup b$ instead of $A \cup \{b\}$. The **Cartesian product** of sets $X_1, X_2, \ldots, X_p$ is $X_1 \times X_2 \times \ldots \times X_p = \{(x_1, x_2, \ldots, x_p) : \ x_i \in X_i, \ 1 \leq i \leq p\}$.

For sets A, B, $A \subseteq B$ means that A is a subset of B; $A \subset B$ stands for $A \subseteq B$ and $A \neq B$. A set B is a **proper subset** of a set A if $B \subset A$ and $B \neq \emptyset$. A collection $S_1, S_2, \ldots, S_t$ of (not necessarily non-empty) subsets of a set S is a **subpartition** of S if $S_i \cap S_j = \emptyset$ for all $1 \leq i \neq j \leq t$. A subpartition $S_1, S_2, \ldots, S_t$ is a **partition** of S if $\cup_{i=1}^{t} S_i = S$. We will often use the name **family** for a collection of sets. A family $\mathcal{F} = \{X_1, X_2, \ldots, X_n\}$ of sets is **covered** by a set S if $S \cap X_i \neq \emptyset$ for every $i \in [n]$. We say that S is a **cover** of $\mathcal{F}$. For a finite set X, the number of elements in X (i.e., its **cardinality**) is denoted by $|X|$. We also say that X is an **$|X|$-element set** (or just an **$|X|$-set**). A set S satisfying a property $\mathcal{P}$ is a **maximum** (**maximal**) set with property $\mathcal{P}$ if there is no set S' satisfying $\mathcal{P}$ and $|S'| > |S|$ $(S \subset S')$. Similarly, one can define minimum (minimal) sets satisfying a property $\mathcal{P}$.

In this book, we will also use **multisets** which, unlike sets, are allowed to have repeated (multiple) elements. The **cardinality** $|S|$ of a multiset M is the total number of elements in S (including repetitions). Often, we will use the words 'family' and 'collection' instead of 'multiset'.

For an $m \times n$ matrix $S = [s_{ij}]$ the **transposed matrix** (of S) is the $n \times m$ matrix $S^T = [t_{kl}]$ such that $t_{ji} = s_{ij}$ for every $i \in [m]$ and $j \in [n]$. Unless otherwise specified, the vectors that we use are column-vectors. The operation of transposition is used to obtain row-vectors.

1.2 Digraphs, Subdigraphs, Neighbours, Degrees

A **directed graph** (or just **digraph**) D consists of a non-empty finite set $V(D)$ of elements called **vertices** and a finite set $A(D)$ of ordered pairs of distinct vertices called **arcs**. We call $V(D)$ the **vertex set** and $A(D)$ the **arc set** of D. We will often write $D = (V, A)$ which means that V and A are the vertex set and arc set of D, respectively. The **order** (**size**) of D is the number of vertices (arcs) in D; the order of D will sometimes be denoted by $|D|$. For example, the digraph D in Figure 1.1 is of order and size 6; $V(D) = \{u, v, w, x, y, z\}$, $A(D) = \{(u, v), (u, w), (w, u), (z, u), (x, z), (y, z)\}$. Often the order (size, respectively) of the digraph under consideration is denoted by n (m, respectively).

For an arc (u, v) the first vertex u is its **tail** and the second vertex v is its **head**. We also say that the arc (u, v) **leaves** u and **enters** v. The head and tail of an arc are its **end-vertices**; we say that the end-vertices are **adjacent**,

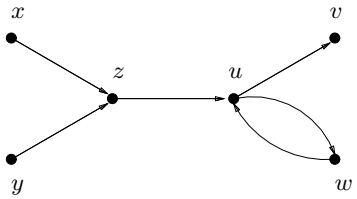


Figure 1.1 A digraph D.

i.e., u is adjacent **to**[1] v and v is adjacent to u. If (u, v) is an arc, we also say that u **dominates** v (v **is dominated by** u) and denote it by $u \rightarrow v$. We say that a vertex u is **incident** to an arc a if u is the head or tail of a. We will often denote an arc (x, y) by xy.

For a pair X, Y of vertex sets of a digraph D, we define

$$(X, Y)_D = \{xy \in A(D) : \ x \in X, y \in Y\},$$

i.e., $(X, Y)_D$ is the set of arcs with tail in X and head in Y. For example, for the digraph H in Figure 1.2, $(\{u, v\}, \{w, z\})_H = \{uw\}$, $(\{w, z\}, \{u, v\})_H = \{wv\}$ and $(\{u, v\}, \{u, v\})_H = \{uv, vu\}$.

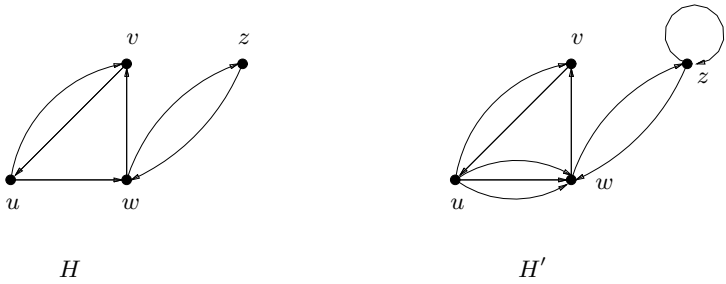


Figure 1.2 A digraph H and a directed pseudograph H'.

For disjoint subsets X and Y of $V(D)$, $X \rightarrow Y$ means that every vertex of X dominates every vertex of Y, $X \Rightarrow Y$ stands for $(Y, X)_D = \emptyset$ and $X \mapsto Y$ means that both $X \rightarrow Y$ and $X \Rightarrow Y$ hold. For example, in the digraph D of Figure 1.1, $u \rightarrow \{v, w\}$, $\{x, y, z\} \Rightarrow \{u, v, w\}$ and $\{x, y\} \mapsto z$.

The above definition of a digraph implies that we allow a digraph to have arcs with the same end-vertices (for example, uv and vu in the digraph H in Figure 1.2), but we do not allow it to contain **parallel** (also called **multiple**) arcs, that is, pairs of arcs with the same tail and the same head, or

[1] Some authors use the convention that x is adjacent to y to mean that there is an arc from x to y, rather than just that there is an arc xy **or** yx in D, as we will do in this book.

loops (i.e., arcs whose head and tail coincide). When parallel arcs and loops are admissible we speak of **directed pseudographs**; directed pseudographs without loops are **directed multigraphs**. In Figure 1.2 the directed pseudograph H' is obtained from H by appending a loop zz and two parallel arcs from u to w. Clearly, for a directed pseudograph D, $A(D)$ and $(X,Y)_D$ (for every pair X, Y of vertex sets of D) are multisets (parallel arcs provide repeated elements). We use the symbol $\mu_D(x,y)$ to denote the number of arcs from a vertex x to a vertex y in a directed pseudograph D. In particular, $\mu_D(x,y) = 0$ means that there is no arc from x to y.

We will sometimes give terminology and notation for digraphs only, but we will provide necessary remarks on their extension to directed pseudographs, unless this is trivial.

Below, unless otherwise specified, $D = (V, A)$ is a directed pseudograph. For a vertex v in D, we use the following notation:

$$N_D^+(v) = \{u \in V - v : \ vu \in A\}, \ N_D^-(v) = \{w \in V - v : \ wv \in A)\}.$$

The sets $N_D^+(v)$, $N_D^-(v)$ and $N_D(v) = N_D^+(v) \cup N_D^-(v)$ are called the **out-neighbourhood, in-neighbourhood** and **neighbourhood** of v. We call the vertices in $N_D^+(v)$, $N_D^-(v)$ and $N_D(v)$ the **out-neighbours**, **in-neighbours** and **neighbours** of v. In Figure 1.2, $N_H^+(u) = \{v, w\}$, $N_H^-(u) = \{v\}$, $N_H(u) = \{v, w\}$, $N_{H'}^+(w) = \{v, z\}$, $N_{H'}^-(w) = \{u, z\}$, $N_{H'}^+(z) = \{w\}$. For a set $W \subseteq V$, we let

$$N_D^+(W) = \bigcup_{w \in W} N_D^+(w) - W, \ N_D^-(W) = \bigcup_{w \in W} N_D^-(w) - W.$$

That is, $N_D^+(W)$ consists of those vertices from $V - W$ which are out-neighbours of at least one vertex from W. In Figure 1.2, $N_H^+(\{w, z\}) = \{v\}$ and $N_H^-(\{w, z\}) = \{u\}$.

For a set $W \subseteq V$, the **out-degree** of W (denoted by $d_D^+(W)$) is the number of arcs in D whose tails are in W and heads are in $V - W$, i.e., $d_D^+(W) = |(W, V - W)_D|$. The **in-degree** of W, $d_D^-(W) = |(V - W, W)_D|$. In particular, for a vertex v, the out-degree is the number of arcs, except for loops, with tail v. If D is a digraph (that is, it has no loops or multiple arcs), then the out-degree of a vertex equals the number of out-neighbours of this vertex . We call out-degree and in-degree of a set its **semi-degrees**. The **degree** of W is the sum of its semi-degrees, i.e., the number $d_D(W) = d_D^+(W) + d_D^-(W)$. For example, in Figure 1.2, $d_H^+(u) = 2, d_H^-(u) = 1, d_H(u) = 3, d_{H'}^+(w) = 2, d_{H'}^-(w) = 4, d_{H'}^+(z) = d_{H'}^-(z) = 1, d_H^+(\{u, v, w\}) = d_H^-(\{u, v, w\}) = 1$. Sometimes, it is useful to count loops in the semi-degrees: the **out-pseudodegree** of a vertex v of a directed pseudograph D is the number of all arcs with tail v. Similarly, one can define the **in-pseudodegree** of a vertex. In Figure 1.2, both in-pseudodegree and out-pseudodegree of z in H' are equal to 2. A vertex v of a directed pseudograph D is called a **sink** (**source**) if it does not have out-neighbours (in-neighbours).

The **minimum out-degree** (**minimum in-degree**) of D is

$$\delta^+(D) = \min\{d_D^+(x) : \ x \in V(D)\} \quad (\delta^-(D) = \min\{d_D^-(x) : \ x \in V(D)\}).$$

The **minimum semi-degree** of D is

$$\delta^0(D) = \min\{\delta^+(D), \delta^-(D)\}.$$

Similarly, one can define the **maximum out-degree** of D, $\Delta^+(D)$, and the **maximum in-degree**, $\Delta^-(D)$. The **maximum semi-degree** of D is

$$\Delta^0(D) = \max\{\Delta^+(D), \Delta^-(D)\}.$$

We say that D is **regular** if $\delta^0(D) = \Delta^0(D)$. In this case, D is also called **$\delta^0(D)$-regular**.

For degrees, semi-degrees as well as for other parameters and sets of digraphs, we will usually omit the subscript for the digraph when it is clear which digraph is meant.

Since the number of arcs in a directed multigraph equals the number of their tails (or their heads) we obtain the following very basic result.

Proposition 1.2.1 *For every directed multigraph D we have*

$$\sum_{x \in V(D)} d^-(x) = \sum_{x \in V(D)} d^+(x) = |A(D)|.$$

□

Clearly, this proposition is valid for directed pseudographs if in-degrees and out-degrees are replaced by in-pseudodegrees and out-pseudodegrees.

A digraph H is a **subdigraph** of a digraph D if $V(H) \subseteq V(D)$, $A(H) \subseteq A(D)$ and every arc in $A(H)$ has both end-vertices in $V(H)$. If $V(H) = V(D)$, we say that H is a **spanning subdigraph** (or a **factor**) of D. The digraph L with vertex set $\{u, v, w, z\}$ and arc set $\{uv, uw, wz\}$ is a spanning subdigraph of H in Figure 1.2. If every arc of $A(D)$ with both end-vertices in $V(H)$ is in $A(H)$, we say that H is **induced** by $X = V(H)$ (we write $H = D\langle X\rangle$) and call H an **induced** subdigraph of D. The digraph G with vertex set $\{u, v, w\}$ and arc set $\{uw, wv, vu\}$ is a subdigraph of the digraph H in Figure 1.2; G is neither a spanning subdigraph nor an induced subdigraph of H. The digraph G along with the arc uv is an induced subdigraph of H. For a subset $A' \subseteq A(D)$ the subdigraph **arc-induced** by A' is the digraph $D\langle A'\rangle = (V', A')$, where V' is the set of vertices in V which are incident with at least one arc from A'. For example, in Figure 1.2, $H\langle\{zw, uw\}\rangle$ has vertex set $\{u, w, z\}$ and arc set $\{zw, uw\}$. If H is a subdigraph of D, then we say that D is a **superdigraph** of H.

It is trivial to extend the above definitions of subdigraphs to directed pseudographs. To avoid lengthy terminology, we call the 'parts' of directed pseudographs just **subdigraphs** (instead of, say, directed subpseudographs).

For vertex-disjoint subdigraphs H, L of a digraph D, we will often use the shorthand notation $(H, L)_D$, $H \to L$, $H \Rightarrow L$ and $H \mapsto L$ instead of $(V(H), V(L))_D$, $V(H) \to V(L)$, $V(H) \Rightarrow V(L)$ and $V(H) \mapsto V(L)$.

A **weighted directed pseudograph** is a directed pseudograph D along with a mapping $c : A(D) \to \mathbb{R}$. Thus, a weighted directed pseudograph is a triple $D = (V(D), A(D), c)$. We will also consider **vertex-weighted directed pseudographs**, i.e., directed pseudographs D along with a mapping $c : V(D) \to \mathbb{R}$. (See Figure 1.3.) If a is an **element** (i.e., a vertex or an arc) of a weighted directed pseudograph $D = (V(D), A(D), c)$, then $c(a)$ is called the **weight** or the **cost** of a. An (unweighted) directed pseudograph can be viewed as a (vertex-)weighted directed pseudograph whose elements are all of weight 1. For a set B of arcs of a weighted directed pseudograph $D = (V, A, c)$, we define the weight of B as follows: $c(B) = \sum_{a \in B} c(a)$. Similarly, one can define the weight of a set of vertices in a vertex-weighted directed pseudograph. The **weight of a subdigraph** H of a weighted (vertex-weighted) directed pseudograph D is the sum of the weights of the arcs in H (vertices in H). For example, in the weighted directed pseudograph D in Figure 1.3 the set of arcs $\{xy, yz, zx\}$ has weight 9.5 (here we have assumed that we used the arc zx of weight 1). In the directed pseudograph H in Figure 1.3 the subdigraph $U = (\{u, x, z\}, \{xz, zu\})$ has weight 5.

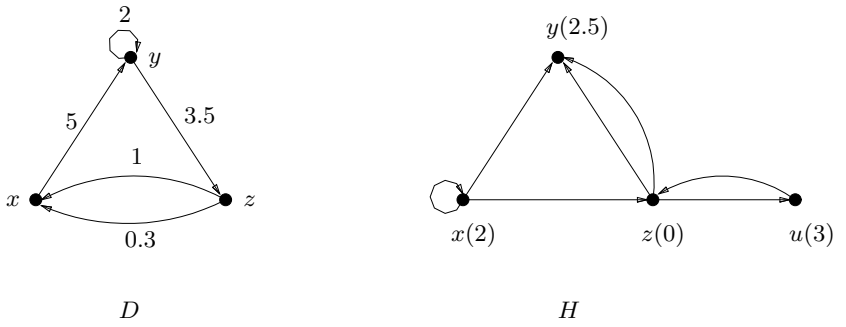


Figure 1.3 Weighted and vertex-weighted directed pseudographs (the vertex weights are given in brackets).

1.3 Isomorphism and Basic Operations on Digraphs

Suppose $D = (V, A)$ is a directed multigraph. A directed multigraph obtained from D by **deleting multiple arcs** is a digraph $H = (V, A')$ where $xy \in A'$ if and only if $\mu_D(x, y) \geq 1$. Let xy be an arc of D. By **reversing the arc** xy, we mean that we replace the arc xy by the arc yx. That is, in

the resulting directed multigraph D' we have $\mu_{D'}(x,y) = \mu_D(x,y) - 1$ and $\mu_{D'}(y,x) = \mu_D(y,x) + 1$.

A pair of (unweighted) directed pseudographs D and H are **isomorphic** (denoted by $D \cong H$) if there exists a bijection $\phi : V(D) \to V(H)$ such that $\mu_D(x,y) = \mu_H(\phi(x), \phi(y))$ for every ordered pair x, y of vertices in D. The mapping ϕ is an **isomorphism**. Quite often, we will not distinguish between isomorphic digraphs or directed pseudographs. For example, we may say that there is only one digraph on a single vertex and there are exactly three digraphs with two vertices. Also, there is only one digraph of order 2 and size 2, but there are two directed multigraphs and six directed pseudographs of order and size 2. For a set Ψ of directed pseudographs, we say that a directed pseudograph D **belongs** to Ψ or is a **member** of Ψ (denoted $D \in \Psi$) if D is isomorphic to a directed pseudograph in Ψ. Since we usually do not distinguish between isomorphic directed pseudographs, we will often write $D = H$ instead of $D \cong H$ for isomorphic D and H.

In case we do want to distinguish between isomorphic digraphs, we speak of **labeled digraphs**. In this case, a pair of digraphs D and H is indistinguishable if and only if they completely coincide (i.e., $V(D) = V(H)$ and $A(D) = A(H)$). In particular, there are four labeled digraphs with vertex set $\{1,2\}$. Indeed, the labeled digraphs $(\{1,2\}, \{(1,2)\})$ and $(\{1,2\}, \{(2,1)\})$ are distinct, even though they are isomorphic.

The **converse** of a directed multigraph D is the directed multigraph H which one obtains from D by reversing all arcs. It is easy to verify, using only the definitions of isomorphism and converse, that a pair of directed multigraphs are isomorphic if and only if their converses are isomorphic. To obtain subdigraphs, we use the following operations of **deletion**. For a directed multigraph D and a set $B \subseteq A(D)$, the directed multigraph $D - B$ is the spanning subdigraph of D with arc set $A(D) - B$. If $X \subseteq V(D)$, the directed multigraph $D - X$ is the subdigraph induced by $V(D) - X$, i.e., $D - X = D\langle V(D) - X\rangle$. For a subdigraph H of D, we define $D - H = D - V(H)$. Since we do not distinguish between a single element set $\{x\}$ and the element x itself, we will often write $D - x$ rather than $D - \{x\}$. If H is a non-induced subdigraph of D and $xy \in A(D) - A(H)$ with $x, y \in V(H)$, we can construct another subdigraph H' of D by adding the arc xy of H; $H' = H + xy$.

Let G be a subdigraph of a directed multigraph D. The **contraction** of G in D is a directed multigraph D/G with $V(D/G) = \{g\} \cup (V(D) - V(G))$, where g is a 'new' vertex not in D, and $\mu_{D/G}(x,y) = \mu_D(x,y)$, and for all distinct vertices $x, y \in V(D) - V(G)$ we have

$$\mu_{D/G}(x,g) = \sum_{v \in V(G)} \mu_D(x,v), \ \mu_{D/G}(g,y) = \sum_{v \in V(G)} \mu_D(v,y).$$

(Note that there is no loop in D/G.) Let $G_1, G_2, \ldots, G_t$ be vertex-disjoint subdigraphs of D. Then

$$D/\{G_1, G_2, \ldots, G_t\} = (\ldots((D/G_1)/G_2)\ldots)/G_t.$$

Clearly, the resulting directed multigraph $D/\{G_1, G_2, \ldots, G_t\}$ does not depend on the order of $G_1, G_2, \ldots, G_t$. Contraction can be defined for sets of vertices, rather than subdigraphs. It suffices to view a set of vertices X as a subdigraph with vertex set X and no arcs. Figure 1.4 depicts a digraph H and the contraction H/L, where L is the subdigraph of H induced by the vertices y and z.

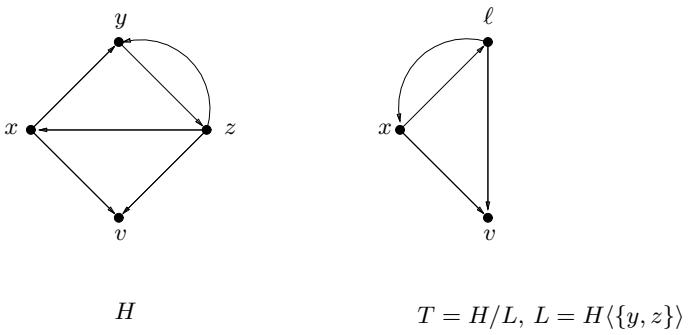


Figure 1.4 Contraction.

We will often use the following variation of the operation of contraction. This operation is called **path-contraction** and is defined as follows. Let P be an (x, y)-path in a directed multigraph $D = (V, A)$. Then $D//P$ stands for the directed multigraph with vertex set $V(D//P) = V \cup \{z\} - V(P)$, where $z \notin V$, and $\mu_{D//P}(uv) = \mu_D(uv)$, $\mu_{D//P}(uz) = \mu_D(ux)$, $\mu_{D//P}(zv) = \mu_D(yv)$ for all distinct $u, v \in V - V(P)$. In other words, $D//P$ is obtained from D by deleting all vertices of P and adding a new vertex z such that every arc with head x (tail y) and tail (head) in $V - V(P)$ becomes an arc with head (tail) z and the same tail (head). Observe that a path-contraction in a digraph results in a digraph (no parallel arcs arise). We will often consider path-contractions of paths of length one, i.e., arcs e. Clearly, a directed multigraph D has a k-cycle ($k \geq 3$) through an arc e if and only if $D//e$ has a cycle through z. Observe that the obvious analogue of path-contraction for undirected multigraphs does not have this nice property which is of use in this section. The difference between (ordinary) contraction (which is also called **set-contraction**) and path-contraction is reflected in Figure 1.5.

As for set-contraction, for vertex-disjoint paths $P_1, P_2, \ldots, P_t$ in D, the path-contraction $D//\{P_1, \ldots, P_t\}$ is defined as the directed multigraph $(\ldots((D//P_1)//P_2)\ldots)//P_t$; clearly, the result does not depend on the order of $P_1, P_2, \ldots, P_t$.

To construct 'bigger' digraphs from 'smaller' ones, we will often use the following operation called **composition**. Let D be a digraph with vertex set

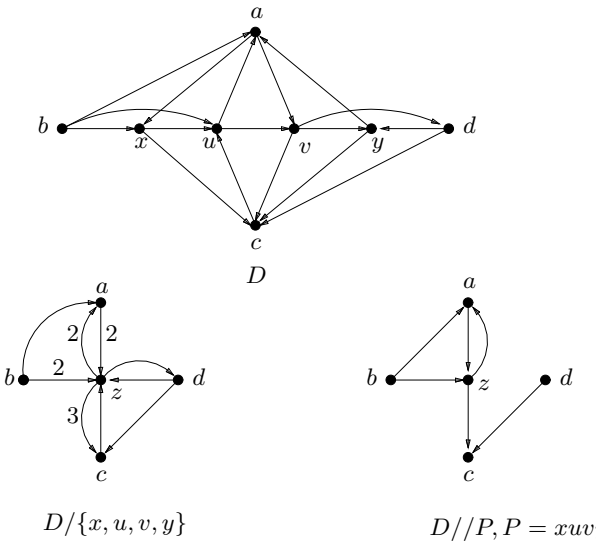


Figure 1.5 The two different kinds of contraction, set-contraction and path-contraction. The integers 2 and 3 indicate the number of corresponding parallel arcs.

$\{v_i : i \in [n]\}$, and let $G_1, G_2, \ldots, G_n$ be digraphs which are pairwise vertex-disjoint. The composition $D[G_1, G_2, \ldots, G_n]$ is the digraph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and arc set $(\cup_{i=1}^{n} A(G_i)) \cup \{g_i g_j : g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in A(D)\}$. Figure 1.6 shows the composition $T[G_x, G_l, G_v]$, where G_x consists of a pair of vertices and an arc between them, G_l has a single vertex, G_v consists of a pair of vertices and the pair of mutually opposite arcs between them, and the digraph T is from Figure 1.4.

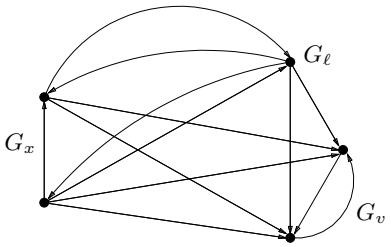


Figure 1.6 $T[G_x, G_\ell, G_v]$.

Let Φ be a set of digraphs. A digraph D is **Φ-decomposable** if D is a member of Φ or $D = H[S_1, \ldots, S_h]$ for some $H \in \Phi$ with $h = |V(H)| \geq 2$ and some choice of digraphs $S_1, S_2, \ldots, S_h$ (we call this decomposition a **Φ-**

decomposition). A digraph D is called **totally Φ-decomposable** if either $D \in \Phi$ or there is a Φ-decomposition $D = H[S_1, \ldots, S_h]$ such that $h \geq 2$, and each S_i is totally Φ-decomposable. In this case, if $D \notin \Phi$, a Φ-decomposition of D, Φ-decompositions $S_i = H_i[S_{i1}, \ldots, S_{ih_i}]$ of all S_i which are not in Φ, Φ-decompositions of those of S_{ij} which are not in Φ, and so on, form a sequence of decompositions which will be called a **total Φ-decomposition** of D. If $D \in \Phi$, we assume that the (unique) total Φ-decomposition of D consists of itself.

To illustrate the last paragraph of definitions, consider $\Psi = \{\overleftrightarrow{K}_1, \overleftrightarrow{K}_2, D_2\}$, where $\overleftrightarrow{K}_1$ is the digraph with a single vertex, $\overleftrightarrow{K}_2$ is the (complete) digraph with two vertices and two arcs, and D_2 has two vertices $\{1, 2\}$ and the arc $(1, 2)$. Construct the digraph D by deleting from the digraph in Figure 1.6 the pair of arcs going from G_ℓ to G_x. The digraph D is totally Ψ-decomposable. Indeed, $D = D_2[D_2, Q]$ is a Ψ-decomposition of D, where Q is the subdigraph of D induced by $V(G_\ell) \cup V(G_v)$. Moreover, $Q = D_2[\overleftrightarrow{K}_1, \overleftrightarrow{K}_2]$ is a Ψ-decomposition of Q. The above two decompositions form a total Φ-decomposition of D.

If $D = H[S_1, \ldots, S_h]$ and none of the digraphs $S_1, \ldots, S_h$ has an arc, then D is an **extension** of H. Distinct vertices x, y are **similar** if x, y have the same in- and out-neighbours in D. For every $i \in [h]$, the vertices of S_i are similar in D. For any set Φ of digraphs, Φ^{ext} denotes the (infinite) set of all extensions of digraphs in Φ, which are called **extended** Φ-digraphs. We say that Φ is **extension-closed** if $\Phi = \Phi^{ext}$.

The **Cartesian product** of a family of digraphs $D_1, D_2, \ldots, D_n$, denoted by $D_1 \times D_2 \times \ldots \times D_n$ or $\prod_{i=1}^{n} D_i$, where $n \geq 2$, is the digraph D having

$$\begin{aligned} V(D) &= V(D_1) \times V(D_2) \times \ldots \times V(D_n) \\ &= \{(w_1, w_2, \ldots, w_n) : \ w_i \in V(D_i), i \in [n]\} \end{aligned}$$

and a vertex $(u_1, u_2, \ldots, u_n)$ dominates a vertex $(v_1, v_2, \ldots, v_n)$ of D if and only if there exists an $r \in [n]$ such that $u_r v_r \in A(D_r)$ and $u_i = v_i$ for all $i \in [n] \setminus \{r\}$. (See Figure 1.7.)

The operation of **splitting** a vertex v of a directed multigraph D consists of replacing v by two (new) vertices u, w so that uw is an arc, all arcs of the form xv by arcs xu and all arcs of the form vy by wy. The **subdivision** of an arc uv of D consists of replacing uv by two arcs uw, wv, where w is a new vertex. If H can be obtained from D by subdividing one or more arcs (here we allow subdividing arcs that are already subdivided), then H is a **subdivision** of D. For a positive integer p and a digraph D, the **pth power** D^p of D is defined as follows: $V(D^p) = V(D)$, $x \to y$ in D^p if $x \neq y$ and there are $k \leq p - 1$ vertices $z_1, z_2, \ldots, z_k$ such that $x \to z_1 \to z_2 \to \ldots \to z_k \to y$ in D. According to this definition $D^1 = D$. For example, for the digraph $H_n = ([n], \{(i, i+1) : \ i \in [n-1]\})$, we have $H_n^2 = ([n], \{(i, j) : \ 1 \leq i < j \leq i + 2 \leq n\} \cup \{(n-1, n)\})$. See Figure 1.8 for the second power of a digraph.

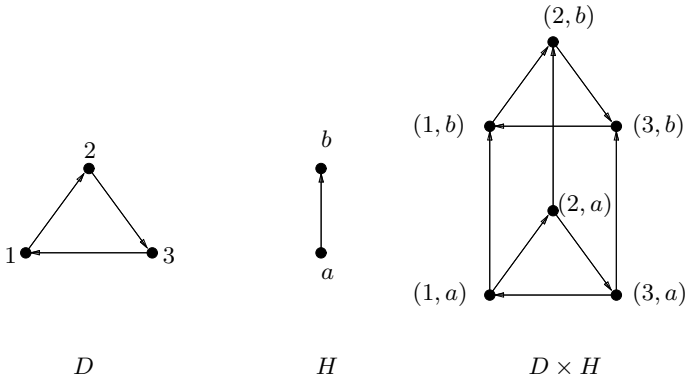


Figure 1.7 The Cartesian product of two digraphs.

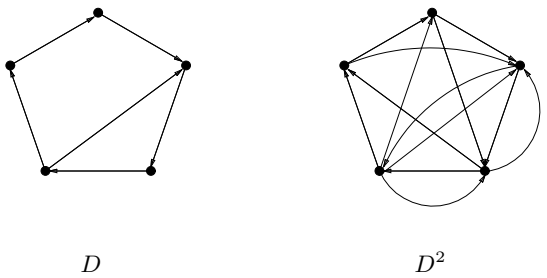


Figure 1.8 A digraph D and its second power D^2.

Let H and L be a pair of directed pseudographs. The **union** $H \cup L$ of H and L is the directed pseudograph D such that $V(D) = V(H) \cup V(L)$ and $\mu_D(x, y) = \mu_H(x, y) + \mu_L(x, y)$ for every pair x, y of vertices in $V(D)$. Here we assume that the function μ_H is naturally extended, i.e., $\mu_H(x, y) = 0$ if at least one of x, y is not in $V(H)$ (and similarly for μ_L). Figure 1.9 illustrates this definition.

1.4 Walks, Trails, Paths, Cycles and Path-Cycle Subdigraphs

In the following, D is always a directed pseudograph, unless otherwise specified. A **walk** in D is an alternating sequence $W = x_1 a_1 x_2 a_2 x_3 \dots x_{k-1} a_{k-1} x_k$ of vertices x_i and arcs a_j from D such that the tail of a_i is x_i and the head of a_i is x_{i+1} for every $i \in [k-1]$. A walk W is **closed** if $x_1 = x_k$, and **open** otherwise. The set of vertices $\{x_i : i \in [k]\}$ is denoted by $V(W)$; the set of arcs $\{a_j : j \in [k-1]\}$ is denoted by $A(W)$. We say that W is a walk **from** x_1 **to** x_k or an $\boldsymbol{(x_1, x_k)}$**-walk**. If W is open, then we say that the vertex x_1

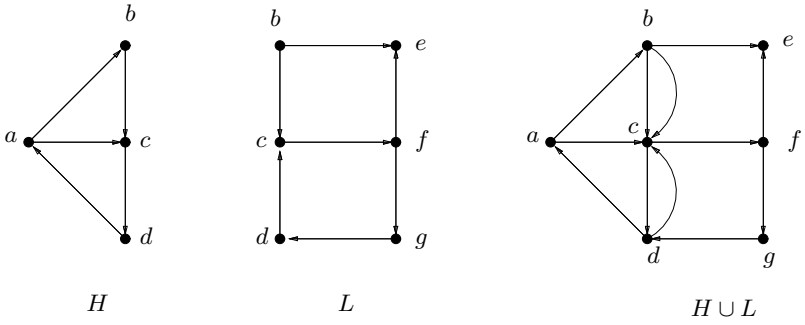


Figure 1.9 The union $D = H \cup L$ of the directed pseudographs H and L.

is the **initial** vertex of W, the vertex x_k is the **terminal** vertex of W, and x_1 and x_k are **end-vertices** of W. The **length** of a walk is the number of its arcs. Hence the walk W above has length $k-1$. A walk is **even** (**odd**) if its length is even (odd). When the arcs of W are defined from the context or simply unimportant, we will denote W by $x_1x_2\dots x_k$.

A **trail** is a walk in which all arcs are distinct. Sometimes, we identify a trail W with the *directed pseudograph* $(V(W), A(W))$, which is a *subdigraph* of D. If the vertices of W are distinct, W is a **path**. If the vertices $x_1, x_2, \dots, x_{k-1}$ are distinct, $k \geq 3$ and $x_1 = x_k$, W is a **cycle**. Since paths and cycles are special cases of walks, the **length** of a path and a cycle is already defined. The same remark is valid for other parameters and notions, e.g., an $\boldsymbol{(x,y)}$**-path**. A path P is an $\boldsymbol{[x,y]}$**-path** if P is a path between x and y, e.g., P is either an (x,y)-path or a (y,x)-path. A **longest** path (cycle) in D is a path (cycle) of maximum length in D.

When W is a cycle and x is a vertex of W, we say that W is a cycle **through** x. In a directed pseudograph D, a loop is also considered a cycle (of length one). A $\boldsymbol{k}$**-cycle** is a cycle of length k. The minimum integer k for which D has a k-cycle is the **girth** of D; denoted by $g(D)$. If D does not have a cycle, we define $g(D) = \infty$. If $g(D)$ is finite, then we call a cycle of length $g(D)$ a **shortest cycle** in D.

For subsets X, Y of $V(D)$, an (x,y)-path P is an $\boldsymbol{(X,Y)}$**-path** if $x \in X$, $y \in Y$ and $V(P) \cap (X \cup Y) = \{x, y\}$. Note that if $X \cap Y \neq \emptyset$, then a vertex $x \in X \cap Y$ forms an (X,Y)-path by itself. Sometimes we will talk about an (H, H')-path when H and H' are subdigraphs of D. By this we mean a $(V(H), V(H'))$-path in D.

For a cycle $C = x_1x_2\dots x_px_1$, the subscripts are considered modulo p, i.e., $x_s = x_i$ for every s and i such that $i \equiv s \bmod p$. As pointed out above (for trails), we will often view paths and cycles as subdigraphs. We can also consider paths and cycles as digraphs themselves. Let $\boldsymbol{\vec{P}_n}$ ($\boldsymbol{\vec{C}_n}$) denote a path (a cycle) with n vertices, i.e., $\vec{P}_n = ([n], \{(1,2),(2,3),\dots,(n-1,n)\})$ and $\vec{C}_n = \vec{P}_n + (n,1)$.

A walk (path, cycle) W is a **Hamilton** (or **hamiltonian**) walk (path, cycle) if $V(W) = V(D)$. A digraph D is **hamiltonian** if D contains a Hamilton cycle; D is **traceable** if D possesses a Hamilton path. A trail $W = x_1x_2 \dots x_k$ is an **Euler** (or **eulerian**) trail if $A(W) = A(D)$, $V(W) = V(D)$ and $x_1 = x_k$; a directed multigraph D is **eulerian** if it has an Euler trail.

To illustrate these definitions, consider Figure 1.10.

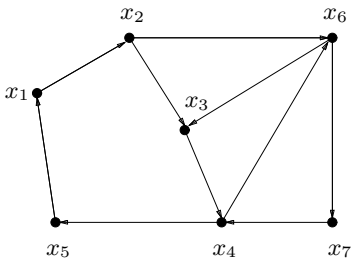


Figure 1.10 A directed graph H.

The walk $x_1x_2x_6x_3x_4x_6x_7x_4x_5x_1$ is a hamiltonian walk in D. The path $x_5x_1x_2x_3x_4x_6x_7$ is hamiltonian path in D. The path $x_1x_2x_3x_4x_6$ is an (x_1, x_6)-path and $x_2x_3x_4x_6x_3$ is an (x_2, x_3)-trail. The cycle $x_1x_2x_3x_4x_5x_1$ is a 5-cycle in D. The girth of D is 3 and the longest cycle in D has length 6.

Let $W = x_1x_2 \dots x_k$, $Q = y_1y_2 \dots y_t$ be a pair of walks in a digraph D. The walks W and Q are **disjoint** if $V(W) \cap V(Q) = \emptyset$ and **arc-disjoint** if $A(W) \cap A(Q) = \emptyset$. If W and Q are open walks, they are called **internally disjoint** if $\{x_2, x_3, \dots, x_{k-1}\} \cap V(Q) = \emptyset$ and $V(W) \cap \{y_2, y_3, \dots, y_{t-1}\} = \emptyset$.

We will use the following notation for a path or a cycle $W = x_1x_2 \dots x_k$ (recall that $x_1 = x_k$ if W is a cycle):

$$W[x_i, x_j] = x_ix_{i+1} \dots x_j.$$

It is easy to see that $W[x_i, x_j]$ is a path for $x_i \neq x_j$; we call it the **subpath** of W from x_i to x_j. If $1 < i \leq k$, then the **predecessor** of x_i on W is the vertex x_{i-1} and is also denoted by x_i^-. If $1 \leq i < k$, then the **successor** of x_i on W is the vertex x_{i+1} and is also denoted by x_i^+. Similarly, one can define $x_i^{++} = (x_i^+)^+$ and $x_i^{--} = (x_i^-)^-$, when these exist (which they always do if W is a cycle).

Also, for a set $X \subseteq V(W)$, we set $X^+ = \{x^+ : x \in X\}$, $X^- = \{x^- : x \in X\}$, $X^{++} = (X^+)^+$, etc. We will normally use such notation when a vertex x under consideration belongs to a unique walk W, otherwise W is given as a subscript, for example, x_W^+.

Proposition 1.4.1 *Let D be a digraph and let x, y be a pair of distinct vertices in D. If D has an (x, y)-walk W, then D contains an (x, y)-path P*

such that $A(P) \subseteq A(W)$. *If* D *has a closed* (x,x)*-walk* W, *then* D *contains a cycle* C *through* x *such that* $A(C) \subseteq A(W)$.

Proof: Consider a walk P from x to y of minimum length among all (x,y)-walks whose arcs belong to $A(W)$. We show that P is a path. Let $P = x_1x_2\dots x_k$, where $x = x_1$ and $y = x_k$. If $x_i = x_j$ for some $1 \leq i < j \leq k$, then the walk $P[x_1, x_i]P[x_{j+1}, x_k]$ is shorter than P; a contradiction. Thus, all vertices of P are distinct, so P is a path with $A(P) \subseteq A(W)$.

Let $W = z_1z_2\dots z_k$ be a walk from $x = z_1$ to itself ($x = z_k$). Since D has no loop, $z_{k-1} \neq z_k$. Let $y_1y_2\dots y_t$ be a shortest walk from $y_1 = z_1$ to $y_t = z_{k-1}$. We have proved above that $y_1y_2\dots y_t$ is a path. Thus, $y_1y_2\dots y_ty_1$ is a cycle through $y_1 = x$. □

An **oriented** graph is a digraph with no cycle of length two. A **tournament** is an oriented graph where every pair of distinct vertices are adjacent. In other words, a digraph T with vertex set $\{v_i : i \in [n]\}$ is a tournament if exactly one of the arcs v_iv_j and v_jv_i is in T for every $i \neq j \in [n]$. In Figure 1.11, one can see a pair of tournaments. It is an easy exercise to verify that each of them contains a Hamilton path. Actually, this is no coincidence by the following theorem of Rédei [768]. (In fact, Rédei proved a stronger result: every tournament contains an odd number of Hamilton paths.)

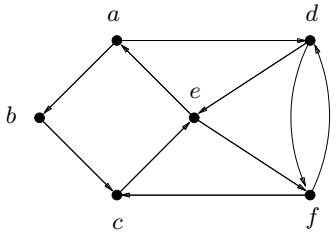
Figure 1.11 Tournaments.

Theorem 1.4.2 *Every tournament is traceable.*

Proof: Let T be a tournament with vertex set $\{v_i : i \in [n]\}$. Suppose that the vertices of T are labelled in such a way that the number of backward arcs, i.e., arcs of the form v_jv_i, $j > i$, is minimum. Then, $v_1v_2\dots v_n$ is a Hamilton path in T. Indeed, if this is not the case, there exists a subscript $i < n$ such that $v_iv_{i+1} \notin A(T)$. Thus, $v_{i+1}v_i \in A(T)$. However, in this case we can switch the vertices v_i and v_{i+1} in the labelling and decrease the number of backward arcs; a contradiction. □

A ***q*-path-cycle subdigraph** $\mathcal{F}$ of a digraph D is a collection of q paths $P_1,\dots, P_q$ and t cycles $C_1,\dots,C_t$ such that all of $P_1,\dots,P_q,C_1,\dots,C_t$ are pairwise disjoint (possibly, $q = 0$ or $t = 0$). We will denote $\mathcal{F}$ by $\mathcal{F} = P_1 \cup \dots \cup P_q \cup C_1 \cup \dots \cup C_t$ (the paths always being listed first). Quite

often, we will consider ***q*-path-cycle factors**, i.e., spanning q-path-cycle subdigraphs. If $t = 0$, $\mathcal{F}$ is a ***q*-path subdigraph** and it is a ***q*-path factor** (or just a **path-factor**) if it is spanning. If $q = 0$, we say that $\mathcal{F}$ is a ***t*-cycle subdigraph** (or just a **cycle subdigraph**) and it is a ***t*-cycle factor** (or just a **cycle factor**) if it is spanning. In Figure 1.12, $abc \cup defd$ is a 1-path-cycle factor, and $abcea \cup dfd$ is a cycle factor (or, more precisely, a 2-cycle factor).



H

Figure 1.12 A digraph H.

The **path covering number** $\mathrm{pc}(D)$ of D is the minimum positive integer k such that D contains a k-path factor. In particular, $\mathrm{pc}(D) = 1$ if and only if D is traceable. The **path-cycle covering number** $\mathrm{pcc}(D)$ of D is the minimum positive integer k such that D contains a k-path-cycle factor. Clearly, $\mathrm{pcc}(D) \leq \mathrm{pc}(D)$. The proof of the following simple yet helpful assertion on the path covering number is left as an easy exercise to the reader (Exercise 1.11).

Proposition 1.4.3 *Let D be a digraph, and let k be a positive integer. Then the following statements are equivalent:*

1. $\mathrm{pc}(D) = k$.
2. *There are $k-1$ (new) arcs $e_1, \ldots, e_{k-1}$ such that $D + \{e_1, \ldots, e_{k-1}\}$ is traceable, but there is no set of $k-2$ arcs with this property.*
3. *$k-1$ is the minimum integer s such that addition of s new vertices to D together with all possible arcs between $V(D)$ and these new vertices results in a traceable digraph.* □

1.5 Strong and Unilateral Connectivity

In a digraph D a vertex y is **reachable** from a vertex x if D has an (x, y)-walk. In particular, a vertex is reachable from itself. By Proposition 1.4.1, y is reachable from x if and only if D contains an (x, y)-path. A digraph D

is **strongly connected** (or, just, **strong**) if, for every pair x, y of distinct vertices in D, there exists an (x, y)-walk and a (y, x)-walk. In other words, D is strong if every vertex of D is reachable from every other vertex of D. We define a digraph with one vertex to be strongly connected. It is easy to see that D is strong if and only if it has a closed Hamilton walk (Exercise 1.29). As $\vec{C}_n$ is strong, every hamiltonian digraph is strong. The following basic result on tournaments is due to Moon [703]. A digraph D is **vertex-pancyclic** if for every $x \in V(D)$ and every integer $k \in \{3, 4, \ldots, n\}$, there exists a k-cycle through x in D.

Theorem 1.5.1 (Moon's theorem) *[703] Every strong tournament is vertex-pancyclic.*

Proof: Let x be a vertex in a strong tournament T on $n \geq 3$ vertices. The theorem is shown by induction on k. We first prove that T has a 3-cycle through x. Since T is strong, both $O = N^+(x)$ and $I = N^-(x)$ are non-empty. Moreover, (O, I) is non-empty; let $yz \in (O, I)$. Then, $xyzx$ is a 3-cycle through x. Let $C = x_0x_1 \ldots x_t$ be a cycle in T with $x = x_0 = x_t$ and $t \in \{3, 4, \ldots, n-1\}$. We prove that T has a $(t+1)$-cycle through x.

If there is a vertex $y \in V(T) - V(C)$ which dominates a vertex in C and is dominated by a vertex in C, then it is easy to see that there exists an index i such that $x_i \rightarrow y$ and $y \rightarrow x_{i+1}$. Therefore, $C[x_0, x_i]yC[x_{i+1}, x_t]$ is a $(t+1)$-cycle through x. Thus, we may assume that every vertex outside of C either dominates every vertex in C or is dominated by every vertex in C. The vertices from $V(T) - V(C)$ that dominate all vertices from $V(C)$ form a set R; the rest of the vertices in $V(T) - V(C)$ form a set S. Since T is strong, both S and R are non-empty and the set (S, R) is non-empty. Hence taking $sr \in (S, R)$ we see that $x_0srC[x_2, x_0]$ is a $(t+1)$-cycle through $x = x_0$. □

Corollary 1.5.2 (Camion's theorem) *[192] Every strong tournament is hamiltonian.* □

In fact, Moon's theorem can be extended to semicomplete digraphs. A digraph D is **semicomplete** if there is an arc between every pair of vertices in D. The next result follows from Moon's theorem due to Theorem 1.6.1.

Theorem 1.5.3 *Every strong semicomplete digraph is vertex-pancyclic.* □

A digraph D is **complete** if, for every pair x, y of distinct vertices of D, both xy and yx are in D. The complete digraph on n vertices will be denoted by $\overleftrightarrow{K}_n$. For a strong digraph $D = (V, A)$, a set $S \subset V$ is a **separator** (or a **separating set**) if $D - S$ is not strong. A digraph D is ***k*-strongly connected** (or ***k*-strong**) if $|V| \geq k + 1$ and D has no separator with less than k vertices. It follows from the definition of strong connectivity that a complete digraph with n vertices is $(n-1)$-strong, but is not n-strong.

The largest integer k such that D is k-strongly connected is the **vertex-strong connectivity** of D (denoted by $\kappa(D)$). If a digraph D is not strong, we set $\kappa(D) = 0$. For a pair s, t of distinct vertices of a digraph D, a set $S \subseteq V(D) - \{s, t\}$ is an **(s, t)-separator** if $D - S$ has no (s, t)-paths. For a strong digraph $D = (V, A)$, a set of arcs $W \subseteq A$ is a **cut** (or a **cutset**) if $D - A$ is not strong. A digraph D is **k-arc-strong** (or **k-arc-strongly connected**) if D has no cut with less than k arcs. The largest integer k such that D is k-arc-strongly connected is the **arc-strong connectivity** of D (denoted by $\lambda(D)$). If D is not strong, we set $\lambda(D) = 0$. Note that $\lambda(D) \geq k$ if and only if $d^+(X), d^-(X) \geq k$ for all proper subsets X of V.

A **strong component** of a digraph D is a maximal induced subdigraph of D which is strong. If $D_1, \ldots, D_t$ are the strong components of D, then clearly $V(D_1) \cup \ldots \cup V(D_t) = V(D)$ (recall that a digraph with only one vertex is strong). Moreover, we must have $V(D_i) \cap V(D_j) = \emptyset$ for every $i \neq j$ as otherwise all the vertices $V(D_i) \cup V(D_j)$ are reachable from each other, implying that the vertices of $V(D_i) \cup V(D_j)$ belong to the same strong component of D. We call $V(D_1) \cup \ldots \cup V(D_t)$ the **strong decomposition** of D. The **strong component digraph** $SC(D)$ of D is obtained by contracting strong components of D and deleting any parallel arcs obtained in this process. In other words, if $D_1, \ldots, D_t$ are the strong components of D, then $V(SC(D)) = \{v_i : i \in [t]\}$ and $A(SC(D)) = \{v_i v_j : (V(D_i), V(D_j))_D \neq \emptyset\}$. The subdigraph of D induced by the vertices of a cycle in D is strong, i.e., is contained in a strong component of D. Thus, $SC(D)$ is acyclic. By Proposition 2.1.3, the vertices of $SC(D)$ have an acyclic ordering. This implies that the strong components of D can be labelled $D_1, \ldots, D_t$ such that there is no arc from D_j to D_i unless $j < i$. We call such an ordering an **acyclic ordering** of the strong components of D . The strong components of D corresponding to the vertices of $SC(D)$ of in-degree (out-degree) zero are the **initial (terminal) strong components** of D. The remaining strong components of D are called **intermediate strong components** of D. Figure 1.13 shows a digraph D and its strong component digraph $SC(D)$.

It is easy to see that the strong component digraph of a tournament T is an acyclic tournament. Thus, there is a unique acyclic ordering of the strong components of T, namely, $T_1, \ldots, T_t$ such that $T_i \rightarrow T_j$ for every $i < j$. Clearly, every tournament has only one initial (terminal) strong component.

A digraph D is **unilateral** if, for every pair x, y of vertices of D, either x is reachable from y or y is reachable from x (or both). Clearly, every strong digraph is unilateral. A path $\vec{P}_n$ is unilateral; hence, being unilateral is a necessary condition for traceability of digraphs. The following is a characterization of unilateral digraphs.

Proposition 1.5.4 *A digraph D is unilateral if and only if there is a unique acyclic ordering $D_1, D_2, \ldots, D_t$ of the strong components of D and $(V(D_i), V(D_{i+1})) \neq \emptyset$ for every $i \in [t-1]$.*

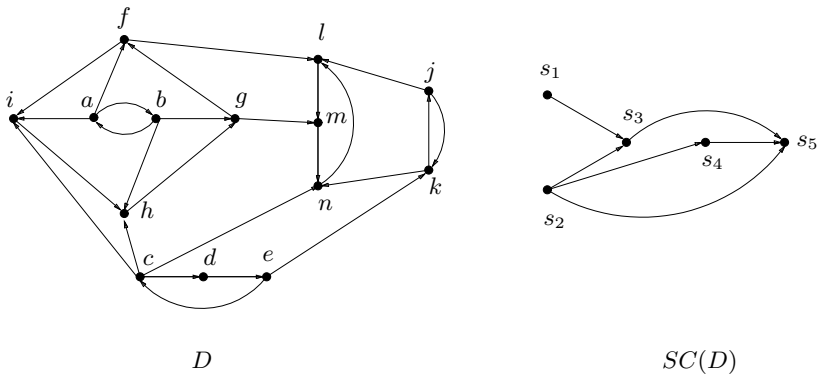


Figure 1.13 A digraph D and its strong component digraph $SC(D)$. The vertices s_1, s_2, s_3, s_4, s_5 are obtained by contracting the sets $\{a, b\}, \{c, d, e\}, \{f, g, h, i\}, \{j, k\}$ and $\{l, m, n\}$ which correspond to the strong components of D. The digraph D has two initial components, D_1, D_2 with $V(D_1) = \{a, b\}$ and $V(D_2) = \{c, d, e\}$. It has one terminal component D_5 with vertices $V(D_5) = \{l, m, n\}$ and two intermediate components D_3, D_4 with vertices $V(D_3) = \{f, g, h, i\}$ and $V(D_4) = \{j, k\}$.

Proof: The sufficiency is trivial. To see the necessity, observe that if $(V(D_i), V(D_{i+1})) = \emptyset$, then no vertex of $V(D_{i+1})$ is reachable from any vertex of $V(D_i)$ and vice versa. Finally, observe that if $(V(D_i), V(D_{i+1})) \neq \emptyset$ for every $i \in [t-1]$, then $D_1, D_2, \ldots, D_t$ is the unique acyclic ordering of the strong components of D, because $SC(D)$ has a hamiltonian path (see Exercise 2.1). □

1.6 Undirected Graphs, Biorientations and Orientations

An **undirected graph** (or a **graph**) $G = (V, E)$ consists of a non-empty finite set $V = V(G)$ of elements called **vertices** and a finite set $E = E(G)$ of unordered pairs of distinct vertices called **edges**. We call $V(G)$ the **vertex set** and $E(G)$ the **edge set** of G. In other words, an edge $\{x, y\}$ is a 2-element subset of $V(G)$. We will often denote $\{x, y\}$ just by xy. If $xy \in E(G)$, we say that the vertices x and y are **adjacent**. Notice that, in the above definition of a graph, we do not allow loops (i.e., pairs consisting of the same vertex) or parallel edges (i.e., multiple pairs with the same end-vertices). The **complement** $\overline{G}$ of a graph G is the graph with vertex set $V(G)$ in which two vertices are adjacent if and only if they are not adjacent in G.

When parallel edges and loops are admissible we speak of **pseudographs**; pseudographs with no loops are **multigraphs**. For a pair u, v of vertices in a pseudograph G, $\mu_G(u, v)$ denotes the number of edges between u and v. In particular, $\mu_G(u, u)$ is the number of loops at u.

A multigraph G is **complete** if every pair of distinct vertices in G are adjacent. We will denote the complete graph on n vertices (which is unique up to isomorphism) by K_n. Its complement $\overline{K}_n$ has no edge.

A multigraph H is ***p*-partite** if there exists a partition $V_1, V_2, \ldots, V_p$ of $V(H)$ into p **partite sets** (i.e., $V(H) = V_1 \cup \ldots \cup V_p$, $V_i \cap V_j = \emptyset$ for every $i \neq j$) such that every edge of H has its end-vertices in different partite sets. The special case of a p-partite graph when $p = 2$ is called a **bipartite graph**. We often denote a bipartite graph B by $B = (V_1, V_2; E)$. A p-partite multigraph H is **complete *p*-partite** if, for every pair $x \in V_i$, $y \in V_j$ $(i \neq j)$, an edge xy is in H. A complete graph on n vertices is clearly a complete n-partite graph for which every partite set is a singleton. We denote the complete p-partite *graph* with partite sets of cardinalities $n_1, n_2, \ldots, n_p$ by $K_{n_1,n_2,\ldots,n_p}$. Complete p-partite graphs for $p \geq 2$ are also called **complete multipartite graphs**.

To obtain short proofs of various results on subdigraphs of a directed multigraph $D = (V, A)$ the following transformation to the class of bipartite (undirected) multigraphs is extremely useful. Let $BG(D) = (V', V''; E)$ denote the bipartite multigraph with partite sets $V' = \{v' : v \in V\}$, $V'' = \{v'' : v \in V\}$ such that $\mu_{BG(D)}(u'w'') = \mu_D(uw)$ for every pair u, w of vertices in D. We call $BG(D)$ the **bipartite representation** of D; see Figure 1.14.

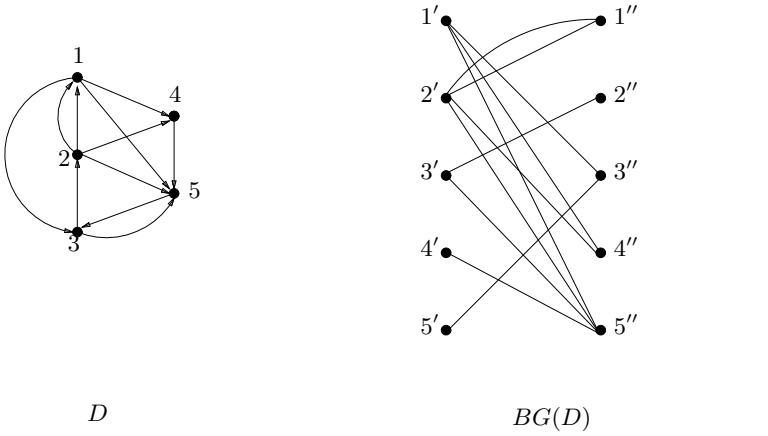


Figure 1.14 A directed multigraph and its bipartite representation.

For a pseudograph G, a directed pseudograph D is called a **biorientation** of G if D is obtained from G by replacing each edge $\{x, y\}$ of G by either xy or yx or the pair xy and yx (except for a loop xx which is replaced by a (directed) loop at x). Note that different copies of the edge xy in G may be replaced by different arcs in D. Thus if $\mu_G(x, y) = 3$, then we may replace one edge $\{x, y\}$ by the arc xy, another by the arc yx and the third by the pair

of arcs xy and yx. An **orientation** of a graph G is a biorientation of G which is an oriented graph (i.e., digraph having no 2-cycle and no loops). Clearly, every digraph is a biorientation and every oriented graph an orientation of some undirected graph. The **underlying graph** $UG(D)$ of a digraph D is the unique graph G such that D is a biorientation of G. The **underlying multigraph** $UMG(D)$ of a directed multigraph D is a multigraph obtained from D by replacing every arc xy with the edge $\{x, y\}$. For example, for a digraph H with vertices u, v and arcs uv, vu, $UG(H)$ has one edge and $UMG(H)$ has two parallel edges.

For a graph G, the **complete biorientation** of G (denoted by $\overleftrightarrow{G}$) is a biorientation D of G such that $xy \in A(D)$ implies $yx \in A(D)$. A digraph $D = (V, A)$ is **symmetric** if $xy \in A$ implies $yx \in A$. Clearly, D is symmetric if and only if D is the complete biorientation of some graph. An **oriented path (cycle)** is an orientation of a path (cycle).

A pseudograph G is **connected** if its complete biorientation $\overleftrightarrow{G}$ is strongly connected. Similarly, G is ***k*-connected** if $\overleftrightarrow{G}$ is k-strong. Strong components in $\overleftrightarrow{G}$ are **connected components**, or just **components** in G. A **bridge** in a connected pseudograph G is an edge whose deletion from G makes G disconnected. A pseudograph G is ***k*-edge-connected** if the graph obtained from G after deletion of at most $k-1$ edges is connected. Clearly, a connected pseudograph is bridgeless if and only if it is 2-edge-connected. The **neighbourhood** $N_G(x)$ of a vertex x in G is the set of vertices adjacent to x. The **degree** $d(x)$ of a vertex x is the number of edges except loops having x as an end-vertex. The **minimum (maximum) degree** of G is

$$\delta(G) = \min\{d(x) : \ x \in V(G)\} \ (\Delta(G) = \max\{d(x) : \ x \in V(G)\}).$$

We say that G is **regular** (or ***δ*(*G*)-regular**) if $\delta(G) = \Delta(G)$. A pair of graphs G and H is **isomorphic** if $\overleftrightarrow{G}$ and $\overleftrightarrow{H}$ are isomorphic.

A digraph is **connected** if its underlying graph is connected. The following well-known theorem is due to Robbins.

Theorem 1.6.1 (Robbins' theorem) *[780] A connected graph G has a strongly connected orientation if and only if G has no bridge.*

Proof: Clearly, if G has a bridge, G has no strong orientation. So assume that G is bridgeless. Then every edge uv is contained in some cycle (see Exercise 1.21). Choose a cycle C in G. Orient C as a directed cycle T_1. Suppose that $T_1, T_2, \ldots, T_k$ are chosen and oriented in such a way that every T_{i+1} $(1 \leq i < k)$ is either a cycle having only one vertex in common with $T^i = T_1 \cup T_2 \cup \ldots \cup T_i$ or a path with only initial and terminal vertices in common with T^i. If $UG(T^k) = G$, then we are done as a simple induction shows that T^k is strong. Otherwise, there is an edge xy which is not in $UG(T^k)$ such that x is in $UG(T^k)$. Let C be a cycle containing xy. Orient C to obtain a (directed) cycle Z. Let z be a vertex in $UG(T^k)$ which is first

encountered while traversing Z (after leaving x). Then, set $T_{k+1} = Z[x, z]$. The path (or cycle) T_{k+1} satisfies the above-mentioned properties. Since $E(G)$ is finite, after a certain number of iterations $\ell \leq m-1$ we have $UG(T^\ell) = G$. □

The notions of walks, trails, paths and cycles in undirected pseudographs are analogous to those for directed pseudographs (we merely disregard orientations). An ***xy*-path** in an undirected pseudograph is a path whose end-vertices are x and y. When we consider a digraph and its underlying graph $UG(D)$, we will often call walks of D **directed** (to distinguish between them and those in $UG(D)$). In particular, we will speak of directed paths, cycles and trails. An undirected graph is a **forest** if it has no cycle. A connected forest is a **tree**. It is easy to see (Exercise 1.24) that every connected undirected graph has a **spanning tree**, i.e., a spanning subgraph, which is a tree.

A **matching** M in a directed (an undirected) pseudograph G is a set of arcs (edges) with no common end-vertices. We also require that no element of M is a loop. If M is a matching, then we say that the edges (arcs) of M are **independent**. A matching M in G is **maximum** if M contains the maximum possible number of edges. A maximum matching is **perfect** if it has $n/2$ edges, where n is the order of G. A set Q of vertices in a directed or undirected pseudograph H is **independent** if the graph $H\langle Q\rangle$ has no edges (arcs). The **independence number**, $\alpha(H)$, of H is the maximum integer k such that H has an independent set of cardinality k. A **(proper) colouring** of a directed or undirected graph H is a partition of $V(H)$ into (disjoint) independent sets. The minimum number, $\chi(H)$, of independent sets in a proper colouring of H is the **chromatic number** of H.

In Section 1.3, the operation of composition of digraphs was introduced. Considering complete biorientations of undirected graphs, one can easily define the operation of **composition** of undirected graphs. Let H be a graph with vertex set $\{v_i : i \in [n]\}$, and let $G_1, G_2, \ldots, G_n$ be graphs which are pairwise vertex-disjoint. The composition $H[G_1, G_2, \ldots, G_n]$ is the graph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and edge set

$$\cup_{i=1}^{n} E(G_i) \cup \{g_i g_j : \ g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in E(H)\}.$$

If none of the graphs $G_1, \ldots, G_n$ in this definition of $H[G_1, \ldots, G_n]$ have edges, then $H[G_1, \ldots, G_n]$ is an **extension** of H.

1.7 Trees and Euler Trails in Digraphs

A digraph D is an **oriented forest** (**tree**) if D is an orientation of a forest (tree). A digraph T is an **out-tree** (an **in-tree**) if T is an oriented tree with just one vertex s of in-degree zero (out-degree zero). The vertex s is the **root** of T. If an out-tree (in-tree) T is a spanning subdigraph of D, T is called an **out-branching** (an **in-branching**). (See Figure 1.15.) We will often use the

notation B_s^+ (B_s^-) to denote an out-branching (in-branching) rooted at s in the digraph in question.

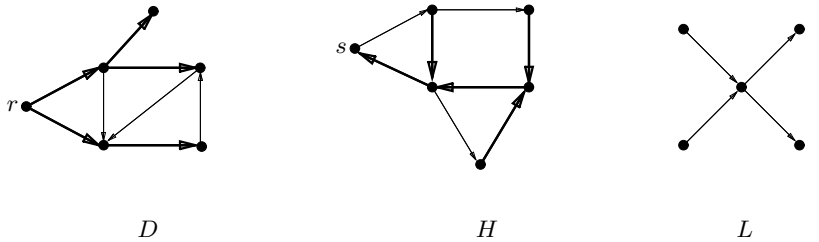


Figure 1.15 The digraph D has an out-branching with root r (shown in bold); H contains an in-branching with root s (shown in bold); L possesses neither an out-branching nor an in-branching.

Since each spanning oriented tree R of a connected digraph is acyclic as an undirected graph, R has at least one vertex of out-degree zero and at least one vertex of in-degree zero (see Proposition 2.1.1). Hence, the out-branchings and in-branchings capture the important cases of uniqueness of the corresponding vertices. The following is a characterization of digraphs with in-branchings (out-branchings).

Proposition 1.7.1 *A connected digraph D contains an out-branching (in-branching) if and only if D has only one initial (terminal) strong component.*

Proof: We prove this characterization only for out-branchings since the second claim follows from the first one by considering the converse of D.

Assume that D contains at least two initial strong components and suppose that D has an out-branching T. Observe that the root r of T is an initial strong component of D. Let x be a vertex in another initial strong component of D. Since r is the root of T, there is a path from r to x in T and, thus, in D, which is a contradiction to the assumption that r and x are in different initial strong components of D.

Now we assume that D contains only one initial strong component D_1, and r is an arbitrary vertex of D_1. We prove that D has an out-branching rooted at r. In $SC(D)$, the vertex x corresponding to D_1 is the only vertex of in-degree zero and, hence, by Proposition 2.1.2, every vertex of $SC(D)$ is reachable from x. Thus, every vertex of D is reachable from r. We construct an oriented tree T as follows. In the first step T consists of r. In Step $i \geq 2$, for every vertex y appended to T in the previous step, we add to T a vertex z, such that $y \rightarrow z$ and $z \notin V(T)$, together with the arc yz. We stop when no vertex can be included in T. Since every vertex of D is reachable from r, T is spanning. Clearly, r is the only vertex of in-degree zero in T. Hence, T is an out-branching. □

The oriented tree T constructed in the proof of Proposition 1.7.1 is a so-called BFS tree of D (see Chapter 3).

We formulate and prove the following well-known characterization of eulerian directed multigraphs (clearly, the deletion of loops in a directed pseudograph D does not change the property of D of being eulerian or otherwise). The 'undirected' version of this theorem marks the beginning of graph theory [303] (see the book [320] by Fleischner for a reprint of Euler's original paper and a translation into English, and see the book [160] by Biggs, Lloyd and Wilson or Wilson's paper [904] for a discussion of the historical record).

Theorem 1.7.2 (Euler's theorem[2]) *A directed multigraph D is eulerian if and only if D is connected and $d^+(x) = d^-(x)$ for every vertex x in D.*

Proof: Clearly, both conditions are necessary. We give a constructive proof of sufficiency by building an Euler trail T. Let T be initially empty and we may assume that D has at least two vertices. Choose an arbitrary vertex x in D. Since D is connected, there is a vertex $y \in N^+(x)$. Append x to T as well as an arc from x to y. Since $d^-(y) = d^+(y)$, there is an arc yz with tail y. Add both y and yz to T. We proceed similarly: after an arc uv is included in T, we append v to T together with an arc $a \notin T$ whose tail is v. Due to the condition $d^+(w) = d^-(w)$ for every vertex w, the above process can terminate only if the last arc appended to T is an arc whose head is the vertex x and the arcs of D with tail x are already in T. If all arcs of D are in T, we are done. Assume it is not so. Since D is connected, this means that T contains a vertex p which is a tail of an arc pq not in T. 'Shift' cyclically the vertices and arcs of T such that T starts and terminates at p. Add the arc pq to T and apply the process described above. This can terminate only when the last appended arc's tail is p and all arcs leaving p are already in T. Again, either we are done (all arcs are already in T) or we can find a new vertex to restart the above process. Since $V(D)$ is finite, after several steps all arcs of D will be included in T. □

The algorithm described in this proof can be implemented to run in $O(|V(D)| + |A(D)|)$ time (see Exercise 18.3). A generalization of the last theorem is given in Theorem 16.2.1. For eulerian directed multigraphs, the following stronger condition on out-degrees and in-degrees holds.

Corollary 1.7.3 *Let D be an eulerian directed multigraph and let $\emptyset \neq W \subset V(D)$. Then, $d^+(W) = d^-(W)$.*

Proof: Observe that

$$\sum_{w \in W} d^+(w) = |(W, W)| + d^+(W), \quad \sum_{w \in W} d^-(w) = |(W, W)| + d^-(W). \quad (1.1)$$

[2] Euler's original paper [303] only dealt with undirected graphs, but it is easy to see that the directed case generalizes the undirected case (see also Exercise 1.27).

By Theorem 1.7.2, $\sum_{w\in W} d^+(w) = \sum_{w\in W} d^-(w)$. The corollary follows from this equality and (1.1). □

1.8 Mixed Graphs, Orientations of Digraphs, and Hypergraphs

Mixed graphs are useful by themselves as a common generalization of undirected and directed graphs. Moreover, mixed graphs are helpful in several proofs on biorientations of graphs.

A **mixed graph** $M = (V, A, E)$ contains both arcs (ordered pairs of vertices in A) and edges (unordered pairs of vertices in E). We do not allow loops or parallel arcs and edges, but M may have an edge and an arc with the same end-vertices. For simplicity, both edges and arcs of a mixed graph are called **edges**. Thus, an arc is viewed as an oriented edge (and an unoriented edge as an edge in the usual sense). A **biorientation** of a mixed graph $M = (V, A, E)$ is obtained from M by replacing every unoriented edge xy of E by the arc xy, the arc yx or the pair xy, yx of arcs. If no unoriented edge is replaced by a pair of arcs, we speak of an **orientation** of a mixed graph[3]. The **complete biorientation** of a mixed graph $M = (V, A, E)$ is a biorientation $\overleftrightarrow{M}$ of M such that every unoriented edge $xy \in E$ is replaced in $\overleftrightarrow{M}$ by the pair xy, yx of arcs. Using the complete biorientation of a mixed graph M, one can easily give the definitions of a walk, trail, path and cycle in M. The only extra condition is that every pair of arcs in $\overleftrightarrow{M}$ obtained in replacement of an edge in M has to be treated as two appearances of one thing. For example, if one of the arcs in such a pair appears in a trail T, then the second one cannot be in T. A mixed graph M is **strong** if $\overleftrightarrow{M}$ is strong. Similarly, one can give the definition of strong components. A mixed graph M is **connected** if $\overleftrightarrow{M}$ is connected. An edge ℓ in a connected mixed graph M is a **bridge** if $M - \ell$ is not connected.

Figure 1.16 illustrates the notion of a mixed graph. The mixed graph M depicted in Figure 1.16 is strong; $u, (u, v), v, \{v, u\}, u$ is a cycle in M; $M - x$ has two strong components: one consists of the vertex y, the other is $M' = M\langle\{u, v, w\}\rangle$; the edge $\{v, w\}$ is a bridge in M', the arc (u, v) and the edge $\{u, v\}$ are not bridges in M'; M is bridgeless.

Theorem 1.8.1 below is due to Boesch and Tindell [162]. This result is an extension of Theorem 1.6.1. We give a short proof obtained by Volkmann [889]. (Another proof which leads to a linear time algorithm is obtained by Chung, Garey and Tarjan [218].)

[3] Note that a mixed graph $M = (V, A, E)$ may have a directed 2-cycle in which case no orientation of M is an oriented graph (because some 2-cycles remain).

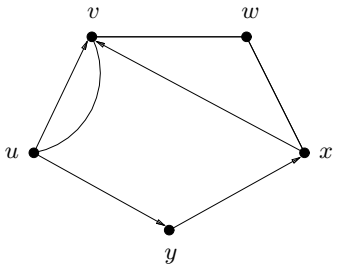


Figure 1.16 A mixed graph.

Theorem 1.8.1 *Let e be an unoriented edge in a strong mixed graph M. The edge e can be replaced by an arc (with the same end-vertices) such that the resulting mixed graph M' is strong if and only if e is not a bridge.*

Proof: If e is a bridge, then clearly there is no orientation of e that results in a strong mixed graph. Assume that e is not a bridge. Let $M' = M - e$. If M' is strong, then any orientation of e leads to a strong mixed graph; thus, assume that M' is not strong. Since e is not a bridge, M' is connected. Let $L_1, L_2, \ldots, L_k$ be strong components of M'. Since M is strong, there is only one initial strong component, say L_1, and only one terminal strong component, say L_k. Let u (v) be the end-vertex of e belonging to L_1 (L_k). By strong connectivity of $L_1, L_2, \ldots, L_k$ and Proposition 2.1.2 (applied to the strong component digraph of $\overleftrightarrow{M'}$), $M' + (v, u)$ is strong. □

An **orientation** of a digraph D is a subdigraph of D obtained from D by deleting exactly one arc between x and y for every pair $x \neq y$ of vertices such that both xy and yx are in D. See Figure 1.17 for an illustration of this definition.

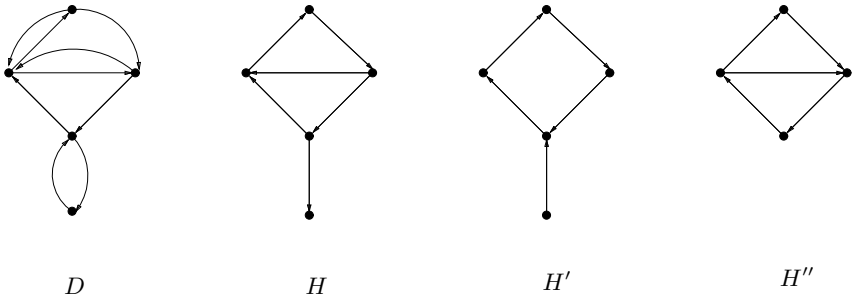


Figure 1.17 A digraph D and subdigraphs H, H' and H'' of D. The digraph H is an orientation of D but neither of H', H'' is an orientation of D.

Since we may transform a digraph to a mixed graph by replacing every 2-cycle with an undirected edge, we obtain the following reformulation of Theorem 1.8.1.

Corollary 1.8.2 *A strong digraph D has a strong orientation if and only if $UG(D)$ has no bridge.* □

A **hypergraph** is an ordered set $H = (V, \mathcal{E})$ such that V is a set (of **vertices** of H) and $\mathcal{E}$ is a family of subsets of V (called **edges** of H). The **rank** of H is the cardinality of the largest edge of H. For example, $H_0 = (\{1,2,3,4\}, \{\{1,2,3\}, \{2,3\}, \{1,2,4\}\})$ is a hypergraph. The rank of H_0 is three. The number of vertices in H is its **order**. We say that H is **2-colourable** if there is a function $f : V \rightarrow \{0,1\}$ such that, for every edge $E \in \mathcal{E}$, there exist a pair of vertices $x, y \in E$ such that $f(x) \neq f(y)$. The function f is called a **2-colouring** of H. It is easy to verify that H_0 is 2-colourable. In particular, $f(1) = f(2) = 0$, $f(3) = f(4) = 1$ is a 2-colouring of H_0. A hypergraph is **uniform** if all its edges have the same cardinality. Thus an undirected graph is a 2-uniform hypergraph.

1.9 Depth-First Search

In this section we will introduce a simple, yet very important, technique in algorithmic graph theory called **depth-first search**. While depth-first search (DFS) has certain similarities with BFS (see Section 3.3.1), DFS and BFS are quite different procedures, each with its own features.

Let $D = (V, A)$ be a digraph. In DFS, we start from an arbitrary vertex of D. At every stage of DFS, we **visit** some vertex x of D. If x has an **unvisited** out-neighbour y, we visit the vertex y[4]. We call the arc xy a **tree arc**. If x has no unvisited out-neighbour, we call x **explored** and return to the predecessor $\text{pred}(x)$ of x (the vertex from which we have moved to x) . If x does not have a predecessor, we find an unvisited vertex to 'restart' the above procedure. If such a vertex does not exist, we stop.

In our formal description of DFS, each vertex x of D gets two time-stamps: $\text{tvisit}(x)$ once x is visited and $\text{texpl}(x)$ once x is declared explored.

DFS
Input: A digraph $D = (V, A)$.
Output: $\text{pred}(v)$, $\text{tvisit}(v)$ and $\text{texpl}(v)$ for every $v \in V$.

1. For each $v \in V$ set $\text{pred}(v) :=$ **nil**, $\text{tvisit}(v) := 0$ and $\text{texpl}(v) := 0$.
2. Set $time := 0$.
3. For each vertex $v \in V$ do: if $\text{tvisit}(v) = 0$ then perform DFS-PROC(v).

[4] If x has more than one unvisited out-neighbour, we choose y as an arbitrary unvisited out-neighbour.

DFS-PROC(v)

1. Set $time := time + 1$, $\text{tvisit}(v) := time$.
2. For each $u \in N^+(v)$ do: if $\text{tvisit}(u) = 0$ then $\text{pred}(u) := v$ and perform DFS-PROC(u).
3. Set $time := time + 1$, $\text{texpl}(v) := time$.

Clearly, the main body of the algorithm takes $O(n)$ time. The total time for executing the different calls of the procedure DFS-PROC is $O(m)$ (as $\sum_{x \in V} d^+(x) = m$ by Proposition 1.2.1). As a result, the time complexity of DFS is $O(n + m)$.

Since each component of $UG(F)$ is a tree, F is a forest. We call F a **DFS forest**; a tree in F is a **DFS tree**; the **root** of a DFS tree is some vertex v used in Step 3 of the main body of DFS to initiate DFS-PROC. Clearly, the root r of a DFS tree T is the only vertex of T whose in-degree is zero. According to the above description of DFS every vertex in T can be reached from r by a path (hence T is an out-branching rooted at r in the subdigraph induced by $V(T)$). We say that a vertex x in T is a **descendant** of another vertex y in T (denoted by $x \succ y$) if y lies on the (r, x)-path in T. In this case, y is an **ancestor** of x. Note that in general there may be many different DFS forests for a given digraph D.

It is convenient to classify the non-tree arcs of a digraph $D = (V, A)$ with respect to a given DFS forest of D as follows. An arc xy is a **forward arc** if y is a descendant of x; xy is a **backward arc** if y is an ancestor of x. All other non-tree arcs are called **cross arcs**.

We illustrate the DFS algorithm and the above classification of arcs in Figure 1.18. The tree arcs are in bold. The non-tree arcs are labelled B,C or F depending on whether they are backward, cross or forward arcs. Every vertex u is time-stamped by $\text{tvisit}(u)/\text{texpl}(u)$ if one or both of them have been changed from the initial value of zero.

Observe that, for every vertex $v \in V$, we have $\text{tvisit}(v) < \text{texpl}(v)$. There is no pair u, v of vertices such that $\text{tvisit}(u) = \text{tvisit}(v)$ or $\text{tvisit}(u) = \text{texpl}(v)$ or $\text{texpl}(u) = \text{texpl}(v)$ due to the fact that before assigning any time to $\text{tvisit}(\ldots)$ or $\text{texpl}(\ldots)$ the value of *time* is increased. We consider some additional simple properties of DFS. We denote the interval from time t to time $t' > t$ by $[t, t']$ and write $I \subseteq I'$ if the interval I is contained in the interval I'.

Proposition 1.9.1 *Let $D = (V, A)$ and let the numbers $\text{tvisit}(v), \text{texpl}(v)$, $v \in V$, be calculated using DFS. For every pair of vertices u and v, one of the assertions below holds:*

(1) The intervals $[\text{tvisit}(u), \text{texpl}(u)]$ and $[\text{tvisit}(v), \text{texpl}(v)]$ are disjoint;
(2) $[\text{tvisit}(u), \text{texpl}(u)] \subseteq [\text{tvisit}(v), \text{texpl}(v)]$;
(3) $[\text{tvisit}(v), \text{texpl}(v)] \subseteq [\text{tvisit}(u), \text{texpl}(u)]$.

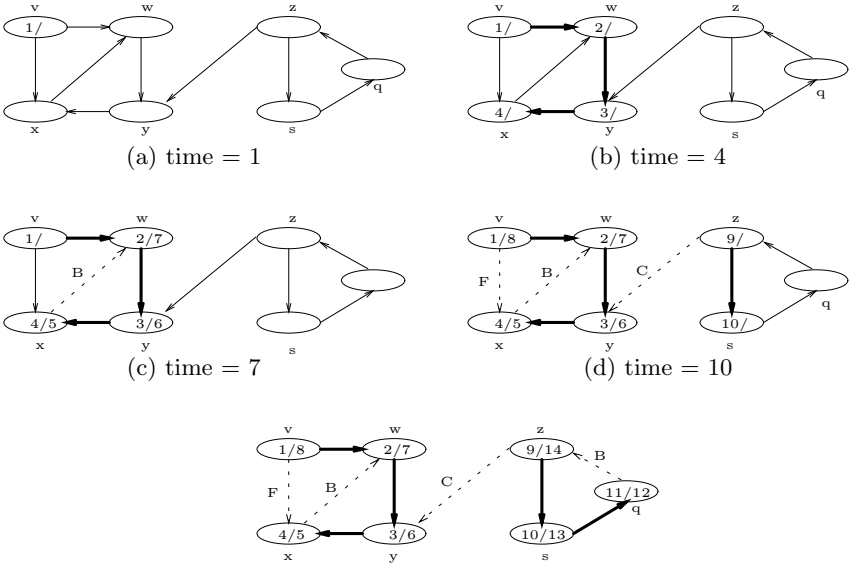


Figure 1.18 Some steps of DFS on a digraph starting from the vertex v.

Proof: Without loss of generality, we may assume that $\mathrm{tvisit}(u) < \mathrm{tvisit}(v)$. If $\mathrm{texpl}(u) < \mathrm{tvisit}(v)$, then the first assertion is valid. So, assume that $\mathrm{texpl}(u) > \mathrm{tvisit}(v)$. This means that v was visited when u has been already visited but u was not explored yet. Thus, there is a directed path from u to v in the DFS forest, implying that $v \succ u$. Since u cannot become explored when v is still unexplored, $\mathrm{texpl}(v) < \mathrm{texpl}(u)$. This implies the third assertion. □

This proposition implies immediately the following.

Corollary 1.9.2 *For a pair x, y of distinct vertices of D, we have $y \succ x$ if and only if* $\mathrm{tvisit}(x) < \mathrm{tvisit}(y) < \mathrm{texpl}(y) < \mathrm{texpl}(x)$. □

Proposition 1.9.3 *Let F be a DFS forest of a digraph $D = (V, A)$ and let x, y be vertices in the same DFS tree T of F. Then $y \succ x$ if and only if, at the time* $\mathrm{tvisit}(x)$*, the vertex y can be reached from x along a path all of whose internal vertices are unvisited.*

Proof: Assume that $y \succ x$. Let z be an internal vertex of the (x, y)-path in T. Thus, $z \succ x$. By Corollary 1.9.2, $\mathrm{tvisit}(x) < \mathrm{tvisit}(z)$. Hence, z is unvisited at time $\mathrm{tvisit}(x)$.

Suppose that y is reachable from x along a path P of unvisited vertices at time $\mathrm{tvisit}(x)$, but $y \not\succ x$. We may assume that $z = y_P^-$ (the predecessor

of y on P) is a descendant of x in T, that is, $z \succ x$ holds. By Corollary 1.9.2, $\text{texpl}(z) < \text{texpl}(x)$. Since y is an out-neighbour of z, y is visited before z is explored. Hence, $\text{tvisit}(y) < \text{texpl}(z)$. Clearly, $\text{tvisit}(x) < \text{tvisit}(y)$. Therefore, $\text{tvisit}(x) < \text{tvisit}(y) < \text{texpl}(x)$. By Proposition 1.9.1, it means that the interval $[\text{tvisit}(y), \text{texpl}(y)]$ is contained in the interval $[\text{tvisit}(x), \text{texpl}(x)]$. By Corollary 1.9.2, we conclude that $y \succ x$; a contradiction. □

1.10 Exercises

1.1. Let X and Y be finite sets. Show that $|X \cup Y| + |X \cap Y| = |X| + |Y|$.

1.2. Let X and Y be finite sets. Show that $|X \cup Y|^2 + |X \cap Y|^2 \geq |X|^2 + |Y|^2$.

1.3. **(−)** Prove that every tournament on $n \geq 2k + 2$ vertices has a vertex of out-degree at least $k + 1$.

1.4. **(−) Transitivity of paths.** Let D be a digraph and let x, y, z be vertices in D, $x \neq z$. Prove that if D has an (x, y)-path and a (y, z)-path, then it contains an (x, z)-path as well.

1.5. **(−) Decomposing a closed walk into cycles.** Prove that every closed walk can be decomposed into a collection of (not necessarily disjoint) cycles.

1.6. **Open walk decomposition.** Prove that every open walk can be decomposed into a path and some cycles (not necessarily disjoint).

1.7. **(−)** Prove that if the in-degree of every vertex in a digraph D is positive, then D has a cycle.

1.8. **(−)** Show that every digraph D contains a path of length at least $\delta^0(D)$.

1.9. Prove Proposition 1.4.3.

1.10. Show that a digraph D has a cycle factor if and only if its bipartite representation $BG(D)$ contains a perfect matching.

1.11. Prove Proposition 1.4.3.

1.12. Show that every oriented graph D on n vertices and with $\delta^0(D) \geq \lceil (n-1)/4 \rceil$ is strong. Show that this is best possible in terms of $\delta^0(D)$.

1.13. Let $T = (V, A)$ be a tournament such that every vertex is on a cycle. Prove that for every $a \in A$ the digraph $T - a$ is unilateral.

1.14. Prove that if a tournament T has a cycle, then it has two hamiltonian paths.

1.15. Let G be an undirected graph. Prove that either G or its complement $\overline{G}$ is connected.

1.16. Prove that every strong tournament T on at least four vertices has two distinct vertices x, y such that $T - x$ and $T - y$ are both strong.

1.17. **Strong connectivity is equivalent to cyclic connectivity in digraphs.** A digraph is **cyclically connected** if for every pair x, y of distinct vertices of D there is a sequence of cycles $C_1, \ldots, C_k$ such that x is in C_1, y is in C_k and C_i and C_{i+1} have at least one common vertex for every $i \in [k-1]$. Prove that a digraph D is strong if and only if it is cyclically connected.

1.18. Prove that a connected digraph is strong if and only if every arc is contained in a cycle. Hint: use the result of Exercise 1.17.

1.19. **(+) Preserving cycle subdigraphs**. Let D be a strong digraph with the property that, for every pair x, y of vertices, the deletion of all arcs between x and y results in a connected digraph. Let $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ be a cycle subdigraph in D such that every cycle C_i has length at least three. Prove that D has a strong spanning oriented subgraph containing $\mathcal{F}$. Hint: use Corollary 1.8.2 (Volkmann [889]).

1.20. Prove that the number of vertices of odd degree in an undirected graph is even.

1.21. Prove that every edge of a 2-edge-connected graph belongs to a cycle.

1.22. **(−)** Prove that an undirected tree of order n has $n - 1$ edges.

1.23. Prove that every undirected tree has a vertex of degree one.

1.24. Prove that every connected undirected graph G has a spanning tree.

1.25. Using the results of the last two exercises, prove that every connected undirected graph G has a vertex x such that $G - x$ is connected.

1.26. An undirected multigraph G is eulerian if it contains a closed trail T such that $E(T) = E(G)$. Prove without using Theorem 1.7.2 that G is eulerian if and only if G is connected and $d(x)$ is even for every vertex x of G.

1.27. **Almost balanced orientation.** Prove that every undirected graph $G = (V, E)$ has an orientation $D = (V, A)$ such that $|d_D^+(v) - d_D^-(v)| \leq 1$ for all $v \in V$. Hint: use Exercises 1.20 and 1.26.

1.28. Let $G = (V, E)$ be an eulerian graph. Using Exercise 1.26 and Corollary 1.7.3, prove that $d(W)$ is even for every proper subset W of V.

1.29. Prove that a digraph is strong if and only if it has a Hamilton closed walk.

1.30. Prove that every strong digraph H has an extension $D = H[\overline{K}_{n_1}, \ldots, \overline{K}_{n_h}]$, $h = |V(H)|$, such that D is hamiltonian. Hint: consider a hamiltonian closed walk in H.

1.31. A **transitive triple** in a digraph D is a set of three vertices x, y, z such that xy, xz and yz are arcs of D. Prove that if a 2-strong digraph D contains a transitive triple, then D has two cycles whose length differ by one.

1.32. Let $D = \vec{C}_r[\overline{K}_{n_1}, \ldots, \overline{K}_{n_r}]$ be an extension of a cycle. Prove that $\kappa(D) = \min\{n_i : \ i \in [r]\}$.

2. Classes of Digraphs

In this chapter we introduce several classes of digraphs. We study these classes with respect to their properties, characterization, recognition and decomposition. Further properties of the classes are studied in the following chapters of this book.

In Section 2.1 we study basic properties of acyclic digraphs. Acyclic digraphs form a very important family of digraphs and the reader will often encounter them in this book. Multipartite digraphs and extended digraphs are introduced in Section 2.2. These digraphs are studied in many other sections of our book. In Section 2.3, we introduce and study the transitive closure and a transitive reduction of a digraph. We use the notion of transitive reduction already in Section 2.6.

Several characterizations and a recognition algorithm for line digraphs are given in Section 2.4. We investigate basic properties of de Bruijn and Kautz digraphs and their generalizations in Section 2.5. These digraphs are extreme or almost extreme with respect to their diameter and vertex-strong connectivity. Series-parallel digraphs are introduced and studied in Section 2.6. These digraphs are of interest due to various applications such as scheduling. In the study of series-parallel digraphs we use notions and results of Sections 2.3 and 2.4.

An interesting generalization of transitive digraphs, the class of quasi-transitive digraphs, is considered in Section 2.7. The path-merging property of digraphs which is quite important for investigation of some classes of digraphs including tournaments is introduced and studied in Section 2.8. Two classes of path-mergeable digraphs, locally in-semicomplete and locally out-semicomplete digraphs, both generalizing the class of tournaments, are defined and investigated with respect to their basic properties in Section 2.9. The intersection of these two classes forms the class of locally semicomplete digraphs, which are studied in Section 2.10. There we give a very useful classification of locally semicomplete digraphs, which is applied in several proofs in other chapters. A characterization of a special subclass of locally semicomplete digraphs, called round digraphs, is also proved.

In Section 2.11, we study three classes of totally decomposable digraphs forming important generalizations of quasi-transitive digraphs as well as some other classes of digraphs. The aim of Section 2.11 is to investigate recognition

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_2,

of these three classes. Planar digraphs are discussed in Section 2.12. Digraphs of restricted tree-width are considered in Section 2.13. We show the usefulness of this class of graphs in designing polynomial algorithms and proving fixed-parameter tractability for some problems on digraphs. In Section 2.13, we also introduce and study directed tree-width, directed path-width and DAG-width. The last section is devoted to digraphs of three classes: circulant digraphs, arc-locally semicomplete digraphs and intersection digraphs.

2.1 Acyclic Digraphs

A digraph D is **acyclic** if it has no cycle. Acyclic digraphs form a well-studied family of digraphs of great interest in graph theory, algorithms and applications (see, e.g., Sections 2.3, 2.6, 3.3.2, 7.3, 10.4, 10.7, 17.2, 17.11, 17.15).

Recall that a vertex x in a digraph is sink (source) if $d^+(x) = 0$ ($d^-(x) = 0$).

Proposition 2.1.1 *Every acyclic digraph has a source and a sink.*

Proof: Let D be a digraph in which all vertices have positive out-degrees. We show that D has a cycle. Choose a vertex v_1 in D. Since $d^+(v_1) > 0$, there is a vertex v_2 such that $v_1 \rightarrow v_2$. As $d^+(v_2) > 0$, v_2 dominates some vertex v_3. Proceeding in this manner, we obtain walks of the form $v_1 v_2 \dots v_k$. As $V(D)$ is finite, there exists the least $k > 2$ such that $v_k = v_i$ for some $1 \leq i < k$. Clearly, $v_i v_{i+1} \dots v_k$ is a cycle.

Thus, an acyclic digraph D has a sink. Since the converse H of D is also acyclic, H has a sink v. Clearly, v has a source in D. □

Proposition 2.1.1 allows one to check whether a digraph D is acyclic: if D has a vertex of out-degree zero, then delete this vertex from D and consider the resulting digraph; otherwise, D contains a cycle. In the end of this section, we give another algorithm for verifying whether a digraph is acyclic.

Proposition 2.1.2 *Let D be an acyclic digraph with precisely one source x and one sink y in D. Then for every vertex $v \in V(D)$ there is an (x, v)-path and a (v, y)-path in D.*

Proof: A longest path starting at v (terminating at v) is certainly a (v, y)-path (an (x, v)-path). □

Let D be a digraph and let $x_1, x_2, \dots, x_n$ be an ordering of its vertices. We call this ordering an **acyclic ordering**[1] if, for every arc $x_i x_j$ in D, we

[1] Notice that in a majority of the literature an acyclic ordering is called a topological sorting. We feel that the name acyclic ordering is more appropriate, since no topology is involved.

have $i < j$. Clearly, an acyclic ordering of D induces an acyclic ordering of every subdigraph H of D. Since no cycle has an acyclic ordering, no digraph with a cycle has an acyclic ordering. On the other hand, the following holds:

Proposition 2.1.3 *Every acyclic digraph has an acyclic ordering of its vertices.*

Proof: We give a constructive proof by describing a procedure that generates an acyclic ordering of the vertices in an acyclic digraph D. At the first step, we choose a vertex v with in-degree zero. (Such a vertex exists by Proposition 2.1.1.) Set $x_1 = v$ and delete x_1 from D. At the ith step, we find a vertex u of in-degree zero in the remaining acyclic digraph, set $x_i = u$ and delete x_i from the remaining acyclic digraph. The procedure has $|V(D)|$ steps.

Suppose that $x_i \rightarrow x_j$ in D, but $i > j$. As x_j was chosen before x_i, it means that x_j was not of in-degree zero at the jth step of the procedure; a contradiction. □

Knuth [602] was the first to give a linear time algorithm for finding an acyclic ordering. Now we will show how to find an acyclic ordering in linear time using DFS described in the previous chapter. Below we assume that the input to the DFS algorithm is an acyclic digraph $D = (V, A)$. In the formal description of DFS let us add the following: $i := n + 1$ in line 2 of the main body of DFS and $i := i - 1$, $v_i := v$ in the last line of DFS-PROC. We obtain the following algorithm which we denote by **DFSA**:

DFSA(D)
Input: A digraph $D = (V, A)$.
Output: An acyclic ordering $v_1, \ldots, v_n$ of D.

1. For each $v \in V$ set pred(v) := **nil**, tvisit(v) := 0 and texpl(v) := 0.
2. Set $time := 0$, $i := n + 1$.
3. For each vertex $v \in V$ do: if tvisit(v) = 0 then perform DFSA-PROC(v).

DFSA-PROC(v)

1. Set $time := time + 1$, tvisit(v) := $time$.
2. For each $u \in N^+(v)$ do: if tvisit(u) = 0 then pred(u) := v and perform DFSA-PROC(u).
3. Set $time := time + 1$, texpl(v) := $time$, $i := i - 1$, $v_i := v$.

Theorem 2.1.4 *The algorithm DFSA correctly determines an acyclic ordering of any acyclic digraph in time $O(n + m)$.*

Proof: Since the algorithm is clearly linear (as DFS is linear), it suffices to show that the ordering $v_1, v_2, \ldots, v_n$ is acyclic. Observe that according to our algorithm

$$\text{texpl}(v_i) > \text{texpl}(v_j) \text{ if and only if } i < j. \tag{2.1}$$

Assume that D has an arc v_kv_s such that $k > s$. Hence, $\text{texpl}(v_s) > \text{texpl}(v_k)$. The arc v_kv_s is not a cross arc, because if it were, then by Proposition 1.9.1 and Corollary 1.9.2, the intervals for v_k and v_s would not intersect, i.e., v_k would be visited and explored before v_s would be visited; this and (2.1) make the existence of v_kv_s impossible. The arc v_kv_s is not a forward arc, because if it were, $\text{texpl}(v_s)$ would be smaller than $\text{texpl}(v_k)$. Therefore, v_kv_s must be a backward arc, i.e., $v_k \succ v_s$. Thus, there is a (v_s, v_k)-path in D. This path and the arc v_kv_s form a cycle, a contradiction. □

Figure 2.1 illustrates the result of applying DFSA to an acyclic digraph. The resulting acyclic ordering is z, w, u, y, x, v.

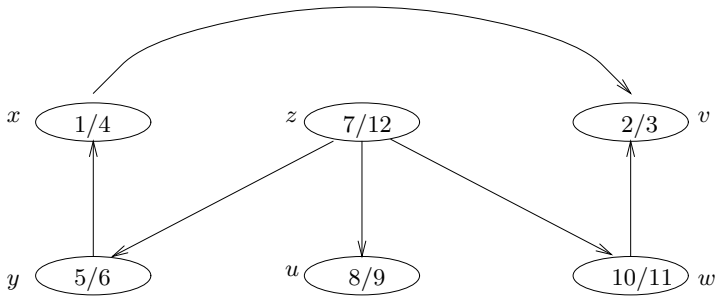


Figure 2.1 The result of applying DFSA to an acyclic digraph.

In Section 5.2 we apply DFSA to an arbitrary not necessarily acyclic digraph and see that the ordering $v_1, v_2, \ldots, v_n$ obtained by DFSA is very useful to determine the strong components of a digraph. DFSA allows us to check, in time $O(n + m)$, whether a digraph D is acyclic: we run DFSA and then verify whether the obtained ordering of the vertices is acyclic. Thus, we have the following:

Proposition 2.1.5 *One can check whether a digraph is acyclic in time* $O(n+m)$. □

2.2 Multipartite Digraphs and Extended Digraphs

A ***p*-partite digraph** is a biorientation of a p-partite graph; see Figure 2.2(b). Bipartite (i.e., 2-partite) digraphs are of special interest. It is well-known (and was proved already by König [618]) that an undirected graph is bipartite if and only if it has no cycle of odd length. The obvious extension of this statement to cycles in digraphs is not valid (the non-bipartite digraph

with vertex set $\{x, y, z\}$ and arc set $\{xy, xz, yz\}$ is such an example that can easily be generalized). However, the obvious extension does hold for strong bipartite digraphs. Theorem 2.2.1 can be traced back to the book [503] by Harary, Norman and Cartwright.

Theorem 2.2.1 *A strongly connected digraph is bipartite if and only if it has no cycle of odd length.*

Proof: If D is bipartite, then it is easy to see that D cannot have an odd cycle. To prove sufficiency suppose that D has no odd cycle. Fix an arbitrary vertex x in D. We claim that for every vertex $y \in V(D) - x$ and every choice of an (x, y)-path P and a (y, x)-path Q, the length of P and Q are equal modulo 2.

Suppose this is not the case for some choice of y, P and Q. Then choose y, P and Q such that the parity of the lengths of P and Q differ and $|V(P)| + |V(Q)|$ is minimum among all such pairs of (x, y)- and (y, x)-paths whose lengths differ in parity. If $V(P) \cap V(Q) = \{x, y\}$, then PQ is an odd cycle, contradicting the assumption. Hence there is a vertex $z \notin \{x, y\}$ in $V(P) \cap V(Q)$. Let z be chosen as the first such vertex that we meet when we traverse Q from y towards x. Then $P[z, y]Q[y_Q^+, z]$ is a cycle and it is even by our assumption. But now it is easy to see that the parity of the paths $P[x, z]$ and $Q[z, x]$ are different and we get a contradiction to the choice of y, P and Q. This proves the claim and, in particular, it follows that for every $y \in V(D) - x$, the lengths of all paths from x to y have the same parity.

Now let $U = \{y :$ the length of every (x, y)-path is even$\}$ and $U' = \{y :$ the length of every (x, y)-path is odd$\}$. This is a bipartition of $V(D)$ and neither U nor U' contains two vertices which are joined by an arc, since that would imply that some vertex had paths of two different parities from x. □

An extension of this theorem to digraphs whose cycles are all of length 0 modulo $k \geq 2$ is given in Theorem 17.8.1.

Recall that tournaments are orientations of complete graphs. Recall that a **semicomplete digraph** is a biorientation of a complete graph (see Figure 2.2(a)) and a **tournament** is an orientation of a complete digraph. The complete biorientation of a complete graph is a **complete digraph** (denoted by $\overleftrightarrow{K}_n$). The notion of semicomplete digraphs and their special subclass, tournaments, can be extended in various ways. A biorientation of a complete p-partite (multipartite) graph is a **semicomplete p-partite (multipartite) digraph**; see Figure 2.2(c). A **multipartite tournament** is an orientation of a complete multipartite graph. A semicomplete 2-partite digraph is also called a **semicomplete bipartite digraph**. A **bipartite tournament** is a semicomplete bipartite digraph with no 2-cycles.

One can use the operation of extension introduced in Section 1.3 to define 'extensions' of the above classes of digraphs. We will speak of **extended semicomplete digraphs** (i.e., extensions of semicomplete digraphs), **extended locally in-semicomplete digraphs**, **extended locally semi-**

(a) K_4 and a semicomplete digraph of order four.

(b) A 3-partite graph G and a biorientation of G.

(c) The complete 3-partite graph $K_{2,1,2}$ and a semicomplete 3-partite digraph D.

Figure 2.2 Multipartite digraphs.

complete digraphs, etc. Clearly, every extended semicomplete digraph is a semicomplete multipartite digraph, which is not necessarily semicomplete, and every extended semicomplete multipartite digraph is still a semicomplete multipartite digraph. Therefore, the class of semicomplete multipartite digraphs is **extension-closed**, and the class of semicomplete digraphs is not.

2.3 Transitive Digraphs, Transitive Closures and Reductions

A digraph D is **transitive** if, for every pair xy and yz of arcs in D with $x \neq z$, the arc xz is also in D. Transitive digraphs form the underlying

graph-theoretical model in a number of applications. For example, transitive oriented graphs correspond very naturally to partial orders (see Section 13.5 for some terminology on partial orders, the correspondence between transitive oriented graphs and partial orders and some basic results). The aim of this section is to give a brief overview of some properties of transitive digraphs as well as transitive closures and reductions of digraphs.

It is easy to show that a tournament is transitive if and only it is acyclic (see Exercise 2.3) and a strong digraph D is transitive if and only if D is complete[2]. We have the following simple characterization of transitive digraphs; its proof is left as Exercise 2.4.

Proposition 2.3.1 *Let D be a digraph with an acyclic ordering $D_1, D_2, \ldots, D_p$ of its strong components. The digraph D is transitive if and only if each of D_i is complete, the digraph H obtained from D by contraction of $D_1, \ldots, D_p$ followed by deletion of multiple arcs is a transitive oriented graph, and $D = H[D_1, D_2, \ldots, D_p]$, where $p = |V(H)|$.* □

The **transitive closure** $TC(D)$ of a digraph D is a digraph with $V(TC(D)) = V(D)$ and, for distinct vertices u, v, the arc $uv \in A(TC(D))$ if and only if D has a (u, v)-path. Clearly, if D is strong, then $TC(D)$ is a complete digraph. The uniqueness of the transitive closure of an arbitrary digraph is obvious. To compute the transitive closure of a digraph one can obviously apply the Floyd-Warshall algorithm in Chapter 3. To obtain a faster algorithm for the problem one can use the fact discovered by a number of researchers (see, e.g., the paper [318] by Fisher and Meyer, or [370] by Furman) that the transitive closure problem and the matrix multiplication problem are closely related: there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of a digraph of order n if and only if the product of two boolean $n \times n$ matrices can be computed in $O(n^a)$ time. Coppersmith and Winograd [230] showed that there exists an $O(n^{2.376})$-algorithm for the matrix multiplication. Goralcikova and Koubek [423] designed an $O(nm_{red})$-algorithm to find the transitive closure of an acyclic digraph D with n vertices and m_{red} arcs in the transitive reduction of D (the notion of transitive reduction is introduced below). This algorithm was also studied and improved by Mehlhorn [691] and Simon [820].

An arc uv in a digraph D is **redundant** if there is a (u, v)-path in D which does not contain the arc uv. A **transitive reduction** of a digraph D is a spanning subdigraph H of D with no redundant arc such that the transitive closures of D and H coincide. Not every digraph D has a unique transitive reduction. Indeed, if D has a pair of hamiltonian cycles, then each of these cycles is a transitive reduction of D. Below we show that a transitive reduction of an acyclic digraph is unique, i.e., we may speak of *the* transitive

[2] By the definition of a transitive digraph, a 2-cycle xyx does not force a loop at x and y.

reduction of an acyclic digraph. The **intersection of digraphs** $D_1, \ldots, D_k$ with the same vertex set V is the digraph H with vertex set V and arc set $A(D_1) \cap \ldots \cap A(D_k)$. Similarly one can define the union of digraphs with the same vertex sets (see Section 1.3). Let $\mathcal{S}(D)$ be the set of all spanning subdigraphs L of D for which $TC(L) = TC(D)$.

Theorem 2.3.2 *[10] For an acyclic digraph D, there exists a unique digraph D' with the property that $TC(D') = TC(D)$ and every proper subdigraph H of D' satisfies $TC(H) \neq TC(D')$. The digraph D' is the intersection of digraphs in $\mathcal{S}(D)$.*

The proof of this theorem, which is due to Aho, Garey and Ullman, follows from Lemmas 2.3.3 and 2.3.4.

Lemma 2.3.3 *Let D and H be a pair of acyclic digraphs on the same vertex set such that $TC(D) = TC(H)$ and $A(D) - A(H) \neq \emptyset$. Then, for every $e \in A(D) - A(H)$, we have $TC(D - e) = TC(D)$.*

Proof: Let $e = xy \in A(D) - A(H)$. Since $e \notin A(H)$, H must have an (x, y)-path passing through some other vertex, say z. Hence, D has an (x, z)-path P_{xz} and a (z, y)-path P_{zy}. If P_{xz} contains e, then D has a (y, z)-path. The existence of this path contradicts the existence of P_{zy} and the hypothesis that D is acyclic. Similarly, one can show that P_{zy} does not contain e. Therefore, $D - e$ has an (x, y)-path. Hence, $TC(D - e) = TC(D)$. □

Lemma 2.3.4 *Let D be an acyclic digraph. Then the set $\mathcal{S}(D)$ is closed under union and intersection.*

Proof: Let G, H be a pair of digraphs in $\mathcal{S}(D)$. Since $TC(G) = TC(H) = TC(D)$, $G \cup H$ is a subdigraph of $TC(D)$. The transitivity of $TC(D)$ now implies that $TC(G \cup H)$ is a subdigraph of $TC(D)$. Since G is a subdigraph of $G \cup H$, we have $TC(D)$ $(= TC(G))$ is a subdigraph of $TC(G \cup H)$. Thus, we conclude that $TC(G \cup H) = TC(D)$ and $G \cup H \in \mathcal{S}(D)$.

Now let $e_1, \ldots, e_p$ be the arcs of $G - A(G \cap H)$. By repeated application of Lemma 2.3.3, we obtain $TC(G - e_1 - e_2 - \ldots - e_p) = TC(G)$. This means that $TC(G \cap H) = TC(G) = TC(D)$, hence $G \cap H \in \mathcal{S}(D)$. □

Aho, Garey and Ullman [10] proved that there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of an arbitrary digraph D of order n if and only if a transitive reduction of D can be constructed in time $O(n^a)$. Therefore, we have

Proposition 2.3.5 *For an arbitrary digraph D, the transitive closure and a transitive reduction can be computed in time $O(n^{2.376})$.* □

Simon [821] described an $O(n+m)$-algorithm to find a transitive reduction of a strong digraph D. The algorithm uses DFS and two digraph transformations preserving $TC(D)$. This means that to have a linear time algorithm for

finding transitive reductions of digraphs from a certain class $\mathcal{D}$, it suffices to design a linear time algorithm for the transitive reduction of strong component digraphs of digraphs in $\mathcal{D}$. (Recall that the strong component digraph $SC(D)$ of a digraph D is obtained by contracting every strong component of D to a vertex followed by deletion of parallel arcs.) Such algorithms are considered, e.g., in the paper [485] by Habib, Morvan and Rampon.

While Simon's linear time algorithm in [821] finds a *minimal* subdigraph D' of a strong digraph D such that $TC(D') = TC(D)$, no polynomial algorithm is known to find a subdigraph D'' of a strong digraph D with *minimum* number of arcs such that $TC(D'') = TC(D)$. This is not surprising due to the fact that the corresponding optimization problem is $\mathcal{NP}$-hard. Indeed, the problem to verify whether a strong digraph D of order n has a subdigraph D'' of size n such that $TC(D'') = TC(D)$ is equivalent to the hamiltonian cycle problem, which is $\mathcal{NP}$-complete by Theorem 6.1.1.

A subdigraph D'' of a digraph D with minimum number of arcs such that $TC(D'') = TC(D)$ is sometimes called a **minimum equivalent subdigraph** of D. By the above discussion, we see that a minimum equivalent subdigraph of an acyclic digraph is unique and can be found in polynomial time. This means that the main difficulty of finding a minimum equivalent subdigraph of an arbitrary digraph D lies in finding such subdigraphs for the strong components of D. This issue is addressed in Section 12.2 for some classes of digraphs studied in this chapter. For the classes in Section 12.2, the minimum equivalent subdigraph problem is polynomial time solvable.

2.4 Line Digraphs

For a directed pseudograph D, the **line digraph** $Q = L(D)$ has vertex set $V(Q) = A(D)$ and arc set

$$A(Q) = \{ab : a, b \in V(Q),\ \text{the head of } a \text{ coincides with the tail of } b\}.$$

A directed pseudograph H is a **line digraph** if there is a directed pseudograph D such that $H = L(D)$. See Figure 2.3. Clearly, line digraphs do not have parallel arcs; moreover, the line digraph $L(D)$ has a loop at a vertex $a \in A(D)$ if and only if a is a loop in D.

The following theorem provides a number of equivalent characterizations of line digraphs. Of these characterizations, (ii) is due to Harary and Norman [502], (iii) to Heuchenne [522] and (iv) and (v) to Richards [777]; conditions (ii) and (iii) have each been rediscovered several times, see the survey [516] by Hemminger and Beineke. The proof presented here is adapted from [516]. For an $n \times n$-matrix $M = [m_{ik}]$, a row i is **orthogonal** to a row j if $\sum_{k=1}^{n} m_{ik}m_{jk} = 0$. One can give a similar definition of orthogonal columns.

Theorem 2.4.1 *Let D be a directed pseudograph with vertex set $\{1, 2, \ldots, n\}$ and with no parallel arcs and let $M = [m_{ij}]$ be its adjacency matrix (i.e., the*

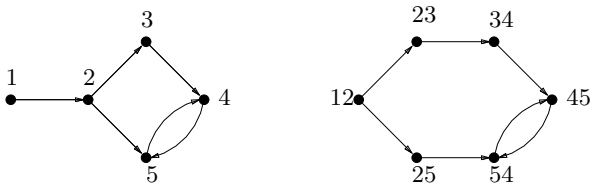


Figure 2.3 A digraph H and its line digraph $Q = L(H)$.

$n \times n$-matrix such that $m_{ij} = 1$, if $ij \in A(D)$, and $m_{ij} = 0$, otherwise). Then the following assertions are equivalent:

(i) D is a line digraph;
(ii) there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that

$$A(D) = \cup_{i\in I} A_i \times B_i;$$

(iii) if vw, uw and ux are arcs of D, then so is vx;
(iv) any two rows of M are either identical or orthogonal;
(v) any two columns of M are either identical or orthogonal.

Proof: We show the following implications and equivalences: (i) $\Leftrightarrow$ (ii), (ii) $\Rightarrow$ (iii), (iii) $\Rightarrow$ (iv), (iv) $\Leftrightarrow$ (v), (iv) $\Rightarrow$ (ii).

(i) $\Rightarrow$ (ii). Let $D = L(H)$. For each $v_i \in V(H)$, let A_i and B_i be the sets of in-coming and out-going arcs at v_i, respectively. Then the arc set of the subdigraph of D induced by $A_i \cup B_i$ equals $A_i \times B_i$. If $ab \in A(D)$, then there is an i such that $a = v_j v_i$ and $b = v_i v_k$. Hence, $ab \in A_i \times B_i$. The result follows.

(ii) $\Rightarrow$ (i). Let Q be the directed pseudograph with ordered pairs (A_i, B_i) as vertices, and with $|A_j \cap B_i|$ arcs from (A_i, B_i) to (A_j, B_j) for each i and j (including $i = j$). Let σ_{ij} be a bijection from $A_j \cap B_i$ to this set of arcs (from (A_i, B_i) to (A_j, B_j)) of Q. Then the function σ defined on $V(D)$ by taking σ to be σ_{ij} on $A_j \cap B_i$ is a well-defined function of $V(D)$ into $V(L(Q))$, since $\{A_j \cap B_i\}_{i,j\in I}$ is a partition of $V(D)$. Moreover, σ is a bijection since every σ_{ij} is a bijection. Furthermore, it is not difficult to see that σ is an isomorphism from D to $L(Q)$ (this is left as Exercise 2.6).

(ii) $\Rightarrow$ (iii). If vw, uw and ux are arcs of D, then there exist i, j such that $\{u, v\} \subseteq A_i$ and $\{w, x\} \subseteq B_j$. Hence, $(v, x) \in A_i \times B_j$ and $vx \in D$.

(iii) $\Rightarrow$ (iv). Assume that (iv) does not hold. This means that some rows, say i and j, are neither identical nor orthogonal. Then there exist k, h such that $m_{ik} = m_{jk} = 1$ and $m_{ih} = 1, m_{jh} = 0$ (or vice versa). Hence, ik, jk, ih are in $A(D)$ but jh is not. This contradicts (iii).

(iv) $\Leftrightarrow$ (v). Both (iv) and (v) are equivalent to the statement:

$$\text{for all } i, j, h, k, \text{ if } m_{ih} = m_{ik} = m_{jk} = 1, \text{ then } m_{jh} = 1.$$

(iv) $\Rightarrow$ (ii). For each i and j with $m_{ij} = 1$, let $A_{ij} = \{h : \ m_{hj} = 1\}$ and $B_{ij} = \{k : \ m_{ik} = 1\}$. Then, by (iv), A_{ij} is the set of vertices in D whose row vectors in M are identical to the ith row vector, whereas B_{ij} is the set of vertices in D whose column vectors in M are identical to the jth column vector (we use the previously proved fact that (iv) and (v) are equivalent). Thus, $A_{ij} \times B_{ij} \subseteq A(D)$, and moreover $A(D) = \cup\{A_{ij} \times B_{ij} : \ m_{ij} = 1\}$. By the orthogonality condition, A_{ij} and A_{hk} are either equal or disjoint, as are B_{ij} and B_{hk}. For zero row vector i in M, let A_{ij} be the set of vertices whose row vector in M is the zero vector, and let $B_{ij} = \emptyset$. Doing the same with the zero column vectors of M completes the partition as in (ii). □

The characterizations (ii)-(v) all imply polynomial algorithms to verify whether a given directed pseudograph is a line digraph. This fact is obvious regarding (iii)-(v); it is slightly more difficult to see that (ii) can be used to construct a very effective polynomial algorithm. We actually design such an algorithm for acyclic digraphs (as a pair of procedures illustrated by an example) just after Proposition 2.4.3. The criterion (iii) also provides the following characterization of line digraphs in terms of forbidden induced subdigraphs. Its proof is left as Exercise 2.7.

Corollary 2.4.2 *A directed pseudograph D is a line digraph if and only if D does not contain, as an induced subdigraph, any directed pseudograph that can be obtained from one of the directed pseudographs in Figure 2.4 (dotted arcs are missing) by adding zero or more arcs (other than the dotted ones).* □

Observe that the digraph of order 4 in Figure 2.4 corresponds to the case of distinct vertices in Part (iii) of Theorem 2.4.1, and the two directed pseudographs of order 2 correspond to the cases $x = u \neq v = w$ and $u = w \neq v = x$, respectively.

Clearly, Theorem 2.4.1 implies a set of characterizations of the line digraphs of digraphs (without parallel arcs and loops). This can be found in [516]. Several characterizations of special classes of line digraphs and iterated line digraphs can be found in surveys by Hemminger and Beineke [516] and Prisner [755].

Many applications of line digraphs deal with the line digraphs of special families of digraphs, for example regular digraphs, in general, and complete digraphs, in particular, see, e.g., the papers [279] by Du, Lyuu and Hsu and [316] by Fiol, Yebra and Alegre. In Section 2.6, we need the following characterization, due to Harary and Norman, of the line digraphs of acyclic directed multigraphs. It is a specialization of Parts (i) and (ii) of Theorem 2.4.1. The proof is left as (an easy) Exercise 2.8.

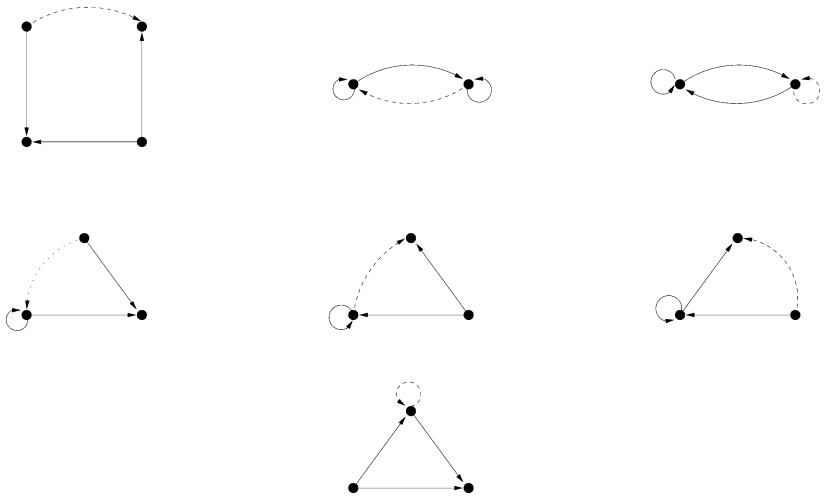


Figure 2.4 Forbidden directed pseudographs.

Proposition 2.4.3 *[502] A digraph D is the line digraph of an acyclic directed multigraph if and only if D is acyclic and there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that $A(D) = \cup_{i\in I} A_i \times B_i$.* □

We will now show how Proposition 2.4.3 can be used to recognize very effectively whether a given acyclic digraph R is the line digraph of another acyclic directed multigraph H, i.e., $R = L(H)$. The two procedures, which we construct and illustrate by Figure 2.7, can actually be used to recognize and represent (that is, to construct H such that $R = L(H)$) arbitrary line digraphs (see Theorem 2.4.1(i) and (ii)).

We first use Proposition 2.4.3 to check whether H above exists. The following procedure **Check-H** can be applied. Initially, all arcs and vertices of R are not marked. At every iteration, we choose an arc uv in R, which is not marked yet, and mark all vertices in $N^+(u)$ by 'B', all vertices in $N^-(v)$ by 'A' and all arcs in $(N^-(v), N^+(u))_R$ by 'C'. If $(N^-(v), N^+(u))_R \neq N^-(v) \times N^+(u)$ or if we mark a certain vertex or arc twice (starting from another arc $u'v'$) by the same symbol, then this procedure stops as there is no H such that $L(H) = R$. (We call these conditions **obstructions**.) If this procedure is performed to the end (i.e., every vertex and arc received a mark), then such H exists. It is not difficult to see, using Proposition 2.4.3, that Check-H correctly verifies whether H exists or not.

To illustrate Check-H, consider the digraph R_0 of Figure 2.7(a). Suppose that we choose the arc ab first. Then ab is marked, at the first iteration, together with the arcs af and ag. The vertex a receives 'A', the vertices b, f, g get 'B'. Suppose that fi is chosen at the second iteration. Then the arcs fh, fi, gh, gi are all marked at this iteration. The vertices f, g receive

'A', the vertices h, i 'B'. Suppose that bc is chosen at the third iteration. We see that this arc is the only arc marked at this iteration. The vertex b receives 'A', the vertex c 'B'. Finally, say, ce is chosen. Then both cd and ce are marked. The vertex c gets 'A', the vertices d, e receive 'B'. Thus, all arcs have been marked and no obstruction has taken place. This means that there exists a digraph H_0 such that $H_0 = L(R_0)$.

Suppose now that H does exist. The following procedure **Build-H** constructs such a directed multigraph H. By Proposition 2.4.3, if H exists, then all arcs of R can be partitioned into arc sets of bipartite tournaments with partite sets A_i and B_i and arc sets $A_i \times B_i$. Let us denote these digraphs by $T_1, \ldots, T_k$. (They can be computed by Check-H if we mark every $(N^-(v), N^+(u))_R$ not only by 'C' but also by a second mark 'i' starting from 1 and increasing by 1 at each iteration of the procedure.) We construct H as follows. The vertex set of H is $\{t_0, t_1, \ldots, t_k, t_{k+1}\}$. The arcs of H are obtained by the following procedure. For each vertex v of R, we append one arc a_v to H according to the rules below:

(a) If $d_R(v) = 0$, then $a_v := (t_0, t_{k+1})$;
(b) If $d_R^+(v) > 0, d_R^-(v) = 0$, then $a_v := (t_0, t_i)$, where i is the index of T_i such that $v \in A_i$;
(c) If $d_R^+(v) = 0, d_R^-(v) > 0$, then $a_v := (t_j, t_{k+1})$, where j is the index of T_j such that $v \in B_j$;
(d) If $d_R^+(v) > 0, d_R^-(v) > 0$, then $a_v := (t_i, t_j)$, where i and j are the indices of T_i and T_j such that $v \in A_j \cap B_i$.

It is straightforward to verify that $R = L(H)$. Note that Build-H always constructs H with only one vertex of in-degree zero and only one vertex of out-degree zero.

To illustrate Build-H, consider R_0 of Figure 2.7 once again. Earlier we showed that there exists H_0 such that $R_0 = L(H_0)$. Now we will construct H_0. The previous procedure applied to verify the existence of H_0 has implicitly constructed the digraphs $T_1 = (\{a, b, f, g\}, \{ab, af, ag\})$, $T_2 = (\{f, g, h, i\}, \{fh, fi, gh, gi\})$, $T_3 = (\{b, c\}, \{bc\})$, $T_4 = (\{c, d, e\}, \{cd, ce\})$. Thus, H_0 has vertices $t_0, \ldots, t_5$. Considering the vertices of R_0 in the lexicographic order, we obtain the following arcs of H_0 (in this order):

$$t_0t_1, t_1t_3, t_3t_4, t_4t_5, t_4t_5, t_1t_2, t_1t_2, t_2t_5, t_2t_5.$$

The directed multigraph H_0 is depicted in Figure 2.7(c). It is easy to check that $R_0 = L(H_0)$.

The iterated line digraphs are defined recursively: $L^1(D) = L(D)$, $L^{k+1}(D) = L(L^k(D))$, $k \geq 1$. It is not difficult to prove by induction (Exercise 2.10) that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1 \ldots v_k$ (which is a walk in D) dominates the vertex $v_1v_2 \ldots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that

$v_k v_{k+1} \in A(D)$. New characterizations of line digraphs and iterated line digraphs are given by Liu and West [648].

The following proposition can be proved by induction on $k \geq 1$ (Exercise 2.12).

Proposition 2.4.4 *Let D be a strong d-regular digraph ($d > 1$) of order n and diameter t. Then $L^k(D)$ is of order $d^k n$ and diameter $t + k$.* □

Bermond, Munos and Marchetti-Spaccamela [150] proposed broadcasting algorithms for line digraphs in the telephone mode. The protocols of [150] use a broadcasting protocol for a digraph D to obtain a broadcasting protocol for iterated line digraphs of D. As a consequence, improved bounds for the broadcasting time in de Bruijn and Kautz digraphs were obtained.

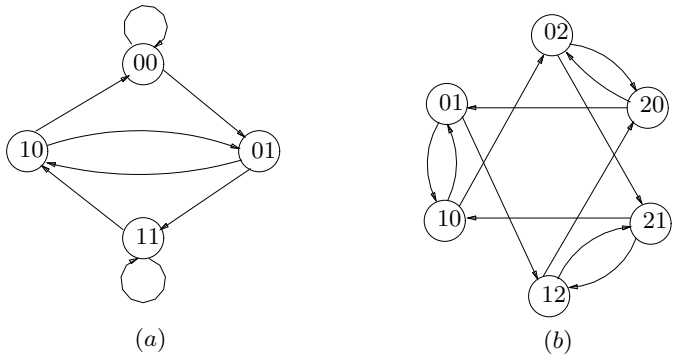
2.5 The de Bruijn and Kautz Digraphs

The following problem is of importance in network design. Given positive integers n and d, construct a digraph D of order n and maximum out-degree at most d such that $\mathrm{diam}(D)$ is as small as possible and the vertex-strong connectivity $\kappa(D)$ is as large as possible. So we have a 2-objective optimization problem. For such a problem, in general, no solution can maximize/minimize both objective functions. However, for this specific problem, there are solutions, which (almost) maximize/minimize both objective functions. The aim of this section is to introduce these solutions, the de Bruijn and Kautz digraphs, as well as some of their generalizations. For more information on the above classes of digraphs, the reader may consult the survey [276] by Du, Cao and Hsu. For applications of these digraphs in design of parallel architectures and large packet radio networks, see e.g. the papers [149] by Bermond and Hell, [151] by Bermond and Peyrat and [792] by Samatan and Pradhan.

Let V be the set of vectors with t coordinates, $t \geq 2$, each taken from $\{0, 1, \ldots d-1\}$, $d \geq 2$. The **de Bruijn digraph** $D_B(d, t)$ is the directed pseudograph with vertex set V such that $(x_1, x_2, \ldots, x_t)$ dominates $(y_1, y_2, \ldots, y_t)$ if and only if $x_2 = y_1, x_3 = y_2, \ldots, x_t = y_{t-1}$. See Figure 2.5(a). Let $D_B(d, 1)$ be the complete digraph of order d with loop at every vertex.

These directed pseudographs are named after de Bruijn who was the first to consider them in [252]. Clearly, $D_B(d, t)$ has d^t vertices and the out-pseudodegree and in-pseudodegree of every vertex of $D_B(d, t)$ equal d. This directed pseudograph has no parallel arcs and contains a loop at every vertex for which all coordinates are the same. It is natural to call $D_B(d, t)$ ***d*-pseudoregular** (recall that in the definition of semi-degrees we do not count loops).

Since $D_B(d, t)$ has loops at some vertices, the vertex-strong connectivity of $D_B(d, t)$ is at most $d - 1$ (indeed, the loops can be deleted without the vertex-strong connectivity being changed). Imase, Soneoka and Okada [550]



(a) (b)

Figure 2.5 (a) The de Bruijn digraph $D_B(2,2)$; (b) the Kautz digraph $D_K(2,2)$.

proved that $D_B(d,t)$ is $(d-1)$-strong, and moreover, for every pair $x \neq y$ of vertices there exist $d-1$ internally disjoint (x,y)-paths of length at most $t+1$. To prove this result we will use the following two lemmas. The proof of the first lemma, due to Fiol, Yebra and Alegre, is left as Exercise 2.13.

Lemma 2.5.1 *[316] For $t \geq 2$, $D_B(d,t)$ is the line digraph of $D_B(d,t-1)$.* □

Lemma 2.5.2 *Let x, y be distinct vertices of $D_B(d,t)$ such that $x \rightarrow y$. Then, there are $d-2$ internally disjoint (x,y)-paths different from xy, each of length at most $t+1$.*

Proof: Let $x = (x_1, x_2, \ldots, x_t)$ and $y = (x_2, \ldots, x_t, y_t)$. Consider the walk W_k given by $W_k = (x_1, x_2, \ldots, x_t), (x_2, \ldots, x_t, k), (x_3, \ldots, x_t, k, x_2), \ldots, (k, x_2, \ldots, x_t), (x_2, \ldots, x_t, y_t)$, where $k \neq x_1, y_t$. For each k, every internal vertex of W_k has coordinates forming the same multiset $M_k = \{x_2, \ldots, x_t, k\}$. Since for different k, the multisets M_k are different, the walks W_k are internally disjoint. Each of these walks is of length $t+1$. Therefore, by Proposition 1.4.1, $D_B(d,t)$ contains $d-2$ internally disjoint (x,y)-paths P_k with $A(P_k) \subseteq A(W_k)$. Since $k \neq x_1, y_t$, we may form the paths P_k such that none of them coincides with xy. □

Theorem 2.5.3 *[550] For every pair x, y of distinct vertices of $D_B(d,t)$, there exist $d-1$ internally disjoint (x,y)-paths, one of length at most t and the others of length at most $t+1$.*

Proof: By induction on $t \geq 1$. Clearly, the claim holds for $t = 1$ since $D_B(d,1)$ contains, as spanning subdigraph, $\overleftrightarrow{K}_d$. For $t \geq 2$, by Lemma 2.5.1, we have that

$$D_B(d,t) = L(D_B(d,t-1)). \tag{2.2}$$

Let x, y be a pair of distinct vertices in $D_B(d,t)$ and let e_x, e_y be the arcs of $D_B(d,t-1)$ corresponding to vertices x, y due to (2.2). Let u be the head of e_x and let v be the tail of e_y.

If $u \neq v$, by the induction hypothesis, $D_B(d,t-1)$ has $d-1$ internally disjoint (u,v)-paths, one of length at most $t-1$ and the others of length at most t. The arcs of these paths together with arcs e_x and e_y correspond to $d-1$ internally disjoint (x,y)-paths in $D_B(d,t)$, one of length at most t and the others of length at most $t+1$.

If $u = v$, we have $x \rightarrow y$ in $D_B(d,t-1)$. It suffices to apply Lemma 2.5.2 to see that there are $d-1$ internally disjoint (x,y)-paths in $D_B(d,t)$, one of length one and the others of length at most $t+1$. □

By this theorem and Corollary 5.4.2, we conclude that $\kappa(D_B(d,t)) = d-1$. From Theorem 2.5.3 and Proposition 3.4.3, we obtain immediately the following simple, yet important property.

Proposition 2.5.4 *The de Bruijn digraph $D_B(d,t)$ achieves the minimum value t of diameter for directed pseudographs of order d^t and maximum out-degree at most d.* □

For $t \geq 2$, the **Kautz digraph** $D_K(d,t)$ is obtained from $D_B(d+1,t)$ by deletion of all vertices of the form $(x_1, x_2, \ldots, x_t)$ such that $x_i = x_{i+1}$ for some i. See Figure 2.5(b). Define $D_K(d,1) := \overleftrightarrow{K}_{d+1}$. Clearly, $D_K(d,t)$ has no loops and is a d-regular digraph. Since we have $d+1$ choices for the first coordinate of a vertex in $D_K(d,t)$ and d choices for each of the other coordinates, the order of $D_K(d,t)$ is $(d+1)d^{t-1} = d^t + d^{t-1}$. It is easy to see that Proposition 2.5.4 holds for the Kautz digraphs as well.

The following lemmas are analogous to Lemmas 2.5.1 and 2.5.2. Their proofs are left as Exercises 2.14 and 2.15.

Lemma 2.5.5 *For $t \geq 2$, the Kautz digraph $D_K(d,t)$ is the line digraph of $D_K(d,t-1)$.* □

Lemma 2.5.6 *Let xy be an arc in $D_K(d,t)$. There are $d-1$ internally disjoint (x,y)-paths different from xy, one of length at most $t+2$ and the others of length at most $t+1$.* □

The following result due to Du, Cao and Hsu [276] shows that the Kautz digraphs are better, in a sense, than de Bruijn digraphs from the local vertex-strong connectivity point of view. This theorem can be proved similarly to Theorem 2.5.3 and is left as Exercise 2.16.

Theorem 2.5.7 *[276] Let x, y be distinct vertices of $D_K(d,t)$. Then there are d internally disjoint (x,y)-paths in $D_K(d,t)$, one of length at most t, one of length at most $t+2$ and the others of length at most $t+1$.* □

This theorem implies that $D_K(d,t)$ is d-strong.

The de Bruijn digraphs were generalized independently by Imase and Itoh [547] and Reddy, Pradhan and Kuhl [767] in the following way. We can transform every vector $(x_1, x_2, \ldots, x_t)$ with coordinates from $Z_d = \{0, 1, \ldots, d-1\}$ into an integer from $Z_{d^t} = \{0, 1, \ldots, d^t - 1\}$ using the polynomial $P(x_1, x_2, \ldots, x_t) = x_1 d^{t-1} + x_2 d^{t-2} + \ldots + x_t$. It is easy to see that this polynomial provides a bijection from Z_d^t to Z_{d^t}. Moreover, for $i, j \in Z_{d^t}$, $i \rightarrow j$ in $D_B(d,t)$ if and only if $j \equiv di + k \pmod{d^t}$ for some $k \in Z_d$.

Let d, n be two natural numbers such that $d < n$. The **generalized de Bruijn digraph** $D_G(d,n)$ is a directed pseudograph with vertex set Z_n and arc set

$$\{(i, di + k \pmod{n}) : i, k \in Z_d\}.$$

For example, $V(D_G(2,5)) = \{0,1,2,3,4\}$ and $A(D_G(2,5)) = \{(0,0),(0,1),(1,2),(1,3),(2,4),(2,0),(3,1),(3,2),(4,3),(4,4)\}$.

Clearly, $D_G(d,n)$ is d-pseudoregular. It is not difficult to show that $\mathrm{diam}(D_G(d,n)) \leq \lceil \log_d n \rceil$. By Proposition 3.4.3, a digraph of maximum outdegree at most $d \geq 2$ and order n has a diameter at least $\lfloor \log_d n(d-1) + 1 \rfloor$. Thus, the generalized de Bruijn digraphs are of optimal or almost optimal diameter. It was proved, by Imase, Soneoka and Okada [549], that $D_G(d,n)$ is $(d-1)$-strong. It follows from these results that the generalized de Bruijn digraphs have almost minimum diameter and almost maximum vertex-strong connectivity.

The Kautz digraphs were generalized by Imase and Itoh [548]. Let n, d be two natural numbers such that $d < n$. The Imase-Itoh digraph $D_I(d,n)$ is the digraph with vertex set $\{0, 1, \ldots, n-1\}$ such that $i \rightarrow j$ if and only if $j \equiv -d(i+1) + k \pmod{n}$ for some $k \in \{0, 1, \ldots, d-1\}$. It has been shown (for a brief account, see the paper [276]) by Du, Cao and Hsu, that $D_I(d,n)$ are of (almost) optimal diameter and vertex-strong connectivity.

Du, Hsu and Hwang [278] suggested a concept of digraphs extending both the generalized de Bruijn digraphs and the Imase-Ito digraphs. Let d, n be two natural numbers such that $d < n$. Given $q \in [n-1]$ and $r \in \{0, 1, \ldots, n-1\}$, a **consecutive-$d$ digraph** $D(d,n,q,r)$ is the directed pseudograph with vertex set $\{0, 1, \ldots, n-1\}$ such that $i \rightarrow j$ if and only if $j \equiv qi + r + k \pmod{n}$ for some $k \in \{0, 1, \ldots, d-1\}$. Several results on diameter, vertex- and arc-strong connectivity and other properties of consecutive-d digraphs are given in [276]. In Section 6.9, we provide results on hamiltonicity of consecutive-d digraphs.

2.6 Series-Parallel Digraphs

In this section we study vertex series-parallel digraphs and arc series-parallel directed multigraphs. Vertex series-parallel digraphs were introduced by Lawler [637] and Monma and Sidney [701] as a model for scheduling problems. While vertex series-parallel digraphs continue to play an important role

for the design of efficient algorithms in scheduling and sequencing problems, they have been extensively studied in their own right as well as in relations to other optimization problems (cf. the papers [55] by Baffi and Petreschi, [153] by Bertolazzi, Cohen, Di Battista, Tamassia and Tollis, [776] by Rendl and [832] by Steiner). Arc series-parallel directed multigraphs were introduced even earlier (than vertex series-parallel digraphs) by Duffin [281] as a mathematical model of electrical networks.

For an acyclic digraph D, let F_D (I_D) be the set of vertices of D of out-degree (in-degree) zero. To define vertex series-parallel digraphs, we first introduce **minimal vertex series-parallel (MVSP) digraphs** recursively.

The digraph of order one with no arc is an MVSP digraph. If $D = (V, A)$, $H = (U, B)$ is a pair of MVSP digraphs ($U \cap V = \emptyset$), so are the acyclic digraphs constructed by each of the following operations (see Figure 2.6):

(a) **Parallel composition:** $P = (V \cup U, A \cup B)$;
(b) **Series composition:** $S = (V \cup U, A \cup B \cup (F_D \times I_H))$.

It is interesting to note that we can embed every MVSP digraph D into the Cartesian plane such that if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, then there is a (u, v)-path in D if and only if $x_u \leq x_v$ and $y_u \leq y_v$. The proof of this non-difficult fact is given in the paper [883] by Valdes, Tarjan and Lawler; see Exercise 2.17. See also Figure 2.8.

An acyclic digraph D is a **vertex series-parallel (VSP)** digraph if the transitive reduction of D is an MVSP digraph (see Section 2.3 for the definition of the transitive reduction). See Figure 2.7.

The following class of acyclic directed multigraphs, **arc series-parallel (ASP) directed multigraphs**, is related to VSP digraphs. The digraph $\vec{P}_2$ is an ASP directed multigraph. If D_1, D_2 is a pair of ASP directed multigraphs with $V(D_1) \cap V(D_2) = \emptyset$, then so are acyclic directed multigraphs constructed by each of the following operations (see Figure 2.9):

(a) **Two-terminal parallel composition:** Choose a vertex u_i of out-degree zero in D_i and a vertex v_i of in-degree zero in D_i for $i = 1, 2$. Identify u_1 with u_2 and v_1 with v_2;
(b) **Two-terminal series composition:** Choose $u \in F_{D_1}$ and $v \in I_{D_2}$ and identify u with v.

Observe that every ASP directed multigraph has a unique vertex of out-degree zero and a unique vertex of in-degree zero. We refer the reader to the book [127] by Battista, Eades, Tamassia and Tollis for several algorithms for drawing graphs nicely, in particular drawing of ASP digraphs.

The next result shows a relation between the classes of digraphs introduced above.

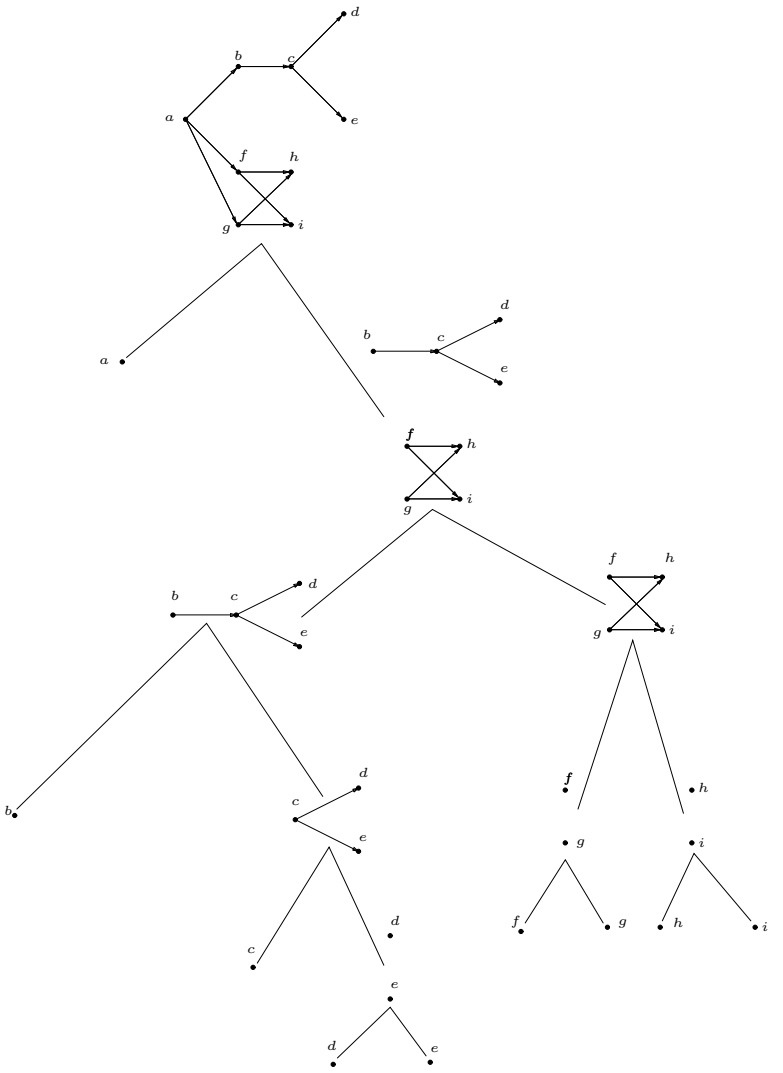
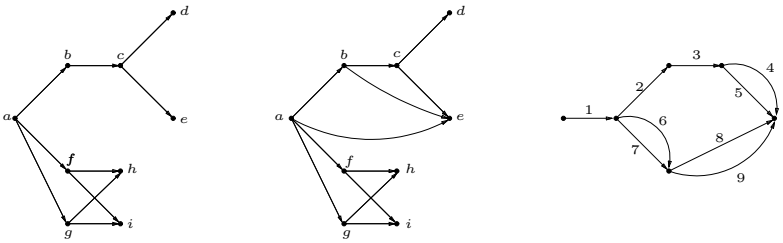


Figure 2.6 (De)construction of an MVSP digraph R_0 by series and parallel (de)compositions.

Theorem 2.6.1 *An acyclic directed multigraph D with a unique vertex of out-degree zero and a unique vertex of in-degree zero is ASP if and only if $L(D)$ is an MVSP digraph.*

Proof: This can be proved easily by induction on $|A(D)|$ using the following two facts:

(i) $L(\vec{P}_2) = \vec{P}_1$, which is an MVSP digraph;



(a) (b) (c)

Figure 2.7 Series-parallel directed multigraphs: (a) an MVSP digraph R_0, (b) a VSP digraph R_1, (c) an ASP directed multigraph H_0.

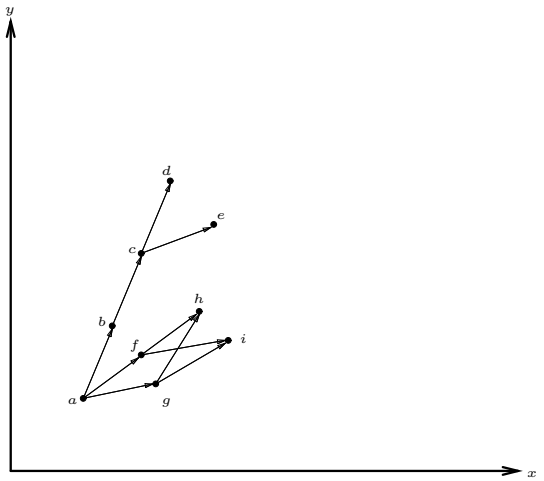


Figure 2.8 The MVSP digraph R_0 of Figure 2.6 embedded into the Cartesian plane such that for every (u, v)-path in R_0 we have $x_u \leq x_v$ and $y_u \leq y_v$ (and vice versa).

(ii) The line digraph of the two-terminal series (parallel) composition of D_1 and D_2 is the series (parallel) composition of $L(D_1)$ and $L(D_2)$. □

It is easy to check that $L(H_0) = R_0$ for directed multigraphs H_0 and R_0 depicted in Figure 2.7. The following operations in a directed multigraph D are called **reductions:**

(a) **Series reduction:** Replace a path uvw, where $d_D^+(v) = d_D^-(v) = 1$ by the arc uw;
(b) **Parallel reduction:** Replace a pair of parallel arcs from u to v by just one arc from u to v.

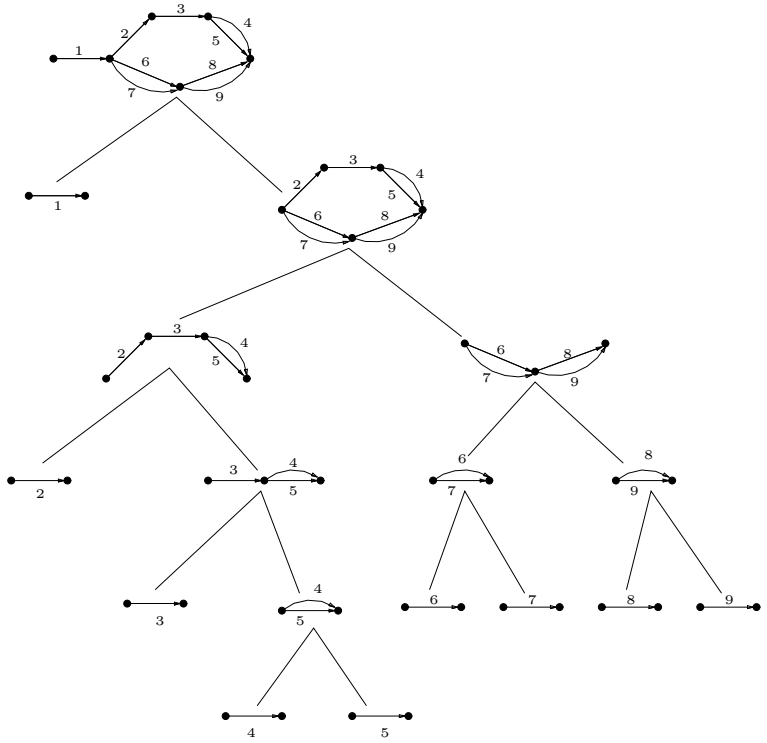


Figure 2.9 (De)construction of an ASP directed multigraph H_0 by two-terminal series and parallel (de)compositions.

The following proposition due to Duffin (see also the paper [883] by Valdes, Tarjan and Lawler) gives a characterization of ASP directed multigraphs. Its proof is left as Exercise 2.18.

Proposition 2.6.2 *[281] A directed multigraph is ASP if and only if it can be reduced to $\vec{P}_2$ by a sequence of series and parallel reductions.* □

The reader is advised to apply a sequence of series and parallel reductions to the directed multigraph H_0 of Figure 2.7 to obtain a digraph isomorphic to $\vec{P}_2$. From the algorithmic point of view, it is important that *every* sequence of series and parallel reductions transforms a directed multigraph to the same digraph. Indeed, this implies an obvious polynomial algorithm to verify if a given directed multigraph is ASP. The proof of the following result, due to Harary, Krarup and Schwenk, is left as Exercise 2.19.

Proposition 2.6.3 *[500] For every acyclic directed multigraph D, the result of application of series and parallel reductions until one can apply such reductions is a unique digraph H.* □

In [883], Valdes, Tarjan and Lawler showed how to construct a linear-time algorithm to recognize ASP directed multigraphs, which is based on Propositions 2.6.2 and 2.6.3. They also presented a more complicated linear-time algorithm to recognize VSP digraphs. Since we are limited in space, we will not discuss the details of the linear-time algorithms. Instead, we will consider the following simplified polynomial algorithm to recognize VSP digraphs.

VSP recognition algorithm
Input: An acyclic digraph D.
Output: YES if D is VSP and NO, otherwise.

1. Compute the transitive reduction R of D.
2. Try to compute an acyclic directed multigraph H with $|I_H| = |F_H| = 1$ such that $L(H) = R$. If there is no such H, then output NO.
3. Verify whether H is an ASP directed multigraph. If it is so, then YES, otherwise, NO.

We prove first the correctness of this algorithm. If the output is YES, then, by Theorem 2.6.1, R is MVSP and thus D is VSP. If H in Step 2 is not found, then, by Theorem 2.6.1, R is not MVSP implying that D is not VSP. If H is not ASP, then R is not MVSP by the same theorem.

Now we prove that the algorithm is polynomial. Step 1 can be performed in polynomial time by Proposition 2.3.5. Step 2 can be implemented using Procedure Build-H described at the end of Section 2.4. This procedure implies that if there is an H such that $L(H) = R$, then there is such an H with additional property that $|I_H| = |F_H| = 1$. The procedure is polynomial. Finally, Step 3 is polynomial by the remark after Proposition 2.6.2.

2.7 Quasi-Transitive Digraphs

A digraph D is **quasi-transitive** if, for every triple x, y, z of distinct vertices of D such that xy and yz are arcs of D, there is at least one arc between x and z. Clearly, a semicomplete digraph is quasi-transitive. Note that if there is only one arc between x and z, it can have any direction; hence quasi-transitive digraphs are generally not transitive.

The aim of this section is to derive a recursive characterization of quasi-transitive digraphs which allows one to show that a number of problems for quasi-transitive digraphs including the longest path and cycle problems are polynomial time solvable (see Sections 6.7 and 6.8). The characterization implies that every quasi-transitive digraph is totally Ψ-decomposable, where Ψ is the union of all transitive digraphs and all extended semicomplete digraphs. Our presentation is based on the paper [103] by Bang-Jensen and Huang.

T Q

Figure 2.10 A transitive digraph T and a quasi-transitive digraph Q.

An (x_1, x_n)-path $P = x_1x_2 \dots x_n$ is **minimal** if, for every (x_1, x_n)-path Q, either $V(P) = V(Q)$ or Q has a vertex not in $V(P)$.

Proposition 2.7.1 *Let D be a quasi-transitive digraph. Suppose that $P = x_1x_2 \dots x_k$ is a minimal (x_1, x_k)-path. Then the subdigraph induced by $V(P)$ is a semicomplete digraph and $x_j \to x_i$ for every $2 \leq i + 1 < j \leq k$, unless $k = 4$, in which case the arc between x_1 and x_k may be absent.*

Proof: The cases $k = 2, 3, 4, 5$ are easily verified. As an example, let us consider the case $k = 5$. If x_i and x_j are adjacent and $2 \leq i + 1 < j \leq 5$, then $x_j \to x_i$ since P is minimal. Since D is quasi-transitive, x_i and x_{i+2} are adjacent for $i = 1, 2, 3$. This and the minimality of P imply that $x_3 \to x_1, x_4 \to x_2$ and $x_5 \to x_3$. From these arcs and the minimality of P we conclude that $x_5 \to x_1$. Now the arcs x_4x_5 and x_5x_1 imply that $x_4 \to x_1$. Similarly, $x_5 \to x_1 \to x_2$ implies $x_5 \to x_2$.

The proof for the case $k \geq 6$ is by induction on k with the case $k = 5$ as the basis. By induction, each of $D\langle\{x_1, x_2, \dots, x_{k-1}\}\rangle$ and $D\langle\{x_2, x_3, \dots, x_k\}\rangle$ is a semicomplete digraph and $x_j \to x_i$ for any $1 < j - i \leq k - 2$. Hence x_3 dominates x_1 and x_k dominates x_3 and the minimality of P implies that x_k dominates x_1. □

Corollary 2.7.2 *If a quasi-transitive digraph D has an (x, y)-path but x does not dominate y, then either $y \to x$, or there exist vertices $u, v \in V(D) - \{x, y\}$ such that $x \to u \to v \to y$ and $y \to u \to v \to x$.*

Proof: This is easy to deduce by considering a minimal (x, y)-path and applying Proposition 2.7.1. □

Lemma 2.7.3 *Suppose that A and B are distinct strong components of a quasi-transitive digraph D with at least one arc from A to B. Then $A \mapsto B$.*

Proof: Suppose A and B are distinct strong components such that there exists an arc from A to B. Then for every choice of $x \in A$ and $y \in B$ there exists a path from x to y in D. Since A and B are distinct strong components, none of the alternatives in Corollary 2.7.2 can hold and hence $x \to y$. □

Lemma 2.7.4 *[103] Let D be a strong quasi-transitive digraph on at least two vertices. Then the following holds:*

(a) $\overline{UG(D)}$ *is disconnected;*
(b) If S and S' are two subdigraphs of D such that $\overline{UG(S)}$ and $\overline{UG(S')}$ are distinct connected components of $\overline{UG(D)}$, then either $S \mapsto S'$ or $S' \mapsto S$, or both $S \to S'$ and $S' \to S$ in which case $|V(S)| = |V(S')| = 1$.

Proof: The statement (b) can be easily verified from the definition of a quasi-transitive digraph and the fact that S and S' are completely adjacent in D (Exercise 2.20). We prove (a) by induction on $|V(D)|$. Statement (a) is trivially true when $|V(D)| = 2$ or 3. Assume that it holds when $|V(D)| < n$ where $n > 3$.

Suppose that there is a vertex z such that $D - z$ is not strong. Then there is an arc from (to) every terminal (initial) component of $D - z$ to (from) z. Since D is quasi-transitive, the last fact and Lemma 2.7.3 imply that $X \to Y$ for every initial (terminal) strong component X (Y) of $D - z$. Similar arguments show that each strong component of $D - z$ either dominates some terminal component or is dominated by some initial component of $D - z$ (intermediate strong components satisfy both). These facts imply that z is adjacent to every vertex in $D - z$. Therefore, $\overline{UG(D)}$ contains a component consisting of the vertex z, implying that $\overline{UG(D)}$ is disconnected and (a) follows.

Assume that there is a vertex v such that $D - v$ is strong. Since D is strong, D contains an arc vw from v to $D - v$. By induction, $\overline{UG(D - v)}$ is not connected. Let connected components S and S' of $\overline{UG(D - v)}$ be chosen such that $w \in S$, $S \mapsto S'$ in D (here we use (b) and the fact that $D - v$ is strong). Then v is completely adjacent to S' in D (as $v \to w$). Hence $\overline{UG(S')}$ is a connected component of $\overline{UG(D)}$ and the proof is complete. □

The following theorem completely characterizes quasi-transitive digraphs in recursive sense (see also Figure 2.11).

Theorem 2.7.5 (Bang-Jensen and Huang) *[103] Let D be a digraph which is quasi-transitive.*

(a) If D is not strong, then there exist a transitive oriented graph T with vertices $\{u_1, u_2, \ldots, u_t\}$ and strong quasi-transitive digraphs $H_1, H_2, \ldots, H_t$ such that $D = T[H_1, H_2, \ldots, H_t]$, where H_i is substituted for u_i, $i = 1, 2, \ldots, t$.
(b) If D is strong, then there exists a strong semicomplete digraph S with vertices $\{v_1, v_2, \ldots, v_s\}$ and quasi-transitive digraphs $Q_1, Q_2, \ldots, Q_s$ such that Q_i is either a vertex or is non-strong and $D = S[Q_1, Q_2, \ldots, Q_s]$, where Q_i is substituted for v_i, $i = 1, 2, \ldots, s$.

Proof: Suppose that D is not strong and let $H_1, H_2, \ldots, H_t$ be the strong components of D. According to Lemma 2.7.3, if there is an arc between

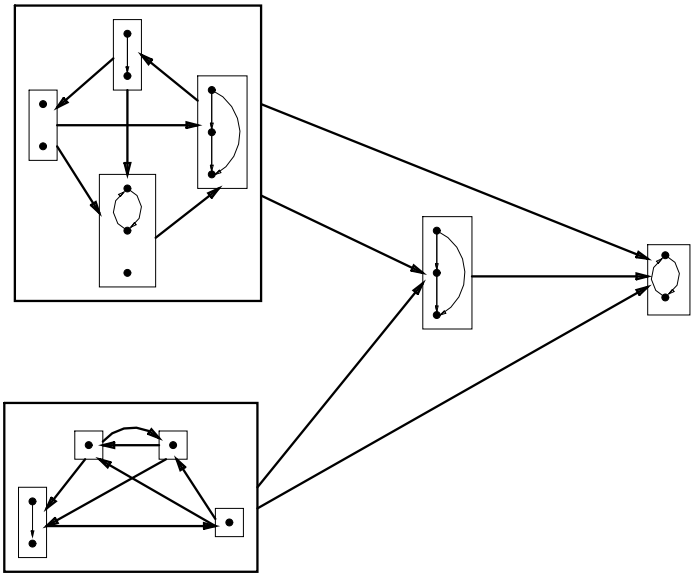


Figure 2.11 A decomposition of a non-strong quasi-transitive digraph. Big arcs between different boxed sets indicate that there is a complete domination in the direction shown.

H_i and H_j, then either $H_i \mapsto H_j$ or $H_j \mapsto H_i$. Now if $H_i \mapsto H_j \mapsto H_k$, then, by quasi-transitivity, $H_i \mapsto H_k$. So by contracting each H_i to a vertex h_i, we get a transitive oriented graph T with vertices $h_1, h_2, \ldots, h_t$. This shows that $D = T[H_1, H_2, \ldots, H_t]$.

Suppose now that D is strong. Let $Q_1, Q_2, \ldots, Q_s$ be the subdigraphs of D such that each $\overline{UG(Q_i)}$ is a connected component of $\overline{UG(D)}$. According to Lemma 2.7.4(a), each Q_i is either non-strong or just a single vertex. By Lemma 2.7.4(b) we obtain a strong semicomplete digraph S if each Q_i is contracted to a vertex. This shows that $D = S[Q_1, Q_2, \ldots, Q_s]$. □

2.8 Path-Mergeable Digraphs

A digraph D is **path-mergeable**, if for any choice of vertices $x, y \in V(D)$ and any pair of internally disjoint (x, y)-paths P, Q, there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. We will see, in several places of this book, that the notion of a path-mergeable digraph is very useful for design of algorithms and proofs of theorems. This makes it worthwhile studying path-mergeable digraphs. The results presented in this section are adapted from [72], where the study of path-mergeable digraphs was initiated by Bang-Jensen.

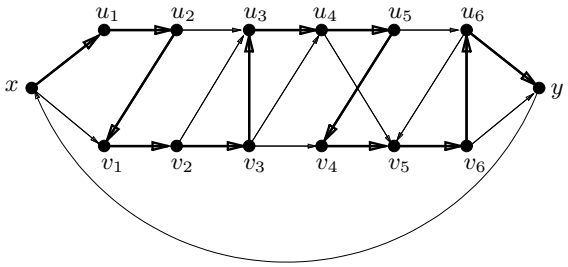


Figure 2.12 A digraph which is path-mergeable. The fat arcs indicate the path $xu_1u_2v_1v_2v_3u_3u_4u_5v_4v_5v_6u_6y$ from x to y which is obtained by merging the two (x,y)-paths $xu_1u_2u_3u_4u_5u_6y$ and $xv_1v_2v_3v_4v_5v_6y$.

We prove a characterization of path-mergeable digraphs, which implies that path-mergeable digraphs can be recognized efficiently.

Theorem 2.8.1 *A digraph D is path-mergeable if and only if for every pair of distinct vertices $x, y \in V(D)$ and every pair $P = xx_1 \dots x_r y$, $P' = xy_1 \dots y_s y$, $r, s \geq 1$ of internally disjoint (x,y)-paths in D, either there exists an $i \in \{1, \dots, r\}$, such that $x_i \rightarrow y_1$, or there exists a $j \in [s]$, such that $y_j \rightarrow x_1$.*

Proof: We prove 'only if' by induction on $r + s$. It is obvious for $r = s = 1$, so suppose that $r + s \geq 3$. If there is no arc between $\{x_1, \dots, x_r\}$ and $\{y_1, \dots, y_s\}$, then clearly P, P' cannot be merged into one path. Hence we may assume without loss of generality that there is an arc $x_i y_j$ for some i, j, $1 \leq i \leq r, 1 \leq j \leq s$. If $j = 1$, then the claim follows. Otherwise apply induction to the paths $P[x, x_i]y_j$, $xP'[y_1, y_j]$.

The proof of 'if' is left to the reader. It is similar to the proof of Proposition 2.8.3 below. □

The proof of the following result is left as Exercise 2.24.

Corollary 2.8.2 *Path-mergeable digraphs can be recognized in polynomial time.* □

The next result shows that if a digraph is path-mergeable, then the merging of paths can always be done in a particularly nice way.

Proposition 2.8.3 *Let D be a digraph which is path-mergeable and let $P = xx_1 \dots x_r y$, $P' = xy_1 \dots y_s y$, $r, s \geq 0$ be internally disjoint (x,y)-paths in D. The paths P and P' can be merged into one (x,y)-path P^* such that vertices from P (respectively, P') remain in the same order as on that path. Furthermore the merging can be done in at most $2(r + s)$ steps.*

Proof: We prove the result by induction on $r + s$. It is obvious if $r = 0$ or $s = 0$, so suppose that $r, s \geq 1$. By Theorem 2.8.1 there exists an i such that

either $x_i \to y_1$ or $y_i \to x_1$. By scanning both paths forward one arc at a time, we can find i in at most $2i$ steps; suppose without loss of generality $x_i \to y_1$. By applying the induction hypothesis to the paths $P[x_i, x_r]y$ and $x_i P'[y_1, y_s]y$, we see that we can merge them into a single path Q in the required order-preserving way in at most $2(r+s-i)$ steps. The required path P^* is obtained by concatenating the paths $xP[x_1, x_i]$ and Q, and we have found it in at most $2(r+s)$ steps, as required. □

2.9 Locally In/Out-Semicomplete Digraphs

A digraph D is **locally in-semicomplete** (**locally out-semicomplete**) if, for every vertex x of D, the in-neighbours (out-neighbours) of x induce a semicomplete digraph. Clearly, the converse of a locally in-semicomplete digraph is a locally out-semicomplete digraph and vice versa. A digraph D is **locally semicomplete** if it is both locally in- and locally out-semicomplete. See Figure 2.13. Clearly every semicomplete digraph is locally semicomplete. A locally in-semicomplete digraph with no 2-cycle is a **locally in-tournament digraph**. Similarly, one can define **locally out-tournament digraphs** and **locally tournament digraphs**. For convenience, we will sometimes refer to locally tournament digraphs as **local tournaments** and to locally in-tournament (out-tournament) digraphs as **local in-tournaments** (**local out-tournaments**).

(*a*) (*b*)

Figure 2.13 (a) A locally out-semicomplete digraph which is not locally in-semicomplete; (b) a locally semicomplete digraph.

Proposition 2.9.1 by Bang-Jensen shows that locally in-semicomplete and locally out-semicomplete digraphs form subclasses of the class of path-mergeable digraphs. In particular, this means that every tournament is path-mergeable. In many theorems and algorithms on tournaments this property is of essential use. In some other cases, the very use of this property allows one to simplify proofs of results on tournaments and their generalizations or speed up algorithms on those digraphs.

Proposition 2.9.1 *[72] Every locally in-semicomplete (out-semicomplete) digraph is path-mergeable.*

Proof: Let D be a locally out-semicomplete digraph and let $P = y_1y_2 \dots y_k$, $Q = z_1z_2 \dots z_t$ be a pair of internally disjoint (x, y)-paths (i.e., $y_1 = z_1 = x$ and $y_k = z_t = y$). We show that there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. Our claim is trivially true when $|A(P)| + |A(Q)| = 3$. Assume now that $|A(P)| + |A(Q)| \geq 4$. Since D is out-semicomplete, either $y_2 \rightarrow z_2$ or $z_2 \rightarrow y_2$ (or both) and the claim follows from Theorem 2.8.1.

The proposition holds for locally in-semicomplete digraphs as they are the converses of locally out-semicomplete digraphs. □

The path-mergeability can be generalized in a natural way as follows. A digraph D is **in-path-mergeable** if, for every vertex $y \in V(D)$ and every pair P, Q of internally disjoint paths with common terminal vertex y, there is a path R such that $V(R) = V(P) \cup V(Q)$, the path R terminates at y and starts at a vertex which is the initial vertex of either P or Q (or, possibly, both). Observe that, in this definition, the initial vertices of paths P and Q may coincide. Therefore, every in-path-mergeable digraph is path-mergeable. However, it is easy to see that not every path-mergeable digraph is in-path-mergeable (see Exercise 2.21). A digraph D is **out-path-mergeable** if the converse of D is in-path-mergeable. Clearly, every in-path-mergeable (out-path-mergeable) digraph is locally in-semicomplete (locally out-semicomplete). The converse is also true (hence this is another way of characterizing locally in-semicomplete digraphs). The proof of Proposition 2.9.2 is left as Exercise 2.25.

Proposition 2.9.2 *Every locally in-semicomplete (out-semicomplete, respectively) digraph is in-path-mergeable (out-path-mergeable, respectively).* □

Some simple, yet very useful, properties of locally in-semicomplete digraphs are described in the following results (in [105], by Bang-Jensen, Huang and Prisner, these results were proved for locally tournament digraphs only, so the statements below are their slight generalizations first stated by Bang-Jensen and Gutin [89]). Observe that a locally out-semicomplete digraph, being the converse of a locally in-semicomplete digraph, has similar properties (see Exercise 2.28). The next lemma follows from Proposition 1.7.1 (see [91]).

Lemma 2.9.3 *Every connected locally in-semicomplete digraph D has an out-branching.* □

Theorem 2.9.4 is illustrated in Figure 2.14.

Theorem 2.9.4 *Let D be a locally in-semicomplete digraph.*

(i) Let A and B be distinct strong components of D. If a vertex $a \in A$ dominates some vertex in B, then $a \mapsto B$.
(ii) If D is connected, then $SC(D)$ has an out-branching.

Proof: Let A and B be strong components of D for which there is an arc (a, b) from A to B. Since B is strong, there is a (b', b)-path in B for every $b' \in V(B)$. By the definition of locally in-semicomplete digraphs and the fact that there is no arc from B to A, we can conclude that $a \rightarrow b'$. This proves (i).

Part (ii) follows from the fact that $SC(D)$ is itself a locally in-tournament digraph and Lemma 2.9.3. □

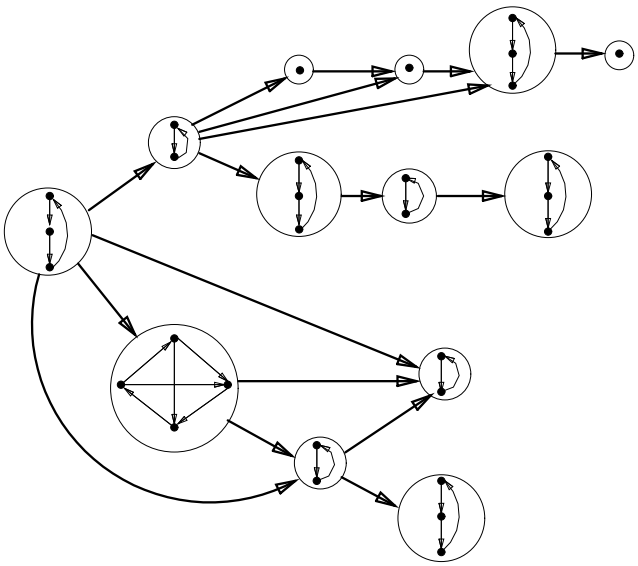


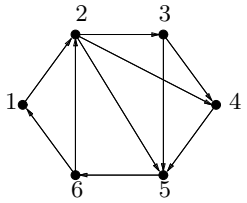
Figure 2.14 The strong decomposition of a non-strong locally in-semicomplete digraph. The big circles indicate strong components and a fat arc from a component A to a component B between two components indicates that there is at least one vertex $a \in A$ such that $a \mapsto B$.

2.10 Locally Semicomplete Digraphs

Locally semicomplete digraphs were introduced in 1990 by Bang-Jensen [66]. As shown in several places in our book, this class of digraphs has many nice properties in common with its proper subclass, semicomplete digraphs. The main aim of this section is to obtain a classification of locally semicomplete digraphs first proved by Bang-Jensen, Guo, Gutin and Volkmann [80]. In the process of deriving this classification, we will show several important properties of locally semicomplete digraphs. We start our consideration from round digraphs, a nice special class of locally semicomplete digraphs.

2.10.1 Round Digraphs

A digraph on n vertices is **round** if we can label its vertices $v_1, v_2, \ldots, v_n$ so that for each i, we have $N^+(v_i) = \{v_{i+1}, \ldots, v_{i+d^+(v_i)}\}$ and $N^-(v_i) = \{v_{i-d^-(v_i)}, \ldots, v_{i-1}\}$ (all subscripts are taken modulo n). We will refer to the ordering $v_1, v_2, \ldots, v_n$ as a **round labelling** of D. See Figure 2.15 for an example of a round digraph. Observe that every strong round digraph D is hamiltonian, since $v_1 v_2 \ldots v_n v_1$ form a hamiltonian cycle, whenever $v_1, v_2, \ldots, v_n$ is a round labelling. Round digraphs form a subclass of locally semicomplete digraphs. We will see below that round digraphs play an important role in the study of locally semicomplete digraphs.



R

Figure 2.15 A round digraph with a round labelling.

Proposition 2.10.1 *[541] Every round digraph is locally semicomplete.*

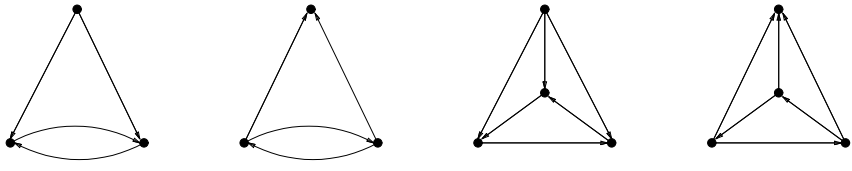
Proof: Let D be a round digraph and let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Consider an arbitrary vertex, say v_i. Let x, y be a pair of out-neighbours of v_i. We show that x and y are adjacent. Assume without loss of generality that v_i, x, y appear in that circular order in the round labelling. Since $v_i \rightarrow y$ and the in-neighbours of y appear consecutively preceding y, we must have $x \rightarrow y$. Thus the out-neighbours of v_i are pairwise adjacent. Similarly, we can show that the in-neighbours of v_i are also pairwise adjacent. Therefore, D is locally semicomplete. □

The main result of this subsection is Theorem 2.10.4 of Huang [541] that gives a characterization of round locally semicomplete digraphs. This characterization generalizes the corresponding characterizations of round local tournaments and tournaments, due to Bang-Jensen [66] and Alspach and Tabib [38], respectively.

An arc xy of a digraph D is **ordinary** if yx is not in D. A cycle or path Q of a digraph D is **ordinary** if all arcs of Q are ordinary.

The following two lemmas due to Huang [541] imply the necessity part of Theorem 2.10.4. A sufficiency proof can be found in [91, 541].

Lemma 2.10.2 *Let D be a round digraph; then the following is true:*



(a) (b) (c) (d)

Figure 2.16 Some forbidden digraphs in Huang's characterization.

(a) Every induced subdigraph of D is round.
(b) None of the digraphs in Figure 2.16 is an induced subdigraph of D.
(c) For each $x \in V(D)$, the subdigraphs induced by $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ are transitive tournaments.

Proof: Exercise 2.31. □

Lemma 2.10.3 *Let D be a round digraph. Then, for each vertex x of D, the subdigraph induced by $N^+(x) \cap N^-(x)$ contains no ordinary cycle.*

Proof: Suppose the subdigraph induced by some $N^+(x) \cap N^-(x)$ contains an ordinary cycle C. Let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Without loss of generality, assume that $x = v_1$. Then C must contain an arc $v_i v_j$ such that $v_j v_i \notin A(D)$ and $i > j$. We have $v_1 \in N^-(v_i)$ but $v_j \notin N^-(v_i)$, contradicting the assumption that $v_1, v_2, \ldots, v_n$ is a round labelling of D. □

Theorem 2.10.4 (Huang) *[541] A connected locally semicomplete digraph D is round if and only if the following holds for each vertex x of D:*

(a) $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ induce transitive tournaments and
(b) $N^+(x) \cap N^-(x)$ induces a (semicomplete) subdigraph containing no ordinary cycle. □

The proof of sufficiency of the conditions of this theorem in [91, 541] can be transformed into a polynomial time algorithm to decide whether a digraph D is round and to find a round labelling of D (if D is round).

Corollary 2.10.5 (Bang-Jensen) *[66] A connected local tournament D is round if and only if, for each vertex x of D, $N^+(x)$ and $N^-(x)$ induce transitive tournaments.* □

2.10.2 Non-Strong Locally Semicomplete Digraphs

The most basic properties of strong components of a connected non-strong locally semicomplete digraph are given in the following result, due to Bang-Jensen.

Theorem 2.10.6 *[66] Let D be a connected locally semicomplete digraph that is not strong. Then the following holds for D.*

(a) If A and B are distinct strong components of D with at least one arc between them, then either $A \mapsto B$ or $B \mapsto A$.
(b) If A and B are strong components of D, such that $A \mapsto B$, then A and B are semicomplete digraphs.
(c) The strong components of D can be ordered in a unique way $D_1, D_2, \ldots, D_p$ such that there are no arcs from D_j to D_i for $j > i$, and D_i dominates D_{i+1} for $i \in [p-1]$.

Proof: Recall that a locally semicomplete digraph is a locally in-semicomplete digraph as well as a locally out-semicomplete digraph. Part (a) of this theorem follows immediately from Part (i) of Theorem 2.9.4 and its analogue for locally out-semicomplete digraphs. Part (b) can be easily obtained from the definition of a locally semicomplete digraph. Finally, Part (c) follows from the fact proved in Theorem 2.9.4 (and its analogue for locally out-semicomplete digraphs) that $SC(D)$ has an out-branching and an in-branching. Indeed, a digraph which is both out-branching and in-branching is merely a hamiltonian path. □

A locally semicomplete digraph D is **round decomposable** if there exists a round local tournament R on $r \geq 2$ vertices such that $D = R[S_1, \ldots, S_r]$, where each S_i is a strong semicomplete digraph. We call $R[S_1, \ldots, S_r]$ a **round decomposition** of D. The following consequence of Theorem 2.10.6, whose proof is left as Exercise 2.32, shows that connected, but not strongly connected locally semicomplete digraphs are round decomposable.

Corollary 2.10.7 *[66] Every connected, but not strongly connected locally semicomplete digraph D has a unique round decomposition $R[D_1, D_2, \ldots, D_p]$, where $D_1, D_2, \ldots, D_p$ is the acyclic ordering of strong components of D and R is the round local tournament containing no cycle which one obtains by taking one vertex from each D_i.* □

Now we describe another kind of decomposition theorem for locally semicomplete digraphs due to Guo and Volkmann. The proof of this theorem is left as Exercise 2.33. The statement of the theorem is illustrated in Figure 2.18.

Theorem 2.10.8 *[440, 442] Let D be a connected locally semicomplete digraph that is not strong and let $D_1, \ldots, D_p$ be the acyclic ordering of strong components of D. Then D can be decomposed into $r \geq 2$ induced subdigraphs $D'_1, D'_2, \ldots, D'_r$ as follows:*

- $D'_1 = D_p, \quad \lambda_1 = p,$
- $\lambda_{i+1} = \min\{\, j \mid N^+(D_j) \cap V(D'_i) \neq \emptyset\}$, *for each* $i \in [r-1]$,

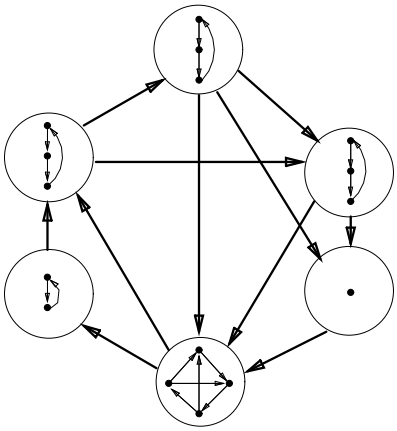


Figure 2.17 A round decomposable locally semicomplete digraph D. The big circles indicate the sets that correspond to the sets $W_1, W_2, \ldots, W_6$ in the decomposition $D = R[W_1, W_2, \ldots, W_6]$, where R is the round locally semicomplete digraph one obtains by replacing each circled set by one vertex. Fat arcs indicate that there is a complete domination in the direction shown.

- $D'_{i+1} = D\langle V(D_{\lambda_{i+1}}) \cup V(D_{\lambda_{i+1}+1}) \cup \cdots \cup V(D_{\lambda_i - 1})\rangle$, *for each* $i \in [r-1]$.

The subdigraphs $D'_1, D'_2, \ldots, D'_r$ *satisfy the properties below:*

(a) D'_i *consists of some strong components of* D *and is semicomplete for each* $i \in [r]$

(b) D'_{i+1} *dominates the initial component of* D'_i *and there exists no arc from* D'_i *to* D'_{i+1} *for any* $i \in [r-1]$

(c) if $r \geq 3$*, then there is no arc between* D'_i *and* D'_j *for* i, j *satisfying* $|j-i| \geq 2$. □

For a connected, but not strongly connected locally semicomplete digraph D, the unique sequence $D'_1, D'_2, \ldots, D'_r$ defined in Theorem 2.10.8 is called the **semicomplete decomposition** of D.

2.10.3 Strong Round Decomposable Locally Semicomplete Digraphs

In the previous subsection we saw that every connected non-strong locally semicomplete digraph is round decomposable. This property does not hold for strong locally semicomplete digraphs (see Lemma 2.10.14). The following assertions, due to Bang-Jensen, Guo, Gutin and Volkmann, provide some important properties concerning round decompositions of strong locally semicomplete digraphs.

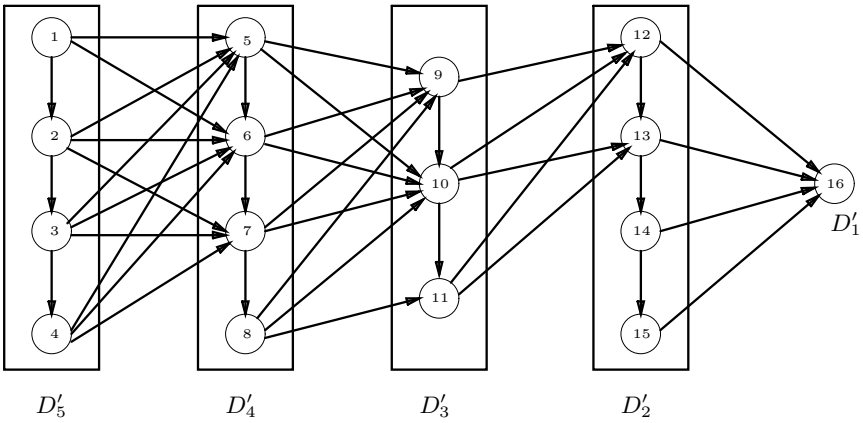


Figure 2.18 The semicomplete decomposition of a non-strong locally semicomplete digraph with 16 strong components (numbered 1-16 corresponding to the acyclic ordering). Each circle indicates a strong component and each box indicates a semicomplete subdigraph formed by consecutive components all of which dominate the first component in the previous layer. For clarity arcs inside components as well as some arcs between components inside a semicomplete subdigraph D'_i (all going from top to bottom) are omitted.

Proposition 2.10.9 *[80] Let $R[H_1, H_2, \ldots, H_\alpha]$ be a round decomposition of a strong locally semicomplete digraph D. Then, for every minimal separating set S, there are two integers i and $k \geq 0$ such that $S = V(H_i) \cup \ldots \cup V(H_{i+k})$.*

Proof: We will first prove that

$$\text{if } V(H_i) \cap S \neq \emptyset, \text{ then } V(H_i) \subseteq S. \tag{2.3}$$

Assume that there exists H_i such that $V(H_i) \cap S \neq \emptyset \neq V(H_i) - S$. Using this assumption we shall prove that $D - S$ is strong, contradicting the definition of S.

Let $s' \in V(H_i) \cap S$. To show that $D - S$ is strong, we consider a pair of different vertices x and y of $D - S$ and prove that $D - S$ has an (x, y)-path. Since S is a minimal separating set, $D' = D - (S - s')$ is strong. Consider a shortest (x, y)-path P in D' among all (x, y)-paths using at most two vertices from each H_j. The existence of such a path follows from the fact that R is strong. Since the vertices of H_i in D' have the same in- and out-neighbourhoods, P contains at most one vertex from H_i, unless $x, y \in V(H_i)$ in which case P contains only these two vertices from H_i. If s' is not on P, we are done. Thus, assume that s' is on P. Then, since P is shortest possible, neither x nor y belongs to H_i. Now we can replace s' with a vertex in $V(H_i) - S$. Therefore, $D - S$ has an (x, y)-path, so (2.3) is proved.

Suppose that S consists of disjoint sets $T_1, \ldots, T_\ell$ such that

$$T_i = V(H_{j_i}) \cup \ldots \cup V(H_{j_i+k_i}) \quad \text{and} \quad (V(H_{j_i-1}) \cup V(H_{j_i+k_i+1})) \cap S = \emptyset$$

for $i \in [\ell]$. If $\ell \geq 2$, then $D - T_i$ is strong and hence it follows from the fact that R is round that H_{j_i-1} dominates $H_{j_i+k_i+1}$ for every $i \in [\ell]$. Therefore, $D - S$ is strong; a contradiction. □

Corollary 2.10.10 *[80] If a locally semicomplete digraph D is round decomposable, then it has a unique round decomposition $D = R[D_1, D_2, \ldots, D_\alpha]$.*

Proof: Suppose that D has two different round decompositions: $D = R[D_1, \ldots, D_\alpha]$ and $D = R'[H_1, \ldots, H_\beta]$.

By Corollary 2.10.7, we may assume that D is strong. By the definition of a round decomposition, this implies that $\alpha, \beta \geq 3$. Let S be a minimal separating set of D. By Proposition 2.10.9, we may assume without loss of generality that $S = V(D_1 \cup \ldots \cup D_i) = V(H_1 \cup \ldots \cup H_j)$ for some i and j. Since $D - S$ is non-strong, by Corollary 2.10.7, $D_{i+1} = H_{j+1}, \ldots, D_\alpha = H_\beta$ (in particular, $\alpha - i = \beta - j$). Now it suffices to prove that

$$D_1 = H_1, \ldots, D_i = H_j \text{ (in particular, } i = j). \tag{2.4}$$

If $D\langle S\rangle$ is non-strong, then (2.4) follows by Corollary 2.10.7. If $D\langle S\rangle$ is strong, then first consider the case $\alpha = 3$. Then $S = V(D_1)$, because $D - S$ is non-strong and $\alpha = 3$. Assuming that $j > 1$, we obtain that the subdigraph of D induced by S has a strong round decomposition. This contradicts the fact that R' is a local tournament, since the in-neighbourhood of the vertex r'_{j+1} in R' contains a cycle (where r'_p corresponds to H_p, $p = 1, \ldots, \beta$). Therefore, (2.4) is true for $\alpha = 3$. If $\alpha > 3$, then we can find a separating set in $D\langle S\rangle$ and conclude by induction that (2.4) holds. □

Proposition 2.10.9 allows us to construct a polynomial algorithm for checking whether a locally semicomplete digraph is round decomposable.

Proposition 2.10.11 *[80] There exists a polynomial algorithm to decide whether a given locally semicomplete digraph D has a round decomposition and to find this decomposition if it exists.*

Proof: We only give a sketch of such an algorithm. Find a minimal separating set S in D starting with $S' = N^+(x)$ for a vertex $x \in V(D)$ and deleting vertices from S' until a minimal separating set is obtained. Construct the strong components of $D\langle S\rangle$ and $D - S$ and label these $D_1, D_2, \ldots, D_\alpha$, where $D_1, \ldots, D_p$, $p \geq 1$, form an acyclic ordering of the strong components of $D\langle S\rangle$ and $D_{p+1}, \ldots, D_\alpha$ form an acyclic ordering of the strong components of $D - S$. For every pair D_i and D_j $(1 \leq i \neq j \leq \alpha)$, we check the following: if there exist some arcs between D_i and D_j, then either $D_i \mapsto D_j$ or $D_j \mapsto D_i$. If we find a pair for which the above condition is false, then D is not round decomposable. Otherwise, we form a digraph $R = D\langle\{x_1, x_2, \ldots, x_\alpha\}\rangle$, where $x_i \in V(D_i)$ for each $i \in [\alpha]$. We check whether R is round using Corollary 2.10.5. If R is not round, then D is not round decomposable. Otherwise, D is round decomposable and $D = R[D_1, \ldots, D_\alpha]$.

It is not difficult to verify that our algorithm is correct and polynomial. □

2.10.4 Classification of Locally Semicomplete Digraphs

We start this subsection with a lemma on minimal separating sets of locally semicomplete digraphs. It will be shown in Lemma 5.8.4 that for a strong locally semicomplete digraph D and a minimal separating set S in D, we have that $D - S$ is connected.

Lemma 2.10.12 *[80] If a strong locally semicomplete digraph D is not semicomplete, then there exists a minimal separating set $S \subset V(D)$ such that $D - S$ is not semicomplete. Furthermore, if $D_1, D_2, \ldots, D_p$ is the acyclic ordering of the strong components of D and $D'_1, D'_2, \ldots, D'_r$ is the semicomplete decomposition of $D - S$, then $r \geq 3$, $D\langle S\rangle$ is semicomplete and we have $D_p \mapsto S \mapsto D_1$.*

Proof: Suppose $D - S$ is semicomplete for every minimal separating set S. Then $D - S$ is semicomplete for all separating sets S. Hence D is semicomplete, because any pair of non-adjacent vertices can be separated by some separating set S. This proves the first claim of the lemma.

Let S be a minimal separating set such that $D - S$ is not semicomplete. Clearly, if $r = 2$ (in Theorem 2.10.8), then $D - S$ is semicomplete. Thus, $r \geq 3$. By the minimality of S every vertex $s \in S$ dominates a vertex in D_1 and is dominated by a vertex in D_p. Thus if some $x \in D_p$ was dominated by $s \in S$, then, by the definition of a locally semicomplete digraph, we would have $D_1 \mapsto D_p$, contradicting the fact that $r \geq 3$. Hence (using that D_p is strongly connected) we get that $D_p \mapsto S$ and similarly $S \mapsto D_1$. From the last observation it follows that S is semicomplete. □

Now we consider strongly connected locally semicomplete digraphs which are not semicomplete and not round decomposable. We first show that the semicomplete decomposition of $D-S$ has exactly three components, whenever S is a minimal separating set such that $D - S$ is not semicomplete.

Lemma 2.10.13 *[80] Let D be a strong locally semicomplete digraph which is not semicomplete. Either D is round decomposable, or D has a minimal separating set S such that the semicomplete decomposition of $D - S$ has exactly three components D'_1, D'_2, D'_3.*

Proof: By Lemma 2.10.12, D has a minimal separating set S such that the semicomplete decomposition of $D - S$ has at least three components.

Assume now that the semicomplete decomposition of $D - S$ has more than three components $D'_1, \ldots, D'_r$ ($r \geq 4$). Let $D_1, D_2, \ldots, D_p$ be the acyclic ordering of strong components of $D - S$. According to Theorem 2.10.8 (c), there is no arc between D'_i and D'_j if $|i - j| \geq 2$. It follows from the definition of a locally semicomplete digraph that

$$N^+(D'_i) \cap S = \emptyset \text{ for } i \geq 3 \text{ and } N^-(D'_j) \cap S = \emptyset \text{ for } j \leq r - 2. \qquad (2.5)$$

By Lemma 2.10.12, $D\langle S\rangle$ is semicomplete and $S = N^+(D_p)$. Let $D_{p+1}, \ldots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S\rangle$. Using (2.5) and the assumption $r \geq 4$, it is easy to check that if there is an arc between D_i and D_j $(1 \leq i \neq j \leq p+q)$, then $D_i \mapsto D_j$ or $D_j \mapsto D_i$. Let $R = D\langle\{x_1, x_2, \ldots, x_{p+q}\}\rangle$ with $x_i \in V(D_i)$ for each $i \in [p+q]$. Now it suffices to prove that R is a round local tournament.

Since R is a subdigraph of D and no pair D_i, D_j induces a strong digraph, we see that R is a local tournament. By Corollary 2.10.7 each of the subdigraphs $R' = R - \{x_{p+1}, \ldots, x_{p+q}\}$, $R'' = R - V(R) \cap V(D'_{r-1})$ and $R''' = R - V(R) \cap V(D'_2)$ is round. Since $N^+(v) \cap V(R)$ (as well as $N^-(v) \cap V(R)$) is completely contained in one of the sets $V(R'), V(R'')$ and $V(R''')$ for every $v \in V(R)$, we see that R is round.

Thus if $r \geq 4$, then D is round decomposable. □

Our next result is a characterization of locally semicomplete digraphs which are not semicomplete and not round decomposable. This characterization was proved for the first time by Guo in [432]. A weaker form was obtained earlier by Bang-Jensen in [71]. Here we give the proof of this result from [80].

Lemma 2.10.14 *Let D be a strong locally semicomplete digraph which is not semicomplete. Then D is not round decomposable if and only if the following conditions are satisfied:*

(a) There is a minimal separating set S such that $D - S$ is not semicomplete and for each such S, $D\langle S\rangle$ is semicomplete and the semicomplete decomposition of $D - S$ has exactly three components D'_1, D'_2, D'_3;

(b) There are integers α, β, μ, ν with $\lambda_2 \leq \alpha \leq \beta \leq p-1$ and $p+1 \leq \mu \leq \nu \leq p+q$ such that

$$N^-(D_\alpha) \cap V(D_\mu) \neq \emptyset \text{ and } N^+(D_\alpha) \cap V(D_\nu) \neq \emptyset,$$

$$\text{or } N^-(D_\mu) \cap V(D_\alpha) \neq \emptyset \text{ and } N^+(D_\mu) \cap V(D_\beta) \neq \emptyset,$$

where $D_1, D_2, \ldots, D_p$ and $D_{p+1}, \ldots, D_{p+q}$ are the acyclic orderings of the strong components of $D - S$ and $D\langle S\rangle$, respectively, and D_{λ_2} is the initial component of D'_2.

Proof: If D is round decomposable and satisfies (a), then we must have $D = R[D_1, D_2, \ldots, D_{p+q}]$, where R is the digraph obtained from D by contracting each D_i into one vertex. This follows from Corollary 2.10.7 and the fact that each of the digraphs $D - S$ and $D - V(D'_2)$ has a round decomposition that agrees with this structure. Now it is easy to see that D does not satisfy (b).

Suppose now that D is not round decomposable. By Lemmas 2.10.12 and 2.10.13, D satisfies (a), so we only have to prove that it also satisfies (b).

If there are no arcs from S to D'_2, then it is easy to see that D has a round decomposition. If there exist components D_{p+i} and D_j with $V(D_j) \subseteq$

$V(D'_2)$, such that there are arcs in both directions between D_{p+i} and D_j, then D satisfies (b). So we can assume that for every pair of sets from the collection $D_1, D_2, \ldots, D_{p+q}$, either there are no arcs between these sets, or one set completely dominates the other. Then, by Corollary 2.10.5, D is round decomposable, with round decomposition $D = R[D_1, D_2, \ldots, D_{p+q}]$ as above, unless we have three subdigraphs $X, Y, Z \in \{D_1, D_2, \ldots, D_{p+q}\}$ such that $X \mapsto Y \mapsto Z \mapsto X$ and there exists a subdigraph $W \in \{D_1, D_2, \ldots, D_{p+q}\} - \{X, Y, Z\}$ such that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$.

One of the subdigraphs X, Y, Z, say without loss of generality X, is a strong component of $D\langle S\rangle$. If we have $V(Y) \subseteq S$ also, then $V(Z) \subseteq V(D'_2)$ and W is either in $D\langle S\rangle$ or in D'_2 (there are four possible positions for W satisfying that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$). In each of these cases it is easy to see that D satisfies (b). For example, if W is in $D\langle S\rangle$ and $W \mapsto X, Y, Z$, then any arc from W to Z and from Z to X satisfies the first part of (b). The proof is similar when $V(Y) \subseteq V(D'_3)$. Hence we can assume that $V(Y) \subseteq V(D'_2)$. If $Z = D_p$, then W must be either in $D\langle S\rangle$ and $X, Y, Z \mapsto W$, or $V(W) \subseteq V(D'_2)$ and $W \mapsto X, Y, Z$ (which means that $W = D_i$ and $Y = D_j$ for some $\lambda_2 \leq i < j < p$). In both cases it is easy to see that D satisfies (b). The last case $V(Y), V(Z) \subseteq V(D'_2)$ can be treated similarly. □

We can now state a classification of locally semicomplete digraphs.

Theorem 2.10.15 (Bang-Jensen, Guo, Gutin, Volkmann) *[80] Let D be a connected locally semicomplete digraph. Then exactly one of the following possibilities holds.*

(a) D is round decomposable with a unique round decomposition given by $D = R[D_1, D_2, \ldots, D_\alpha]$, where R is a round local tournament on $\alpha \geq 2$ vertices and D_i is a strong semicomplete digraph for each $i \in [\alpha]$;
(b) D is not round decomposable and not semicomplete and it has the structure as described in Lemma 2.10.14;
(c) D is a semicomplete digraph which is not round decomposable. □

We finish this section with the following useful proposition, whose proof is left as Exercise 2.36.

Proposition 2.10.16 *[80] Let D be a strong non-round decomposable locally semicomplete digraph and let S be a minimal separating set of D such that $D - S$ is not semicomplete. Let $D_1, \ldots, D_p$ be the acyclic ordering of the strong components of $D - S$ and $D_{p+1}, \ldots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S\rangle$. Suppose that there is an arc $s \to v$ from S to D'_2 with $s \in V(D_i)$ and $v \in V(D_j)$, then*

$$D_i \cup D_{i+1} \cup \ldots \cup D_{p+q} \mapsto D'_3 \mapsto D_{\lambda_2} \cup \ldots \cup D_j.$$ □

2.11 Totally Φ-Decomposable Digraphs

Theorem 2.7.5 is a very important starting point for construction of polynomial algorithms for hamiltonian paths and cycles in quasi-transitive digraphs (see Chapter 6) and solving more general problems in this class of digraphs. This theorem shows that quasi-transitive digraphs are totally Φ-decomposable, where Φ is the union of extended semicomplete and transitive digraphs. Since both extended semicomplete digraphs and transitive digraphs are special subclasses of much wider classes of digraphs, it is natural to study totally Φ-decomposable digraphs, where Φ is a much more general class of digraphs than the union of extended semicomplete and transitive digraphs. However, our choice of candidates for the class Φ should be restricted in such a way that we can still construct polynomial algorithms for some important problems such as the hamiltonian cycle problem using properties of digraphs in Φ.

This idea was first used by Bang-Jensen and Gutin [86] to introduce the following three classes of digraphs:

(a) Φ_0 is the union of all semicomplete multipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs,
(b) Φ_1 is the union of all semicomplete bipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs, and
(c) Φ_2 is the union of all connected extended locally semicomplete digraphs and all acyclic digraphs.

The aim of this section is to show that totally Φ_i-decomposable digraphs can be recognized in polynomial time for $i = 0, 1, 2$. (If these recognition problems were not polynomial, then the study of the properties of totally Φ_i-decomposable digraphs would be of much less interest.)

A set Φ of digraphs is **hereditary** if $D \in \Phi$ implies that every induced subdigraph of D is in Φ. Observe that every Φ_i, $i = 0, 1, 2$, is a hereditary set.

Lemma 2.11.1 *Let Φ be a hereditary set of digraphs. If a given digraph D is totally Φ-decomposable, then every induced subdigraph D' of D is totally Φ-decomposable. In other words, total Φ-decomposability is a hereditary property.*

Proof: By induction on the number of vertices of D. The claim is obviously true if D has less than 3 vertices.

If $D \in \Phi$, then our claim follows from the fact that Φ is hereditary. So we may assume that $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where $R \in \Phi$ and each of $H_1, \ldots, H_r$ is totally Φ-decomposable.

Let D' be an induced subdigraph of D. If there is an index i so that $V(D') \subset V(H_i)$, then D' is totally Φ-decomposable by induction. Otherwise, $D' = R'[T_1, \ldots, T_{r'}]$, where $r' \geq 2$ and $R' \in \Phi$, is the subdigraph of R

induced by those vertices i of R, whose H_i has a non-empty intersection with $V(D')$ and the T_j's are the corresponding H_i's restricted to the vertices of D'. Observe that $R' \in \Phi$, since Φ is hereditary. Moreover, by induction, each T_j is totally Φ-decomposable, hence so is D'. □

Lemma 2.11.2 *There exists an $O(mn + n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either acyclic or semicomplete multipartite or semicomplete bipartite or connected extended locally semicomplete.*

Proof: If D is not connected and $D_1,\ldots,D_c$ are its components, then $D = \overline{K}_c[D_1, \ldots, D_c]$. Hence, in the rest of the proof we may assume that D is connected. We consider the different possibilities for R we are interested in, one by one.

Check whether R can be acyclic: First find the strong components $D_1, \ldots, D_k$ of D. If $k = 1$, then R cannot be acyclic and we can stop verifying that possibility. So suppose $k \geq 2$.

If we find two strong components D_i and D_j such that there is an arc between them but there are non-adjacent vertices $x \in D_i$ and $y \in D_j$, then we replace D_i and D_j by their union. This is justified because D_i and D_j cannot be in different sets H_s and H_t in a possible decomposition. Repeat this step but now check also the possibility for a pair D' and D'' of new 'components' to have arcs between D' and D'' in different directions. In the last case we also replace D' and D'' by their union. Continue this procedure until all remaining sets satisfy that either there is no arc between them, or there are all possible arcs from one to the other. Let $V_1, \ldots, V_r$, $r \geq 1$, denote the distinct vertex sets of the obtained 'components'. If $r = 1$, then we cannot find an acyclic graph as R. Otherwise, $D = R[V_1, \ldots, V_r]$, $r \geq 2$, and we obtain R by taking one vertex from each V_i.

Check whether R can be a semicomplete multipartite digraph: Find the connected components $\overline{G}_1, \ldots, \overline{G}_c$, $c \geq 1$, of the complement of the underlying graph $UG(D)$ of D. If $c = 1$, then R cannot be semicomplete multipartite. So we may assume that $c \geq 2$ below. Let G_j be the subgraph of $UG(D)$ induced by the vertices V_j of the jth component $\overline{G}_j$ of the complement of $UG(D)$. Furthermore, let $G_{j1}, \ldots, G_{jn_j}$, $n_j \geq 1$, be the connected components of G_j. Denote $V_{jk} = V(G_{jk})$.

Starting with the collection $W = \{V_1, \ldots, V_c\}$, we identify two of the sets V_i and V_j if there exist V_{ia} and V_{jb} $a \in [n_i]$, $b \in [n_j]$ such that we have none of the possibilities $V_{ia} \mapsto V_{jb}$, $V_{jb} \mapsto V_{ia}$ or $V_{ia} \to V_{jb}$ and $V_{jb} \to V_{ia}$. Clearly the obtained set $V_i \cup V_j$ induces a connected subdigraph of D. Let $Q_1, \ldots, Q_r$ denote the sets obtained, by repeating this process until no more changes occur. If $r = 1$, then R cannot be semicomplete multipartite. Otherwise, R is the semicomplete multipartite digraph obtained by set-contracting each connected component of Q_i into a vertex.

Checking whether R can be a semicomplete bipartite digraph or a connected extended locally semicomplete digraph is left as Exercise 2.39.

It is not difficult to see that, for every R being either acyclic or semicomplete multipartite, the procedures above can be realized as an $O(nm + n^2)$-algorithm. The same complexity is proved for semicomplete bipartite digraphs and extended locally semicomplete digraphs in Exercise 2.39. □

Theorem 2.11.3 *[86] There exists an $O(n^2m+n^3)$-algorithm for checking if a digraph with n vertices and m arcs is totally Φ_i-decomposable for $i = 0, 1, 2$.*

Proof: We describe a recursive algorithm to check Φ_i-decomposability. We have shown in Lemma 2.11.2 how to verify whether $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where R is acyclic, semicomplete multipartite, semicomplete bipartite or connected extended locally semicomplete. Whenever we find an R that could be used, the algorithm checks total Φ_i-decomposability of $H_1, \ldots, H_r$ in recursive calls.

Notice how the algorithm exploits the fact that total Φ_i-decomposability is a hereditary property (see Lemma 2.11.1): if some R is found appropriate, then R can be used, because if D is totally Φ_i-decomposable, then each of $H_1, \ldots, H_r$ (being an induced subdigraph of D) must also be totally Φ_i-decomposable. Since there are $O(n)$ recursive calls, the complexity of the algorithm is $O(n^2m + n^3)$. □

2.12 Planar Digraphs

We now discuss planar (di)graphs, i.e., (di)graphs that can be drawn without crossings between (arcs) edges (except at endpoints). Clearly this property does not depend on the orientation of the arcs and hence we can ignore the orientation below when we give a formal definition. Furthermore, most of the results and definitions in this section are for undirected graphs, but are valid also for planar digraphs as far as their underlying graphs are concerned.

An undirected graph $G = (V, E)$ is **planar** if there exists a mapping f which maps G to $\mathbb{R}^2$ in the following way:

- Each vertex is mapped to a point in $\mathbb{R}^2$ and distinct vertices are mapped to distinct points.
- Each edge $uv \in E$ is mapped to a simple (that is, not self-intersecting) curve C_{uv} from $f(u)$ to $f(v)$ and no two curves corresponding to distinct edges intersect, except possibly at their endpoints.

For algorithmic purposes as well as for arguing about planar graphs, it is inconvenient to allow arbitrary curves in the embeddings of planar graphs. A **polygonal curve** from u to v is a piecewise linear curve consisting of finitely many lines such that the first line starts at u, the last line ends at v and each other line starts at the last point of the previous line. Since we can

approximate any simple curve arbitrarily well by a polygonal curve we may assume that the curves used in the embedding are always polygonal curves.

A planar graph G may have many different embeddings in the plane (each embedding corresponds to a mapping f as above). Sometimes we wish to refer to properties of a specific embedding f of G. In this case we say that G is **plane** (that is, already embedded) with planar embedding f. A plane graph G partitions $\mathbb{R}^2$ into a finite number of (topologically) connected regions called **faces**. Precisely one of these faces is unbounded and we call this the **outer face**. It is easy to see that, for any fixed face F of G, we may re-embed G in $\mathbb{R}^2$ in such a way that F becomes the outer face. The boundary of a face F is denoted by $bd(F)$ and we normally describe a face by listing the vertices in clockwise order around the face (for the unbounded face this corresponds to listing the vertices on the boundary in the anti-clockwise order). See Figure 2.19 for an illustration of the definitions.

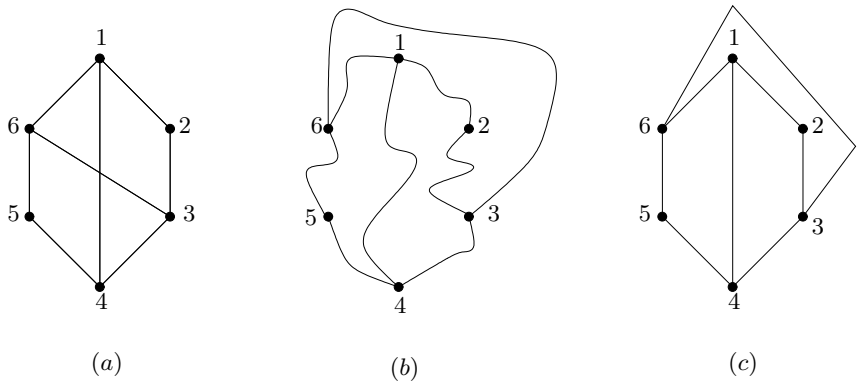


Figure 2.19 (a) shows a non-planar embedding of a graph H; (b) shows a planar embedding of H; (c) shows a planar embedding of H where all curves are polygonal. With respect to the embedding in (c), the faces are 12341, 14561, 16321 and 36543. The outer face is 36543.

Observe that if we add the edge 25 to the graph H in Figure 2.19, then the resulting graph, which is isomorphic to $K_{3,3}$, is no longer planar. In fact, planar graphs have a famous characterization, due to Kuratowski:

Theorem 2.12.1 (Kuratowski's theorem) *[632] A graph has a planar embedding if and only if it does not contain a subdivision*[3] *of K_5 or $K_{3,3}$.* □

Based on this it is possible to show that planar graphs (and hence also planar digraphs) can be recognized efficiently. In fact, Hopcroft and Tarjan

[3] A **subdivision** H' of a graph H is any graph that can be obtained from H by replacing each edge by a path all of whose internal vertices have degree 2 in H'.

[534] showed that it can be done in linear time and if the graph is planar, one can find a planar embedding in the same time.

The following relation between the number of vertices, edges and faces in a plane graph, known as **Euler's formula**, is easy to prove by induction on the number of faces.

Theorem 2.12.2 *If G is a connected plane graph on n vertices and m edges, then $n - m + \phi = 2$, where ϕ denotes the number of faces in the embedding on G. In particular, the number of faces is the same in every embedding of G.* □

We leave it to the reader to derive the following easy consequence of Theorem 2.12.2 (see Exercise 2.46):

Corollary 2.12.3 *For every planar graph on $n \geq 3$ vertices and m edges we have $m \leq 3n - 6$.* □

If we allow multiple edges, then we cannot bound the number of edges as we did above. However, for planar digraphs we have the following easy consequence:

Corollary 2.12.4 *No planar digraph on $n \geq 3$ vertices has more than $6n-12$ arcs.* □

We finish this section by a conjecture of Neumann-Lara first posed in 1982 [724] that links planar digraphs with acyclic digraphs.

Conjecture 2.12.5 *The vertices of every planar digraph can be partitioned into two sets such that each set induces an acyclic digraph.*

2.13 Digraphs of Bounded Width

The tree-width is one of the most important parameters in the area of undirected graphs [573]. It is a cornerstone of the Graph Minors Theory, it is used to prove theorems in structural graph theory, and it has many algorithmic applications due to the fact that many $\mathcal{NP}$-hard problems can be solved in linear time when restricted to graphs of bounded tree-width [573]. Naturally, researchers tried to extend the notion of tree-width to digraphs. In particular, Johnson, Robertson, Seymour and Thomas [573] introduced and studied the notion of the directed tree-width, and Berwanger, Dawar, Hunter and Kreutzer [154] and Obdržálek [731] came up with the notion of DAG-width. There are several other directed width parameters, for example, Kelly-with introduced by Hunter and Kreutzer [544].

While the authors of [154, 544, 573, 731] managed to obtain some 'positive' algorithmic results on digraphs of bounded directed tree-width, DAG-width

and Kelly-width similar to those on undirected graphs with bounded tree-width, there are several 'negative' complexity results obtained by Dankelmann, Gutin and Kim [241] and Kreutzer and Ordyniak [626] indicating that the directed width parameters are of somewhat lesser interest than the tree-width.

In the first subsection of this section we consider digraphs of bounded tree-width and, in the second subsection, we study digraphs in which directed width parameters are bounded.

2.13.1 Digraphs of Bounded Tree-Width

To illustrate the usefulness of tree-width, we will show that one can find, in a linear time, a minimum size kernel[4] in a digraph whose underlying graph is bounded by a constant tree-width. This result allows us to prove that, in a planar digraph D of order n, one can check, in polynomial time, whether D has a kernel of size $O(\log^2 n)$, and if D has such a kernel, then to find one of minimal size.

A non-trivial use of the tree-width is given by Alon, Fomin, Gutin, Krivelevich and Saurabh [21, 22] who proved fixed-parameter tractability of the problem of verifying whether a digraph contains, as a subdigraph, an out-tree with at least k leaves, i.e., vertices of in-degree zero (for the definition of fixed-parameter tractability, see Section 18.4). A refinement of the approach in [21] allowed Bonsma and Dorn [173, 174] to prove fixed-parameter tractability of the problem of verifying whether a digraph has an out-branching with at least k leaves. Another application of tree-width can be found in [472], where Gutin, Razgon and Kim proved that the problem of checking whether a digraph has an out-branching with at least k non-leaves is also fixed-parameter tractable.

A **tree decomposition** of an (undirected) graph G is a pair (S, T) where T is a tree whose vertices we will call **nodes** and $S = \{S_i : i \in V(T)\}$ is a collection of subsets of $V(G)$ (called **bags**) such that

1. $\bigcup_{i \in V(T)} S_i = V(G)$,
2. for each edge $\{v, w\} \in E(G)$, there is an $i \in V(T)$ such that $v, w \in S_i$, and
3. for each $v \in V(G)$ the set of nodes $\{i : v \in S_i\}$ forms a subtree of T.

The **width** of a tree decomposition $(\{S_i : i \in V(T)\}, T)$ is defined as the number $\max_{i \in V(T)}\{|S_i| - 1\}$. The **tree-width** of a graph G ($\mathrm{tw}(G)$) is the minimum width over all tree decompositions of G. The **tree-width** of a digraph D ($\mathrm{tw}(D)$) is the tree-width of its underlying graph.

It is not difficult to see that a connected digraph D is of tree-width one if and only if D is a biorientation of a tree (Exercise 2.47). An undirected

[4] A set S of vertices of a digraph D is a **kernel** if S is an independent set and for each $x \in V(D) - S$ there is an out-neighbour in S. For more information on kernels, see Section 3.8.

graph G is called **series-parallel** if there is an ASP digraph D such that $G = UG(D)$. It is well-known (see, e.g., [253] by de Fluiter and Bodlaender) that an undirected graph G has tree-width at most two if and only if each block of G is series-parallel (a **block** of a graph G is a maximal connected subgraph H of G such that $H - x$ is connected for every $x \in V(H)$).

There are several characterizations of undirected graphs of tree-width at most k [601]. We will describe one of the most intuitive such characterizations. A graph G is **chordal**, if every cycle in G of length at least four has a chord, i.e., there is an edge connecting two non-consecutive vertices in the cycle. A **triangulation** of a graph G is a spanning supergraph of G which is a chordal graph.

Theorem 2.13.1 *Let G be a graph with more than k vertices. The graph G is of tree-width at most k if and only if G has a triangulation whose maximum clique has at most $k+1$ vertices.* □

To facilitate our description below we make use of a **nice tree decomposition** (see, e.g., [601] by Kloks). In a nice tree decomposition, we have a binary rooted tree T, i.e., T is a rooted tree such that every node has at most two children. The nodes of T are of four types:

- *An insert node i.* The node i in T has only one child j and there is a vertex $x \in V$ not in S_j such that $S_i = S_j \cup \{x\}$.
- *A forget node i.* The node i in T has only one child j and there is a vertex $x \in V$ not in S_i such that $S_j = S_i \cup \{x\}$.
- *A join node i* has two children p and q. The bags S_i, S_p and S_q are exactly the same.
- *A leaf node i* is simply a leaf of T.

It is not hard to transform a tree decomposition of G into a nice tree decomposition. In fact, the following holds.

Lemma 2.13.2 *[601] Given a tree decomposition of a graph G with n vertices that has width k and $O(n)$ nodes, we can find a nice tree decomposition of G that also has width k and $O(n)$ nodes in time $O(n)$.* □

We will use Lemma 2.13.2 in the following result by Gutin, Kloks, Lee and Yeo [466].

Theorem 2.13.3 *Let D be a digraph of order n. Let the underlying graph G of D have a tree decomposition with $O(n)$ nodes and of width at most t. Then, in $O(n4^t)$ time, we can either find a minimum size kernel in D or determine that D has no kernel.*

Proof: By Lemma 2.13.2, G has a nice tree decomposition with $O(n)$ nodes and of width at most t. Let $(T, \mathcal{S})$ be such a nice tree decomposition of G. Let $S_1, S_2, \ldots, S_r$ be the bags of the tree decomposition (i.e., the nodes of T

are $1, 2, \ldots, r$). Let *root* denote the root node of T. Recall that every vertex (and arc) in D lies in at least one of the bags.

Let Y_i denote the union of the bags S_j of the subtree of T with root node i. For every i, consider a partition (K_i, MC_i, DC_i) of S_i (the three sets of a partition are disjoint and every vertex of S_i is in one of the sets). A (K_i, MC_i, DC_i)-**kernel** is an independent set Q in D such that $K_i \subseteq Q \subseteq Y_i$, $(DC_i \cup MC_i) \cap Q = \emptyset$ and every vertex in $Y_i - DC_i$ either lies in Q or has an out-neighbor in Q[5].

The vertices in DC_i may have an out-neighbor in Q, or not. Since $(DC_i \cup MC_i) \cap Q = \emptyset$, every vertex in MC_i has an out-neighbor in Q. We define $\kappa_i(K_i, MC_i, DC_i)$ as the minimal size of a (K_i, MC_i, DC_i)-kernel, if one exists. If it does not exist, then $\kappa_i(K_i, MC_i, DC_i) = \infty$.

If we can compute $\kappa_i(K_i, MC_i, DC_i)$ for all partitions (K_i, MC_i, DC_i) and all i, then

$$\mu = \min\{\kappa_{root}(K, S_{root} - K, \emptyset) : \ K \subseteq S_{root}\} \tag{2.6}$$

gives us the size of a minimum size kernel in D.

Let i be a node of T. We show how to compute, in time $O(4^t)$, all possible $\kappa_i(K_i, MC_i, DC_i)$. In fact we can also compute the actual minimum (K_i, MC_i, DC_i)-kernels, for all possible partitions (K_i, MC_i, DC_i) in $O(4^t)$ time, but we will leave the details of this to the reader. This will imply the desired complexity above as T has $O(n)$ vertices. We consider the cases when i is a leaf, i has one child and i has two children, separately. We assume that if i does have some children, then all κ_i's are known for these children. We will for each step argue that we find the correct values.

Case 1: i is a leaf. There are $O(3^{|S_i|})$ distinct partitions (K_i, MC_i, DC_i), and we can easily find all of these in $O(|S_i|3^{|S_i|})$ time. For each partition (K_i, MC_i, DC_i) we can check whether K_i is an independent set and every vertex in MC_i has an out-neighbor in K_i in time $O(|S_i|^2)$. If the outcomes of both checks are positive, we have $\kappa_i(K_i, MC_i, DC_i) = |K_i|$. Otherwise, we have $\kappa_i(K_i, MC_i, DC_i) = \infty$. This gives us a time complexity of $O(|S_i|3^{|S_i|} + |S_i|^2 3^{|S_i|}) \subseteq O(4^{|S_i|}) \subseteq O(4^t)$ (recall that $|S_i| \leq t + 1$).

Case 2: i has one child. Let j be the child of i in T. By the definition of a nice tree decomposition, S_j and S_i are identical, except for one vertex, say x, which lies in either S_i or S_j. We consider the following cases.

If $x \in K_i$, then if x is adjacent to a vertex in K_i, then $\kappa_i(K_i, MC_i, DC_i) = \infty$. Otherwise set $DC_j = DC_i \cup N^-(x)$, $MC_j = MC_i - N^-(x)$ and $K_j = K_i - x$. Clearly $\kappa_i(K_i, MC_i, DC_i) = 1 + \kappa_j(K_j, MC_j, DC_j)$ now holds.

If $x \in MC_i$ and x has no out-neighbor in K_i, then $\kappa_i(K_i, MC_i, DC_i) = \infty$.

If $x \in DC_i$ or $x \in MC_i$ and x has an out-neighbor in K_i, then we have $\kappa_i(K_i, MC_i, DC_i) = \kappa_j(K_i, MC_i - x, DC_i - x)$.

If $x \in S_j$, then we have the following:

[5] *MC* and *DC* stand for Must Cover and Don't Care if a vertex from the set has an out-neighbor in the kernel.

$$\kappa_i(K_i, MC_i, DC_i) = \min\{\kappa_j(K_i \cup \{x\}, MC_i, DC_i), \kappa_j(K_i, MC_i \cup \{x\}, DC_i)\}.$$

As all the above cases can be considered in $O(|S_i|)$ time, we get the time complexity $O(|S_i|3^{|S_i|}) = O(4^t)$ for computing κ_i's for all possible partitions.

Case 3: i has two children. Let j and l be the two children, and recall that $S_i = S_j = S_l$. It is not difficult to see that $\kappa_i(K_i, MC_i, DC_i)$ is equal to the minimum value of $\kappa_j(K_i, W, MC_i \cup DC_i - W) + \kappa_l(K_i, MC_i - W, DC_i \cup W) - |K_i|$, over all $W \subseteq MC_i$. The above can be done in $O(2^{|MC_i|})$ time and there are $\binom{|S_i|}{m}2^{|S_i|-m}$ partitions (K_i, MC_i, DC_i) with $|MC_i| = m$. Thus, we can compute κ_i's for all possible partitions of S_i in time $O(\sum_{m=0}^{|S_i|} 2^m\binom{|S_i|}{m}2^{|S_i|-m}) = O(4^t)$.

Since each $\kappa_i(K_i, MC_i, DC_i)$ is computed correctly above, we note that our algorithm will return the correct value of μ in (2.6). If we remember a minimum (K_i, MC_i, DC_i)-kernel for every possible i and partition (K_i, MC_i, DC_i), then our algorithm can in fact return the minimum-sized kernel, and not only its size. Certainly, if $\mu = \infty$, D has no kernel. □

A set S of vertices of an undirected graph G is called **dominating** if for every $x \in V(G) \setminus S$ there is a vertex $s \in S$ adjacent to x. The following result was obtained by Fomin and Thilikos [328].

Theorem 2.13.4 *Let G be a planar graph with n vertices. There is an $O(n^4)$-time algorithm that either constructs a tree decomposition of G with $O(n)$ nodes and of width at most $9.55\sqrt{k}$, or determines that G has no dominating set of size at most k.* □

Observe that every kernel in a digraph D is a dominating set in $UG(D)$. This observation and Theorems 2.13.3 and 2.13.4 imply the following:

Theorem 2.13.5 *[466] Let D be a planar digraph of order n. There is an $O(n2^{19.1\sqrt{k}} + n^4)$-time algorithm that checks whether D has a kernel of size at most k. Moreover, the algorithm finds a minimum size kernel in D, if D has a kernel of size at most k.* □

Theorem 2.13.5 implies that the problem to verify whether a planar graph has a kernel with at most k vertices is fixed-parameter tractable.

Corollary 2.13.6 *[466] Let D be a planar digraph of order n. In polynomial time, one can check whether D has a kernel of size $O(\log^2 n)$, and if D has such a kernel, then find one of minimal size.* □

We conclude this subsection by briefly considering the complexity of checking whether $\mathrm{tw}(G) \leq k$ for a graph G. Unfortunately, the problem is $\mathcal{NP}$-complete, but it is fixed-parameter tractable, and, provided, k is fixed, there is a linear time algorithm for the problem (see [161, 601]).

2.13.2 Digraphs of Bounded Directed Widths

In this subsection, we consider three of the most studied directed width parameters: DAG-widths, directed path-widths and directed tree-width. We will start from the notion of DAG-width rather than that of directed tree-width as the former seems easier to understand than the latter.

A **DAG-decomposition (DAGD)** of a digraph D is a pair (H, χ) where H is an acyclic digraph and $\chi = \{W_h : h \in V(H)\}$ is a family of subsets of $V(D)$ satisfying the following three properties: (i) $V(D) = \bigcup_{h \in V(H)} W_h$, (ii) for all $h, h', h'' \in V(H)$, if h' lies on a directed path from h to h'', then $W_h \cap W_{h''} \subseteq W_{h'}$, and (iii) if $(u, v) \in A(D)$, then there exist $h_1, h_2 \in V(H)$ (it is possible that $h_1 = h_2$) such that $u \in W_{h_1}$, $v \in W_{h_2}$ and there is a directed (h_1, h_2)-path in H. The **width** of a DAGD (H, χ) is $\max_{h \in V(H)} |W_h| - 1$. The **DAG-width** of a digraph D ($\mathrm{dagw}(D)$) is the minimum width over all possible DAGDs of D.

A **directed path decomposition (DPD)** is a special case of DAGD when H is a directed path. The **directed path-width** of a digraph D ($\mathrm{dpw}(D)$) is defined as the DAG-width above, but DAGDs are replaced by DPDs.

The following notion of vertex separation allows one to evaluate the directed path-width of a digraph without constructing any DPD. Let D be a digraph and let $\pi = (v_1, v_2, \ldots, v_n)$ be an ordering of $V(D)$. We define $V_j = \{v_i : 1 \leq j \leq i\}$ and $\partial V_j = \{v_i \in V_j : (x, v_i) \in A(D)$ for some $x \in V(D) \setminus V_j\}$. With the **vertex separation** of D **with respect to** π given as $\mathrm{vs}_\pi(D) = \max_j |\partial V_j|$, the **vertex separation** of D is defined as $\mathrm{vs}(D) = \min\{\mathrm{vs}_\pi(D) : \pi$ is an ordering of $V(D)\}$.

It is well-known that, for undirected graphs, the path-width equals to the vertex separation (see Kirousis and Papadimitriou [597]). We extend this result to digraphs.

Theorem 2.13.7 *For any digraph D,* $\mathrm{vs}(D) = \mathrm{dpw}(D)$.

Proof: Let $\pi = (v_1, v_2, \ldots, v_n)$ be an ordering of $V(D)$ and suppose $\mathrm{vs}_\pi(D) = k$. We will prove that $\mathrm{dpw}(D) \leq k$. Set $W_i = \{v_i\} \cup \partial V_{i-1}$ for $i \geq 2$ and $W_1 = \{v_1\}$. We claim that $(12 \ldots n, \chi)$, where $\chi = \{W_1, W_2, \ldots, W_n\}$, is a DPD of width k.

Obviously the property (i) of DPD is satisfied. To check the property (ii), let us choose an arbitrary vertex $v_i \in V(G)$ and see whether the sets W_j containing v_i appear in a row. By the construction of W_j's, the vertex v_i appears in the set W_i and does not appear in any W_j with $j < i$. If there is no backward arc entering v_i, this set is the only one containing v_i and there is nothing to prove. Otherwise let $(v_{i'}, v_i) \in A(D)$ is a backward arc and let i' be the maximum such index. Observe that W_i and W_j for $i < j \leq i'$ contain v_i and in fact no other set contains v_i. To check the last property (iii), it is enough to see that both end-vertices of every backward

arc $(v_i, v_j) \in A(D)$ are in W_i. It remains to observe that $|W_j| \leq k+1$, which implies that $\mathrm{dpw}(D) \leq k$.

For the converse, let $(12 \ldots l, \chi)$, where $\chi = \{W_1, W_2, \ldots, W_l\}$, be a DPD of width k. Without loss of generality we may assume that these sets are all distinct. Let $X_1 = W_1$ and $X_i = W_i \setminus W_{i-1}$ for each $i \geq 2$. Order the vertices of $V(D)$ as follows. We begin with the empty ordering (the 0-th iteration). At the j-th iteration $(1 \leq j \leq l)$ we add a permutation of X_j to the end of the previous iteration ordering. Suppose we have performed all l iterations and obtained an ordering $\pi = (v_1, v_2, \ldots, v_n)$. We will prove that $\mathrm{vs}_\pi(G) \leq k$.

We will prove that $|\partial V_i| \leq k$ for each i. Consider an arbitrary vertex $v_i \in V(D)$ and suppose that v_i was included in π at the j-th iteration, which means $v_i \in W_j$. Notice that $V_i \subseteq W_1 \cup \ldots \cup W_j$. We will first show that $\partial V_i \subseteq W_j$. Consider an arbitrary backward arc (x, y) with $x \in V(D) \setminus V_i$ and $y \in V_i$. Observe that $y \in W_p$ for some $p \leq j$, and if $x \in W_q$ then $q \geq j$. By the property (iii) of DPD, $\{x, y\} \subseteq W_s$ for some $s \geq j$. Thus, by the property (ii) of DPD, $y \in W_j$. Hence, we have shown that $\partial V_i \subseteq W_j$, which implies $|\partial V_i| \leq k+1$. To improve this inequality, we will consider the following two cases:

(a) V_i is a proper subset of $W_1 \cup \ldots \cup W_j$. Then ∂V_j is a proper subset of W_j and $|\partial V_i| \leq k$.

(b) $V_i = W_1 \cup \ldots \cup W_j$. As above we can show that $y \in W_{j'}$ for some $j' > j$. Thus, $y \in W_{j+1}$ and $|\partial V_j| \leq |W_j \cap W_{j+1}| \leq k$. The last inequality holds due to the fact that W_j and W_{j+1} are distinct.

In both cases we conclude that $|\partial V_i| \leq k$, which completes the proof. □

It follows from Theorem 2.13.7 that each directed cycle is of directed path-width 1.

Let Z be a set of vertices of a digraph D. A set $S \subseteq V(D) - Z$ is Z-**normal** if every directed walk that leaves and again enters S must traverse a vertex of Z. For vertices r, r' of an out-tree T we write $r \leq r'$ if there is a path from r to r' or $r = r'$. An **arboreal decomposition** of a digraph D is a triple (R, X, W), where R is an out-tree (not a subdigraph of D), $X = \{X_e : e \in A(R)\}$ and $W = \{W_r : r \in V(R)\}$ are sets of vertices of D that satisfy two conditions: (1) $\{W_r : r \in V(R)\}$ is a partition of $V(D)$ into nonempty sets, and (2) if for each $e = (r', r'') \in A(R)$ the set $\bigcup\{W_r : r \in V(R), r \geq r''\}$ is X_e-normal. The **width** of (R, X, W) is the least integer w such that for all $r \in V(R)$, $|W_r \cup \bigcup_{e \sim r} X_e| \leq w+1$, where $e \sim r$ means that r is head or tail of e. The **directed tree-width** of D, $\mathrm{dtw}(D)$, is the least integer w such that D has an arboreal decomposition of width w.

Now we will study some basic results on the three directed width parameters. The first lemma can be proved using only the definitions above (Exercise 2.48).

Lemma 2.13.8 *Let D be a digraph. For $\mathrm{d} \in \{\mathrm{dag}, \mathrm{dt}, \mathrm{dp}\}$, we have $\mathrm{dw}(D) = 0$ if and only if D is acyclic.* □

Let D be a digraph. It is immediately follows from the definitions of DAG-width and directed path-width that $\mathrm{dagw}(D) \leq \mathrm{dpw}(D)$. It is easy to show that $\mathrm{dtw}(D) \leq \mathrm{dpw}(D)$ (Exercise 2.49) from the definitions of the two parameters. Berwanger, Dawar, Hunter and Kreutzer [154] proved that $\mathrm{dtw}(D) \leq 3 \cdot \mathrm{dagw}(D) + 1$. Thus, we have the following:

Lemma 2.13.9 *For a digraph D, we have* $\mathrm{dagw}(D) \leq \mathrm{dpw}(D)$, $\mathrm{dtw}(D) \leq \mathrm{dpw}(D)$ *and* $\mathrm{dtw}(D) \leq 3 \cdot \mathrm{dagw}(D) + 1$. □

The last two lemmas imply, in particular, that if $\mathrm{dpw}(D) = 1$ then $\mathrm{dtw}(D) = \mathrm{dagw}(D) = 1$. Thus, for every directed cycle C, we have $\mathrm{dpw}(D) = \mathrm{dtw}(C) = \mathrm{dagw}(C) = 1$. Lemma 2.13.9 has many applications in this book.

Johnson, Robertson, Seymour and Thomas [573] proved that $\mathrm{tw}(G) = \mathrm{dtw}(\overleftrightarrow{G})$ for each undirected graph G and Obdržálek [731] showed that $\mathrm{tw}(G) = \mathrm{dagw}(\overleftrightarrow{G})$ for each undirected graph G. Since the tree-width problem is $\mathcal{NP}$-hard, so are the problems of checking whether $\mathrm{dtw}(D) \leq k$ and $\mathrm{dagw}(D) \leq k$ for a digraph D. However, there are $O(n^{O(k)})$-time algorithms for the two problems [573, 731].

Since $\mathrm{tw}(G) = \mathrm{dtw}(\overleftrightarrow{G})$ and $\mathrm{tw}(G) = \mathrm{dagw}(\overleftrightarrow{G})$ for each undirected graph G, it is easy to prove (Exercise 2.50) that $\mathrm{dtw}(D) \leq \mathrm{tw}(D)$ and $\mathrm{dagw}(D) \leq \mathrm{tw}(D)$ for each digraph D.

2.14 Other Families of Digraphs

This section is devoted to digraphs of three classes: circulant digraphs, arc-locally semicomplete digraphs and intersection digraphs.

2.14.1 Circulant Digraphs

For an integer $n \geq 2$ and a set $S \subseteq \{1, 2, \ldots, n-1\}$, the **circulant digraph** $C_n(S)$ is defined as follows: $V(C_n(S)) = \{1, 2, \ldots, n\}$ and

$$A(C_n(S)) = \{(i, i + j \pmod n) : \ 1 \leq i \leq n, j \in S\}.$$

In particular, $C_n(1, 2, \ldots, n-1) = \overleftrightarrow{K}_n$ and $C_n(1) = \vec{C}_n$ (it is customary to omit the set brackets when S is given by a list of its elements). Also, consecutive-1 digraphs introduced at the end of Section 2.5 are circulant digraphs. Circulant digraphs are a special family of Caley digraphs, see, e.g., [568] and are of importance in many applications of graph theory, see, e.g., [269]. Circulant digraphs are of great interest in digraph theory as well, cf. Sections 3.8.1, 6.9 and 15.6. We start from some basic properties of circulant digraphs.

Proposition 2.14.1 *Let $C_n(S)$ be a circulant digraph. Then the following holds:*

(a) $C_n(S)$ has a 2-cycle if and only if there is a pair s,t of elements of S such that $s+t=n$,
(b) the converse of $C_n(S)$ is isomorphic to $C_n(S)$,
(c) $C_n(S)$ is strong if and only if $\gcd(n,s_1,s_2,\ldots,s_p)=1$, where we have $\{s_1,s_2,\ldots,s_p\}=S$. □

Part (a) is easy to see: if $i\to j$ and $j\to i$, then we set $s=i-j$ and $t=j-i$; also, if $s+t=n$, then $(1,1+s)$ and $(1+s,1+s+t)=(1+s,1)$ are arcs. If n is odd $|S|=(n-1)/2$, then $C_n(S)$ is a tournament called a **rotational tournament** by Alspach [35]. To prove (b) observe that $C_n(-S)$ is the converse of $C_n(S)$, where $-S=\{-s:\ s\in S\}$, and that the bijection $f(i)=n-i$ of $[n]$ to itself is an isomorphism of $C_n(S)$ to $C_n(-S)$. It seems Ariyoshi [45] was the first to obtain Part (c); we leave the proof of (c) as an exercise.

In applications it is important to know which circulant digraphs $C_n(S)$ are $|S|$-strong [269] (since $C_n(S)$ is $|S|$-regular, $\kappa(C_n(S))\le|S|$ and so $|S|$-strong connectivity is maximal possible for $C_n(S)$). In [269] van Doorn obtained two sufficient conditions:

Theorem 2.14.2 *[269] A circulant digraph $C_n(S)$ is $|S|$-strong if at least one of the following conditions holds:*

(a) $\gcd(n,s)=1$ for each $s\in S$,
(b) $i\in S$ for each $i=1,2,\ldots,\lceil|S|/2\rceil$. □

2.14.2 Arc-Locally Semicomplete Digraphs

A digraph D is **arc-locally semicomplete** if for every arc xy of D, the following two conditions hold:

(a) if $u\in N^-(x)$, $v\in N^-(y)$ and $u\ne v$, then u and v are adjacent,
(b) if $u\in N^+(x)$, $v\in N^+(y)$ and $u\ne v$, then u and v are adjacent.

This class of digraphs was introduced by Bang-Jensen in [70]. Clearly, every semicomplete or semicomplete bipartite digraph is arc-locally semicomplete. The same holds for extensions of cycles. Bang-Jensen [75] proved that if we restrict ourselves to strong digraphs, the above three classes of digraphs are, in fact, all arc-locally semicomplete digraphs.

Theorem 2.14.3 *[75] A strong arc-locally semicomplete digraph is either semicomplete or semicomplete bipartite or an extension of a cycle.* □

If an arc-locally semicomplete digraph D is non-strong, we do not have a complete picture of how D 'looks like' apart from the case when every vertex of D is on some cycle. In this case, Bang-Jensen [70] showed that D is either semicomplete or semicomplete bipartite. The class of arc-locally semicomplete digraphs was also studied by Galeana-Sánchez [381].

It is natural to define **arc-in-locally (arc-out-locally) semicomplete digraphs** as digraphs satisfying the property (a) (the property (b)) above. To the best of our knowledge, nobody has studied the structure of these two classes of digraphs so far.

2.14.3 Intersection Digraphs

Let U and V be sets and let $\mathcal{F} = \{(S_v, T_v) : S_v, T_v \subseteq U \text{ and } v \in V\}$ be a family of ordered subsets of U (one for each $v \in V$). The **intersection digraph** corresponding to $\mathcal{F}$ is the digraph $D_{\mathcal{F}} = (V, A)$ such that $vw \in A$ if and only if $S_v \cap T_w \neq \emptyset$. The set U is called the **universal set** for $D_{\mathcal{F}}$. The above family of pairs form a **representation** of D. The concept of an intersection digraph is a natural analogue of the notion of an intersection graph and was introduced by Beineke and Zamfirescu [133] and Sen, Das, Roy and West [806]. Since an arc is an ordered pair of vertices, every line digraph $L(D)$ is the intersection digraph of the family $A(D')$, where D' is the converse of D. It follows from the definition of an intersection digraph that every digraph D is the intersection digraph of the family $\{(A^+(v), A^-(v)) : v \in V(D)\}$, where $A^+(v)$ $(A^-(v))$ is the set of arcs leaving v (entering v). Here the universal set is $A(D)$.

Clearly, a digraph can be represented as the intersection digraph of various families of ordered pairs. It is quite natural to ask how large the universal set U has to be. For a digraph D the minimum number of elements in U such that $D = D_{\mathcal{F}}$ for some family $\mathcal{F}$ of ordered pairs of subsets of U is called the **intersection number**, $\text{in}(D)$ of D. Sen, Das, Roy and West [806] prove the following theorem for the intersection number of an arbitrary digraph D. For a digraph $D = (V, A)$, a set $B \subseteq A$ is **one-way** if there is a pair of sets $X, Y \subset V$ (called a **generating pair**) such that $B = (X, Y)_D$, that is, B is the set of arcs from X to Y.

Theorem 2.14.4 *[806] The intersection number of a digraph $D = (V, A)$ equals the minimum number of one-way sets required to cover A.*

Proof: Let $B_1, \ldots, B_k$ be a minimum collection of one-way sets covering A and let $(X_1, Y_1), \ldots, (X_k, Y_k)$ be the corresponding generating pairs. Let $S_v = \{i : \ v \in X_i\}$, and $T_v = \{i : \ v \in Y_i\}$. Then $S_v \cap T_w \neq \emptyset$ if and only if $vw \in A$, showing that $\text{in}(D) \leq k$.

Now let U be a universal set of cardinality $u = \text{in}(D)$ such that D has a representation by a set of ordered pairs (S_v, T_v) of subsets of U. We may assume that $U = [u]$. Define u one-way sets covering A as follows: $v \in X_i$ if

and only if $i \in S_v$ and $v \in Y_i$ if and only if $i \in T_v$. Then $vw \in A$ if and only if $v \in X_i,\ w \in Y_i$ for some i. Thus, $k \leq \text{in}(D)$. □

Notice that the minimum number of one-way sets required to cover A is studied in Subsection 13.12.1.

A **subtree intersection digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of subtrees in an undirected tree. A **matching diagram digraph** is digraph representable as the intersection digraph of a family of ordered pairs of straight-line segments between two parallel lines. An **interval digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of closed intervals on the real line. Subtree intersection digraphs, matching diagram digraphs and interval digraphs are 'directed' analogues of chordal graphs, permutation graphs and interval graphs, respectively, where subtrees, straight-line segments and real line intervals are also used for representation (see the book [421] by Golumbic). While chordal graphs form a special family of undirected graphs, Harary, Kabell and McMorris showed that every digraph is a subtree intersection digraph.

Proposition 2.14.5 *[499] Every digraph is a subtree intersection digraph.*

Proof: Let $D = (V, A)$ be an arbitrary digraph. Let $G = (U, E), U = V \cup \{x\}$, $E = \{\{x, v\} : v \in V\}$, $x \notin V$. Clearly, G is an undirected tree. Setting $S_v = G\langle\{v\}\rangle$ and $T_v = G\langle\{x\} \cup \{w : wv \in A\}\rangle$ provides the required representation. □

The following construction by Müller shows that every interval digraph is a matching diagram digraph [708]. Let $\{([a_v, b_v], [c_v, d_v]) : v \in V(D)\}$ be a representation of an interval digraph D. To obtain a representation $\{(S_v, T_v) : v \in V(D)\}$ of D as a matching diagram digraph we set S_v to be the line segment between points $(a_v, 0)$ and $(b_v, 1)$ in the plane, and T_v to be the line segment connecting the points $(c_v, 1)$ and $(d_v, 0)$.

There are several characterizations of interval digraphs, see, e.g., the papers [793] by Sanyal and Sen and [903] by West. We restrict ourselves to just one of them.

Theorem 2.14.6 *[806] A digraph D is an interval digraph if and only if there exist independent row and column permutations of the adjacency matrix $M(D)$ of D which result in a matrix M' satisfying the following property: the zero entries of M' can be labeled R or C such that every position above and to the right of an R is an R and every position below and to the left of a C is a C.* □

None of the characterizations given in [793, 903] implies a polynomial algorithm to recognize interval digraphs. Müller [708] obtained such an algorithm. A polynomial algorithm is also given in [708] to recognize unit interval digraphs, i.e., interval digraphs that have interval representations, where all

intervals are of the same length. Brown, Busch and Lundgren [182] showed that a tournament of order n is an interval digraph if and only if it contains a transitive tournament of order $n-1$ (as a subdigraph).

2.15 Exercises

2.1. **Uniqueness of acyclic orderings.** Prove that an acyclic digraph D has a unique acyclic ordering if and only if D is traceable.

2.2. **Linear time algorithm for finding an acyclic ordering of an acyclic digraph**. Verify that the algorithm given in the proof of Proposition 2.1.3 can be implemented as an $O(n+m)$ algorithm using the adjacency list representation (see Section 18.1).

2.3. Prove that a tournament is transitive if and only if it is acyclic. Hint: apply Theorem 1.5.1.

2.4. Prove Proposition 2.3.1.

2.5. Let D be a semicomplete multipartite digraph such that every vertex of D is on some cycle. Prove that D is unilateral.

2.6. In part (ii) $\Rightarrow$ (i) of Theorem 2.4.1, prove that $\sigma(D) = L(Q)$.

2.7. Derive Corollary 2.4.2 from Theorem 2.4.1 (iii).

2.8. **(−)** Prove Proposition 2.4.3 using Theorem 2.4.1 (i) and (ii).

2.9. Prove the following simple properties of line digraphs:
(i) $L(D) \cong \vec{P}_{n-1}$ if and only if $D \cong \vec{P}_n$;
(ii) $L(D) \cong \vec{C}_n$ if and only if $D \cong \vec{C}_n$.

2.10. Let D be a digraph. Show by induction that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1 \dots v_k$ dominates the vertex $v_1v_2 \dots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that $v_kv_{k+1} \in A(D)$.

2.11. Using the results in Exercise 2.9, prove the following elementary properties of iterated line digraphs: Let D be a digraph. Then
(i) $L^k(D)$ is a digraph with no arcs, for some k, if and only if D is acyclic;
(ii) if D has a pair of cycles joined by a path (possibly of length 0), then

$$\lim_{k \to \infty} n_k = \infty,$$

where n_k is the order of $L^k(D)$;
(iii) if no pair of cycles of D is joined by a path, then for all sufficiently large values of k, each connected component of $L^k(D)$ has at most one cycle.

2.12. Prove Proposition 2.4.4 by induction on $k \geq 1$.

2.13. Prove Lemma 2.5.1.

2.14. Prove Lemma 2.5.5.

2.15. Prove Lemma 2.5.6.

2.16. Prove Theorem 2.5.7.

2.17. **Upwards embeddings of MVSP digraphs.** Prove that one can embed every MVSP digraph D into the Cartesian plane such that if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, and there is a (u, v)-path in D, then $x_u \leq x_v$ and $y_u \leq y_v$. Hint: consider series composition and parallel composition separately.

2.18. Prove Proposition 2.6.2. Hint: use induction on the number of reductions applied for the 'if' part and the number of arcs for the 'only if' part.

2.19. Prove Proposition 2.6.3.

2.20. Prove part (b) of Lemma 2.7.4. Hint: if u and v are in S, then there is a path from u to v in $\overline{UG(S)}$. Similarly, if x and y are in S'. Use these paths (corresponding to sequences of non-adjacent vertices in D) to show that if xu and vy are arcs, then $u = v$ and $x = y$ must hold if D is quasi-transitive.

2.21. (−) Construct an infinite family of path-mergeable digraphs, which are not in-path-mergeable.

2.22. (−) Show that the following 'claim' is wrong. Let D be a locally in-semicomplete digraph and let D contain internally disjoint paths P_1, P_2 such that P_i is an (x_i, y)-path $(i = 1, 2)$ and $x_1 \neq x_2$. Then x_1 and x_2 are adjacent.

2.23. **Orientations of path-mergeable digraphs.** Prove that every orientation of a path-mergeable digraph is a path-mergeable oriented graph.

2.24. (+) Prove Corollary 2.8.2.

2.25. Prove Proposition 2.9.2.

2.26. **Path-mergeable digraphs which are neither locally in-semicomplete nor locally out-semicomplete.** Show by a construction that there exists an infinite class of path-mergeable digraphs, none of which is locally in-semicomplete or locally out-semicomplete. Then extend your construction to arbitrary degrees of vertex-strong connectivity. Hint: consider extensions.

2.27. (−) **Path-mergeable transitive digraphs.** Prove that a transitive digraph $D = (V, A)$ is path-mergeable if and only if for every $x, y \in V$ and every pair xuy, xvy of (x, y)-paths of length 2, either $u \to v$ or $v \to u$ holds.

2.28. Prove Lemma 2.9.3.

2.29. **Orientations of locally in-semicomplete digraphs**. Prove that every orientation of a digraph which is locally in-semicomplete is a locally in-tournament digraph.

2.30. **Strong orientations of strong locally in-semicomplete digraphs**. Prove that every strong locally in-semicomplete digraph on at least three vertices has a strong orientation.

2.31. Prove Lemma 2.10.2.

2.32. Prove Corollary 2.10.7.

2.33. Prove Theorem 2.10.8.

2.34. (+) Using Lemma 2.10.13, show that if D is a non-round decomposable locally semicomplete digraph, then the independence number of $UG(D)$ is at most two.

2.35. (−) Give an example of a locally semicomplete digraph on 4 vertices with no 2-king.

2.36. Prove Proposition 2.10.16.

2.37. Prove the assertion stated in Exercise 2.34 using Lemma 2.10.14 and Proposition 2.10.16.

2.38. **Extending in-path-mergeability.** Prove that if P, Q are internally disjoint (x, z)- and (y, z)-paths in an extended locally in-semicomplete digraph D and no vertex on $P - z$ is similar to a vertex of $Q - z$, then there is a path R from either x or y to z in D such that $V(R) = V(P) \cup V(Q)$.

2.39. Prove that there exists an $O(mn + n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either semicomplete bipartite or connected extended locally semicomplete.

2.40. (−) Let D be a connected digraph which is both quasi-transitive and locally semicomplete. Prove that D is semicomplete.

2.41. (−) Let D be a connected digraph which is both quasi-transitive and locally in-semicomplete. Prove that the diameter of $UG(D)$ is at most 2.

2.42. **Traceable semicomplete bipartite digraph characterization.** Prove that a semicomplete bipartite digraph B is traceable if and only if it contains a 1-path-cycle factor $\mathcal{F}$. Hint: demonstrate that if $\mathcal{F}$ consists of a path and a cycle only, then B is traceable; use it to establish the desired result (Gutin [445]). (See also Chapter 6.)

2.43. Prove that if a bipartite tournament has a cycle, then it has a 4-cycle.

2.44. Show that every orientation of a quasi-transitive digraph is a quasi-transitive digraph.

2.45. (−) Prove that the intersection number $\text{in}(D) \leq n$ for every digraph D of order n. Show that this upper bound is sharp (Sen, Das, Roy and West [806]).

2.46. Prove Corollary 2.12.3. Hint: use that each edge is on the boundary of precisely two faces and that each face has at least three edges.

2.47. Using only the definition of tree-width prove that a connected digraph D is of tree-width one if and only if D is a biorientation of a tree.

2.48. Prove Lemma 2.13.8.

2.49. Prove that $\text{dtw}(D) \leq \text{dpw}(D)$ every digraph D using only the definitions of directed tree-width and directed path-width.

2.50. (−) Using the fact that $\text{tw}(G) = \text{dtw}(\overleftrightarrow{G})$ and $\text{tw}(G) = \text{dagw}(\overleftrightarrow{G})$ for each undirected graph G prove that $\text{dtw}(D) \leq \text{tw}(D)$ and $\text{dagw}(D) \leq \text{tw}(D)$ for each digraph D.

2.51. Prove Proposition 2.14.1 (c).

3. Distances

In this chapter, we study polynomial algorithms which find distances in weighted and unweighted digraphs as well as some related complexity results. We consider bounds on the diameter of a digraph and describe several results on minimizing the diameter of an orientation of directed and undirected graphs. We deal with kings, kernels and quasi-kernels in digraphs.

Additional terminology and notation are given in Section 3.1. Some basic results on the structure of shortest paths in weighted digraphs are proved in Section 3.2. In Section 3.3 we study algorithms for finding shortest paths from a vertex to the rest of the vertices of weighted and unweighted digraphs. We also consider the Floyd-Warshall algorithm to compute distances between all pairs of vertices in a weighted digraph. In Section 3.4 we consider bounds on the diameter of strong directed and oriented graphs. The problem of minimizing the diameter of an orientation of a bridgeless graph is studied extensively in Section 3.5.

Section 3.6 is devoted to (almost) minimum diameter orientations of graphs belonging to special families of directed and undirected graphs. Quasi-transitive digraphs, semicomplete bipartite digraphs and locally semicomplete digraphs are considered in Subsection 3.6.1. We finish the subsection with a conjecture for semicomplete multipartite digraphs. In Subsection 3.6.2, we deal with extended digraphs. Cartesian products of undirected graphs are considered in Subsection 3.6.3 and chordal graphs in Subsection 3.6.4.

The notion of kings is investigated in Section 3.7. This notion is related to the study of domination in biology and sociology. We study kernels and quasi-kernels in Section 3.8.

3.1 Terminology and Notation on Distances

Let $D = (V, A)$ be a directed pseudo-graph. Recall that for a set $W \subseteq V$,

$$N_D^+(W) = \bigcup_{w \in W} N^+(w) - W, \; N_D^-(W) = \bigcup_{w \in W} N^-(w) - W.$$

Let $N_D^0(W) = W$, $N_D^{+1}(W) = N_D^+(W)$, $N_D^{-1}(W) = N_D^-(W)$. For every positive integer p, we can define the ***p*th out-neighbourhood** of W as follows:

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_3,

$$N_D^{+p}(W) = N_D^+(N_D^{+(p-1)}(W)) - \bigcup_{i=0}^{p-1} N_D^{+i}(W).$$

Similarly, one can define $N_D^{-p}(W)$ for every positive integer p. In particular, $N^{+2}(W) = N^+(N^+(W)) - (W \cup N^+(W))$. Sometimes, $N_D^{+p}(W)$ ($N_D^{-p}(W)$) is called the **open *p*th out-neighbourhood** (**open *p*th in-neighbourhood**) of W. We will also use the **closed *p*th in- and out-neighbourhoods** of a set W of vertices of D which are defined as follows ($p > 0$):

$$N_D^0[W] = W, \quad N_D^{+p}[W] = \bigcup_{i=0}^{p} N_D^{+i}(W), \quad N_D^{-p}[W] = \bigcup_{i=0}^{p} N_D^{-i}(W).$$

To simplify the notation, we set $N_D^+[W] = N_D^{+1}[W]$ and $N_D^-[W] = N_D^{-1}[W]$. See Figure 3.1.

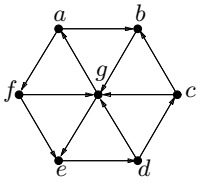


Figure 3.1 A digraph D. The out-neighbourhoods of the set $W = \{a, b\}$ are $N^+(\{a, b\}) = \{f, g\}$, $N^{+2}(\{a, b\}) = \{e\}$, $N^{+3}(\{a, b\}) = \{d\}$, $N^{+4}(\{a, b\}) = \{c\}$. The closed out-neighbourhoods are $N^+[W] = \{a, b, f, g\}$, $N^{+2}[W] = \{a, b, e, f, g\}$, $N^{+3}[W] = \{a, b, d, e, f, g\}$, $N^{+4}[W] = \{a, b, c, d, e, f, g\}$.

Let $D = (V, A, c)$ be a directed multigraph with a weight function $c : A \to \mathcal{R}$ on its arcs. Recall that the weight of a subdigraph $D' = (V, A')$ of D is given by $c(A') = \sum_{a \in A'} c(a)$. Whenever we speak about the length of a walk we mean the weight of that walk (with respect to c). A **negative cycle** in a weighted digraph $D = (V, A, c)$ is a cycle W whose weight $c(W)$ is negative.

We assume that D has no negative cycle, for otherwise the following definition becomes meaningless. If x and y are vertices of D, then the **distance from *x* to *y*** in D, denoted $\mathrm{dist}(x, y)$, is the minimum length of an (x, y)-walk, if y is reachable from x, and otherwise $\mathrm{dist}(x, y) = \infty$. Since D has no cycle of negative weight, it follows that $\mathrm{dist}(x, x) = 0$ for every vertex $x \in V$. It follows from Proposition 1.4.1 that there is a shortest (x, y)-walk which is, in fact, a path (if D has no cycle of zero weight either, a shortest walk is always a path). Furthermore, by Proposition 1.4.1, the distance function satisfies the triangle inequality:

$$\mathrm{dist}(x, z) \leq \mathrm{dist}(x, y) + \mathrm{dist}(y, z) \text{ for every triple of vertices } x, y, z. \quad (3.1)$$

The above definitions are applicable to unweighted directed multigraphs as well: simply take the weight of every arc equal to one (then, the length of a walk in the 'weighted' and 'unweighted' cases coincide).

The **distance from a set X to a set Y** of vertices in D is

$$\mathrm{dist}(X,Y) = \max\{\mathrm{dist}(x,y) : \ x \in X, y \in Y\}.^{1} \tag{3.2}$$

The **diameter** of D is $\mathrm{diam}(D) = \mathrm{dist}(V,V)$. Clearly, D has finite diameter if and only if D is strong. The **out-radius** $\mathrm{rad}^+(D)$ and the **in-radius** $\mathrm{rad}^-(D)$ of D are defined by

$$\mathrm{rad}^+(D) = \min\{\mathrm{dist}(x,V) : \ x \in V\}, \qquad \mathrm{rad}^-(D) = \min\{\mathrm{dist}(V,x) : \ x \in V\}.$$

Because of the obvious duality between out-radius and in-radius, in many cases, we will consider only one of them. The **radius** of D is

$$\mathrm{rad}(D) = \min\{(\mathrm{dist}(x,V) + \mathrm{dist}(V,x))/2 : \ x \in V\}.$$

To illustrate the definitions above, consider the digraph D in Figure 3.1. Here we have $\mathrm{dist}(a,V) = \mathrm{dist}(b,V) = \mathrm{dist}(e,V) = 4$ and $\mathrm{dist}(c,V) = \mathrm{dist}(d,V) = \mathrm{dist}(f,V) = \mathrm{dist}(g,V) = 3$. Furthermore, we have $\mathrm{dist}(V,c) = \mathrm{dist}(V,f) = 4$, $\mathrm{dist}(V,a) = \mathrm{dist}(V,b) = \mathrm{dist}(V,d) = 3$ and $\mathrm{dist}(V,e) = \mathrm{dist}(V,g) = 2$. Now we see that $\mathrm{rad}^+(D) = 3$, $\mathrm{rad}^-(D) = 2$, $\mathrm{rad}(D) = 2.5$ and $\mathrm{diam}(D) = 4$. It is also easy to see that $\mathrm{dist}(\{a,c\},\{b,f\}) = 3$.

The following proposition gives a characterization of weighted digraphs D of finite out-radius.

Proposition 3.1.1 *A weighted digraph D has a finite out-radius if and only if D has a unique initial strong component.*

Proof: A digraph with two or more initial strong components is obviously of infinite out-radius. If D has only one initial strong component, then D contains an out-branching (by Proposition 1.7.1). Thus, $\mathrm{rad}^+(D) < \infty$. □

This proposition implies that a weighted digraph D has a finite in-radius if and only if D has a unique terminal strong component. Notice that $\mathrm{rad}(D) < \infty$ if and only if D is strong.

For an undirected graph G, we can introduce the notions of distance between pairs of vertices, vertex sets, radius, etc. by considering $\overleftrightarrow{G}$.

3.2 Structure of Shortest Paths

In this section we study elementary, but very important properties of shortest paths in weighted digraphs. We also consider some complexity results on paths in directed and mixed weighted graphs.

[1] This definition may seem somewhat unnatural (with max instead of min), but it simplifies some of the notation in this chapter and also appears quite useful.

We assume that $D = (V, A, c)$ is a weighted digraph with no negative cycle.

Proposition 3.2.1 *If $P = x_1x_2 \dots x_k$ is a shortest (x_1, x_k)-path in D, then $P[x_i, x_j]$ is a shortest (x_i, x_j)-path for all $1 \leq i \leq j \leq k$.*

Proof: Suppose that x_iQx_j is an (x_i, x_j)-path whose length is smaller than that of $P[x_i, x_j]$. Then the weight of the walk $W = P[x_1, x_i]QP[x_j, x_k]$ is less than the length of P. However, by Proposition 1.4.1, and the fact that D has no negative cycle, W contains an (x_i, x_j)-path R whose length is at most that of W and hence is smaller than that of P, a contradiction. □

Let s be a fixed vertex of D such that $\text{dist}(s, V) < \infty$. Consider spanning subdigraphs of D, each of which contains a shortest path from s to every other vertex in D. The proof of the following theorem shows that given any such subdigraph D' of D, we can construct an out-branching of D rooted at s, which contains a shortest (s, u)-path for every $u \in V - s$.

Theorem 3.2.2 *Let D' and s be as above. There exists an out-branching B_s^+ such that, for every $u \in V$, the unique (s, u)-path in B_s^+ is a shortest (s, u)-path in D.*

Proof: We will give a constructive proof showing how to build B_s^+ from any collection $\{P_v : v \in V - s\}$ of shortest paths from s to the rest of the vertices.

Choose a vertex $u \in V - s$ arbitrarily. Let initially $B_s^+ := P_u$. By Proposition 3.2.1, for every $x \in V(B_s^+)$, the unique (s, x)-path in B_s^+ is a shortest (s, x)-path in D. Hence, if $V(B_s^+) = V$, then we are done. Thus, we may assume that there exists $w \notin V(B_s^+)$. Let z be the last vertex on P_w which belongs to B_s^+. Define H as follows:

$$V(H) := V(B_s^+) \cup V(P_w[z, w]), \;\; A(H) := A(B_s^+) \cup A(P_w[z, w]).$$

We claim that, for every vertex x in $P_w[z, w]$, the unique (s, x)-path in H is a shortest (s, x)-path in D. By Proposition 3.2.1, $P_w[s, z]$ is a shortest (s, z)-path in D. Since $z \in V(B_s^+)$, the unique (s, z)-path Q in H has the same length as $P_w[s, z]$. Therefore, the length of the path $QP_w[z, x]$ is equal to the length of the path $P_w[s, x]$. Now observe that $QP_w[z, x]$ is the unique (s, x)-path in H. We set $B_s^+ := H$ and use an analogous approach to include all vertices of D and preserve the desired property of B_s^+. □

Our constructive proof above implies the existence of a polynomial algorithm to construct the final out-branching, starting from a collection of shortest paths from s to all other vertices. We call such an out-branching a **shortest path tree from** s. As we will see in Exercises 3.8 and 3.9, the algorithms described in the next section can be easily modified so that they construct a shortest path tree directly, while searching for the shortest paths.

If we allow D to have negative weight cycles, then we obtain the following result for shortest paths (recall that in the presence of negative cycles the length of a shortest walk may not be defined, whereas the length of a shortest path is still well-defined).

Proposition 3.2.3 *It is $\mathcal{NP}$-hard to find a shortest path between a pair of vertices of a given weighted digraph.*

Proof: Let $D = (V, A)$ be an (unweighted) digraph and let $x \neq y$ be vertices of D. Set $c(uv) = -1$ for every arc $uv \in A$. We have obtained a weighted digraph $D' = (V, A, c)$. Clearly, D' has an (x, y)-path of length $1 - n$ if and only if D has a hamiltonian (x, y)-path. Since the problem of finding a hamiltonian (x, y)-path is $\mathcal{NP}$-complete (see Exercise 7.2) and D' can be constructed from D in polynomial time, our claim follows. □

Clearly D' above has a negative cycle if and only if D has any directed cycle. As we will show in Subsection 3.3.2, we can find a longest path in an acyclic digraph in polynomial time, using a reduction to the shortest path problem.

In Section 3.3, we will see that one can check whether a weighted digraph has a negative cycle in polynomial time. However, unless $\mathcal{P} = \mathcal{NP}$, this result cannot be extended to weighted mixed graphs, because of the following theorem by Arkin and Papadimitriou [48].

Theorem 3.2.4 *Given a weighted mixed graph, it is $\mathcal{NP}$-complete to determine whether a negative cycle exists.* □

It follows from Proposition 3.2.3 that it is $\mathcal{NP}$-hard to find a shortest path between a pair of vertices in a weighted mixed graph. More interestingly, Arkin and Papadimitriou showed that the same is true even if we restrict ourselves to weighted mixed graphs without negative cycles [48].

3.3 Algorithms for Finding Distances in Digraphs

In this section we describe well-known algorithms to find distances in weighted and unweighted digraphs. Almost all algorithms which we describe are for finding the distances from a fixed vertex of a digraph to the rest of the vertices. If the given digraph is unweighted, then one can use the very simple and fast breadth-first search algorithm that is introduced in Subsection 3.3.1. If the given digraph D is weighted and acyclic, another fast and simple approach based on dynamic programming is provided in Subsection 3.3.2. When D is an arbitrary digraph, but its weights are non-negative, Dijkstra's algorithm introduced in Subsection 3.3.3 solves the problem. When the weights may be negative, but no negative cycle is allowed, the Bellman-Ford-Moore algorithm given in Subsection 3.3.4 can be applied. This algorithm has the

following additional useful property: it can be used to detect a negative cycle (if one exists).

If we are interested in finding the distances between all pairs of vertices of a weighted digraph D, we can apply the Bellman-Ford-Moore algorithm from every vertex of D. However, there is a much faster algorithm, due to Floyd and Warshall. We describe the Floyd-Warshall algorithm in Subsection 3.3.5. The reader can find comprehensive overviews of theoretical and practical issues on the topic in the papers [211] by Cherkassky and Goldberg and [212] by Cherkassky, Goldberg and Radzik.

3.3.1 Breadth-First Search (BFS)

This approach allows one to find quickly the distances from a given vertex s to the rest of the vertices in an unweighted digraph $D = (V, A)$. BFS is based on the following simple idea. Starting at s, we visit each vertex x dominated by s. We set $\mathrm{dist}'(s, x) := 1$ and $s := \mathrm{pred}(x)$ (s is the predecessor of x). Now we visit all vertices y not yet visited and dominated by vertices x of distance 1 from s. We set $\mathrm{dist}'(s, y) := 2$ and $x := \mathrm{pred}(y)$. We continue in this fashion until we have reached all vertices which are reachable from s (this will happen after at most $n-1$ iterations, by Proposition 1.4.1). For the rest of the vertices z (not reachable from s), we set $\mathrm{dist}'(s, z) := \infty$. In other words, we visit the first (open) out-neighbourhood of s, then its second (open) out-neighbourhood, etc. A more formal description of BFS is as follows. At the end of the algorithm, $\mathrm{pred}(v) =$ **nil** means that either $v = s$ or v is not reachable from s. The correctness of the algorithm is due to the fact that $\mathrm{dist}(s, x) = \mathrm{dist}'(s, x)$ for every $x \in V$. This will be proved below.

BFS
Input: A digraph $D = (V, A)$ and a vertex $s \in V$.
Output: $\mathrm{dist}'(s, v)$ and $\mathrm{pred}(v)$ for all $v \in V$.

1. For each $v \in V$ set $\mathrm{dist}'(s, v) := \infty$ and $\mathrm{pred}(v) :=$ **nil**.
2. Set $\mathrm{dist}'(s, s) := 0$. Create a queue Q consisting of s.
3. While Q is not empty do the following. Delete a vertex u, the head of Q, from Q and consider the out-neighbours of u in D one by one. If, for an out-neighbour v of u, $\mathrm{dist}'(s, v) = \infty$, then set $\mathrm{dist}'(s, v) := \mathrm{dist}'(s, u)+1$, $\mathrm{pred}(v) := u$, and put v to the end of Q.

If D is represented by adjacency lists, the complexity of the above algorithm is $O(n + m)$. Indeed, Step 1 requires $O(n)$ time. The time to perform Step 3 is $O(m)$ as the out-neighbours of every vertex are considered only once and $\sum_{x \in V} d^+(x) = m$, by Proposition 1.2.1.

To prove the correctness of BFS, it suffices to prove that $\mathrm{dist}(s, x) = \mathrm{dist}'(s, x)$ for every $x \in V$. By Steps 2 and 3 of the algorithm, $\mathrm{dist}(s, x) \leq \mathrm{dist}'(s, x)$. Indeed, $v_1 v_2 \dots v_k$, where $v_1 = s$, $v_k = x$ and $v_i = \mathrm{pred}(v_{i+1})$

for every $i \in [k-1]$, is an (s,x)-path. By induction on $\mathrm{dist}(s,x)$, we prove that, in fact, the equality holds. If $\mathrm{dist}(s,x) = 0$, then $x = s$ and the result follows. Suppose that $\mathrm{dist}(s,x) = k > 0$ and consider a shortest (s,x)-path P. Let y be the predecessor of x, i.e., $y = x_P^-$. By the induction hypothesis, $\mathrm{dist}'(s,y) = \mathrm{dist}(s,y) = k-1$. Since y dominates x, by the algorithm, $\mathrm{dist}'(s,x) \leq \mathrm{dist}'(s,y)+1 = k = \mathrm{dist}(s,x)$. Combining $\mathrm{dist}(s,x) \leq \mathrm{dist}'(s,x)$ with $\mathrm{dist}'(s,x) \leq \mathrm{dist}(s,x)$, we are done.

The BFS algorithm allows one to compute the radius, out-radius, in-radius and diameter of a digraph in time $O(n^2+nm)$. Using the array pred one can generate the actual paths. We finish this section with the following two important observations which are stated as propositions. Proposition 3.3.1 follows from the description of BFS. Proposition 3.3.2 has already been proved. In both propositions $D = (V,A)$ is a directed multigraph with a specified vertex s.

Proposition 3.3.1 *Let U be the set of vertices reachable from s. Then (U,B), where $B = \{(\mathrm{pred}(v),v) : v \in U - s\}$ is an out-branching in $D\langle U\rangle$ with root s.* □

We call the out-branching in the above proposition a **BFS tree** of $D\langle U\rangle$ with **root** s, or simply a **BFS tree from** s. It is instructive to compare Proposition 3.3.1 with Theorem 3.2.2.

Proposition 3.3.2 *Let $\mathrm{dist}(s,V) < \infty$. For every non-negative integer $p \leq \mathrm{dist}(s,V)$, we have $N^{+p}(s) = \{v \in V : \mathrm{dist}(s,v) = p\}$.* □

Given a directed multigraph $D = (V,A)$ and a vertex s we call sets

$$N^0(s), N^+(s), N^{+2}(s), N^{+3}(s), \ldots,$$

the **distance classes from** s. By the proposition above, $N^{+i}(s)$ consists precisely of those vertices whose distance from s is i. See Figure 3.2 for an illustration of a BFS tree and the corresponding distance classes.

Summarizing the discussion above we obtain the following.

Theorem 3.3.3 *When applied to a directed multigraph D and a vertex s in D, the BFS algorithm correctly determines a BFS tree T from s in D in time $O(n+m)$. Furthermore, the distance classes from s in D are the same as the distance classes from s in T.* □

3.3.2 Acyclic Digraphs

Let $D = (V,A,c)$ be an acyclic weighted digraph. We will show that the distances from a vertex s to the rest of the vertices can be found quite easily, using dynamic programming. Without loss of generality, we may assume that

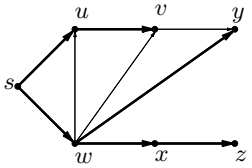


Figure 3.2 A digraph D with a BFS tree indicated by the bold arcs. The distance classes from s are $N^0(s) = s$, $N^+(s) = \{u, w\}$, $N^{+2}(s) = \{v, x, y\}$ and $N^{+3}(s) = \{z\}$.

the in-degree of s is zero. Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be an acyclic ordering of the vertices of D such that $v_1 = s$. Clearly, $\text{dist}(s, v_1) = 0$. For every i, $2 \leq i \leq n$, we have

$$\text{dist}(s, v_i) = \begin{cases} \min\{\text{dist}(s, v_j) + c(v_j, v_i) : \ v_j \in N^-(v_i)\} & \text{if } N^-(v_i) \neq \emptyset \\ \infty & \text{otherwise.} \end{cases} \tag{3.3}$$

The correctness of this formula can be shown by the following argument. We may assume that v_i is reachable from s. Since the ordering $\mathcal{L}$ is acyclic, the vertices of a shortest path P from s to v_i belong to $\{v_1, v_2, \ldots, v_i\}$. Let v_k be the vertex dominating v_i in P. By induction, $\text{dist}(s, v_k)$ is computed correctly using (3.3). The term $\text{dist}(s, v_k) + c(v_k, v_i)$ is one of the terms on the right-hand side of (3.3). Clearly, it provides the minimum.

The algorithm has two phases: the first finds an acyclic ordering, the second implements (3.3). The complexity of this algorithm is $O(n+m)$ since the first phase runs in time $O(n+m)$ (see Section 2.1) and the second phase requires the same asymptotic time due to the formula $\sum_{x \in V} d^-(x) = m$ in Proposition 1.2.1. Hence we have shown the following:

Theorem 3.3.4 *The shortest paths from a fixed vertex s to all other vertices can be found in time $O(n+m)$ for acyclic digraphs.* □

We can also find the length of longest (s, x)-paths in linear time in any acyclic digraph, by replacing the weight $c(uv)$ of every arc uv with $-c(uv)$. In particular, we can see immediately that the longest path problem for acyclic digraphs is solvable in polynomial time. In fact, a longest path of an acyclic digraph can always be found in linear time:

Theorem 3.3.5 *For acyclic digraphs a longest path can be found in time $O(n+m)$.*

Proof: Exercise 3.6. □

3.3.3 Dijkstra's Algorithm

The next algorithm, due to Dijkstra [259], finds the distances from a given vertex s in a weighted digraph $D = (V, A, c)$ to the rest of the vertices, provided that all the weights of arcs are non-negative.

In the course of the execution of Dijkstra's algorithm, the vertex set of D is partitioned into two sets, P and Q. Moreover, a parameter δ_v is assigned to every vertex $v \in V$. Initially all vertices are in Q. In the process of the algorithm, the vertices reachable from s move from Q to P. While a vertex v is in Q, the corresponding parameter δ_v is an upper bound on $\mathrm{dist}(s, v)$. Once v moves to P, we have $\delta_v = \mathrm{dist}(s, v)$. A formal description of Dijkstra's algorithm follows.

Dijkstra's algorithm
Input: A weighted digraph $D = (V, A, c)$, such that $c(a) \geq 0$ for every $a \in A$, and a vertex $s \in V$.
Output: The parameter δ_v for every $v \in V$ such that $\delta_v = \mathrm{dist}(s, v)$.

1. Set $P := \emptyset$, $Q := V$, $\delta_s := 0$ and $\delta_v := \infty$ for every $v \in V - s$.
2. While Q is not empty do the following.
   Find a vertex $v \in Q$ such that $\delta_v = \min\{\delta_u : u \in Q\}$.
   Set $Q := Q - v$, $P := P \cup v$.
   $\delta_u := \min\{\delta_u, \delta_v + c(v, u)\}$ for every $u \in Q \cap N^+(v)$.

To prove the correctness of Dijkstra's algorithm, it suffices to show that the following proposition holds.

Proposition 3.3.6 *At any time during the execution of the algorithm, we have that*

(a) For every $v \in P$, $\delta_v = \mathrm{dist}(s, v)$.
(b) For every $u \in Q$, δ_u is the distance from s to u in the subdigraph of D induced by $P \cup u$.

Proof: When $P = \emptyset$, $\delta_s = \mathrm{dist}(s, s) = 0$ and the estimates $\delta_u = \infty$, $u \in V - s$, are also correct.

Assume that $P = P_0$ and $Q = Q_0$ are such that the statement of this proposition holds. If $Q_0 = \emptyset$, we are done. Otherwise, let v be the next vertex chosen by the algorithm. Since Part (b) follows from Part (a) and the way in which we update δ_u in the algorithm, it suffices to prove Part (a) only. Suppose that (a) does not hold for $P = P_0 \cup v$. This means that $\delta_v > \mathrm{dist}(s, v)$. Let W be a shortest (s, v)-path in D. Since $\delta_v > \mathrm{dist}(s, v)$, W must contain at least one vertex from $Q = Q_0 - v$. Let u be the first vertex on W which is not in P_0. Clearly, $u \neq v$. By Proposition 3.2.1 and the fact that $u \in W$, we have $\mathrm{dist}(s, u) \leq \mathrm{dist}(s, v)$. Furthermore, since the statement of this proposition holds for P_0 and Q_0, we obtain that $\mathrm{dist}(s, u) = \delta_u$. This implies that $\delta_u = \mathrm{dist}(s, u) \leq \mathrm{dist}(s, v) < \delta_v$. In particular, $\delta_u < \delta_v$, which contradicts the choice of v by the algorithm. □

Each time a new vertex v is to be chosen we use $O(n)$ comparisons to find $\min\{\delta_u : u \in Q\}$. Updating the parameters takes $O(n)$ time as well. Since Step 2 is performed $n-1$ times, we conclude that the complexity of Dijkstra's algorithm is $O(n^2)$. In fact, Dijkstra's algorithm can be implemented (using

so-called Fibonacci heaps) in time $O(n \log n + m)$ (see the paper [360] by Fredman and Tarjan).

Summarizing the discussion above we obtain

Theorem 3.3.7 *Dijkstra's algorithm determines the distances from s to all other vertices in time $O(n \log n + m)$.* □

Figure 3.3 illustrates Dijkstra's algorithm.

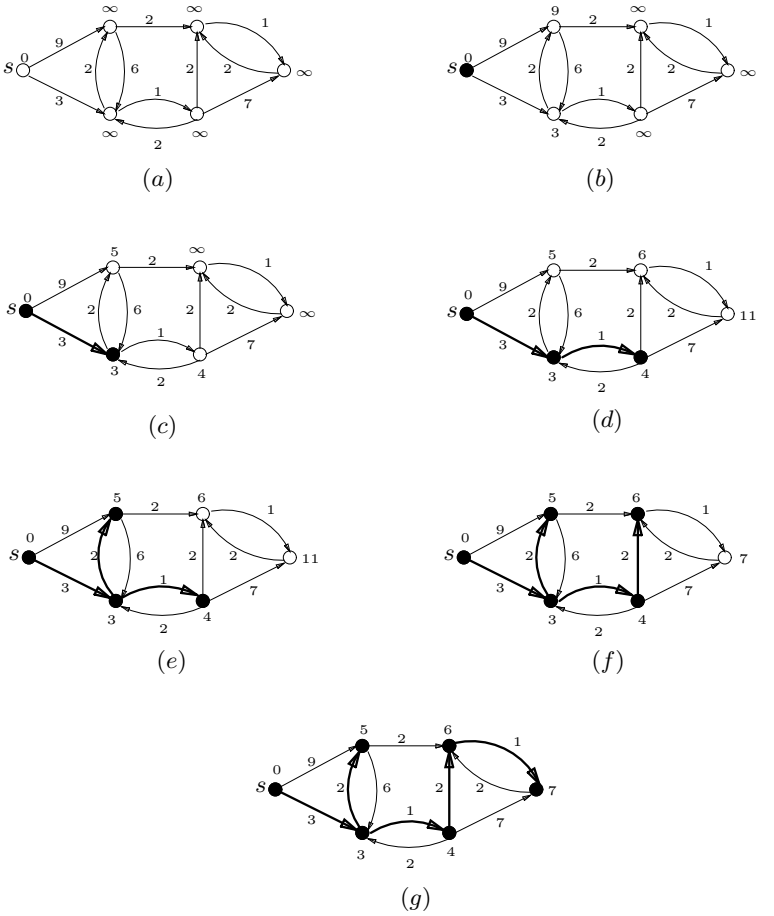


Figure 3.3 Execution of Dijkstra's algorithm. The white vertices are in Q; the black vertices are in P. The number above each vertex is the current value of the parameter δ. (a) The situation after performing the first step of the algorithm. (b)–(g) The situation after each successive iteration of the loop in the second step of the algorithm. The fat arcs indicate the corresponding shortest path tree found by the algorithm if extended as in Exercise 3.8.

It is a challenging open question whether there exists a linear algorithm for calculating the distances from one vertex to all other vertices in a given digraph with no negative cycles. It is easy to see that Dijkstra's algorithm sorts the vertices according to their distances from s. Fredman and Tarjan [360] showed that if Dijkstra's algorithm can be implemented as a linear time algorithm, then one can sort numbers in linear time. Thorup [870] showed that the opposite claim holds as well: if one can sort numbers in linear time, then Dijkstra's algorithm can be implemented as a linear time algorithm. Currently, no one knows how to sort in linear time[2].

In the case when D is the complete biorientation of an undirected graph G and $c(u,v) = c(v,u)$ holds for every arc uv of D, Thorup [871] recently gave a linear algorithm for calculating shortest paths from a fixed vertex to all other vertices. Thorup's algorithm avoids the sorting bottleneck by building a hierarchical bucketing structure, identifying vertex pairs that may be visited in any order.

3.3.4 The Bellman-Ford-Moore Algorithm

This algorithm originates from the papers [134] by Bellman, [330] by Ford and [705] by Moore. Let $D = (V, A, c)$ be a weighted digraph, possibly with arcs of negative weight. The algorithm described below can be applied to find the distances from a given vertex s in D to the rest of the vertices, provided D has no negative cycle.

Let $\delta(v,m)$ be the length of a shortest (s,v)-path that has at most m arcs. Clearly, $\delta(s,0) = 0$ and $\delta(v,0) = \infty$ for every $v \in V - s$. Let $v \in V$. We prove that for every $m \geq 0$,

$$\delta(v, m+1) = \min\{\delta(v,m), \min\{\delta(u,m) + c(u,v) : \ u \in N^-(v)\}\}. \qquad (3.4)$$

We show (3.4) by induction on m. For $m = 0$, (3.4) trivially holds. For $m \geq 1$, (3.4) is valid due to the following argument. Assume that there is a shortest (s,v)-path P with no more than $m+1$ arcs. If P has at most m arcs, its length is $\delta(v,m)$, otherwise P contains $m+1$ arcs and, by Proposition 3.2.1, consists of a shortest (s,u)-path with m arcs and the arc uv for some $u \in N^-(v)$. If every shortest (s,v)-path has more than $m+1$ arcs, then there is no in-neighbour u of v such that $\delta(u,m) < \infty$. Therefore, (3.4) implies correctly that $\delta(v, m+1) = \infty$.

Since no path has more than $n-1$ arcs, $\delta(v, n-1) = \mathrm{dist}(s,v)$ for every $v \in V - s$. Thus, using (3.4) for $m = 0, 1, \ldots, n-2$, we obtain the distances from s to the vertices of D. This results in the following algorithm.

[2] Some readers may be confused about this as they may know of a lower bound of $\Omega(n \log n)$ for sorting a set of n numbers. However, this lower bound is only valid for comparison based sorting. There are algorithms for sorting n numbers that are faster than $\Omega(n \log n)$, see e.g. the paper [41] by Andersson.

The Bellman-Ford-Moore algorithm
Input: A weighted digraph $D = (V, A, c)$ with no negative cycle, and a fixed vertex $s \in V$.
Output: The parameter δ_v for every $v \in V$ such that $\delta_v = \text{dist}(s, v)$ for all $v \in V$.

1. Set $\delta_s := 0$ and $\delta_v := \infty$ for every $v \in V - s$.
2. For $i = 1$ to $n-1$ do: for each $vu \in A$ update the parameter δ_u by setting $\delta_u := \min\{\delta_u, \delta_v + c(v, u)\}$.

It is easy to verify that the complexity of this algorithm is $O(nm)$. Hence we have

Theorem 3.3.8 *When applied to a weighted directed graph $D = (V, A, c)$ with no negative cycle and a fixed vertex $s \in V$, the Bellman-Ford-Moore algorithm correctly determines the distances from s to all other vertices in D in time $O(nm)$.* □

Figure 3.4 illustrates the execution of the Bellman-Ford-Moore algorithm.

Checking whether D has no negative cycle can be accomplished as follows. Let us assume that D is strong (otherwise, we will consider the strong components of D one by one; an effective algorithm to build the strong components is described in Chapter 2). Let us append the following additional step to the above algorithm:

3. For every arc $vu \in A$ do: if $\delta_u > \delta_v + c(vu)$ then return the message 'the digraph contains a negative cycle'.

Theorem 3.3.9 *A strong weighted digraph D has a negative cycle if and only if Step 3 returns its message.*

Proof: Suppose that D has no negative cycle. By the description of Step 2 and Proposition 3.2.1, $\delta_u \leq \delta_v + c(vu)$ for every arc $vu \in A$. Hence, the message will not be returned.

Assume that D has a negative cycle $Z = v_1 v_2 \dots v_k v_1$. Assume for the purpose of contradiction that Step 3 of the Bellman-Ford-Moore algorithm does not return the message. Thus, in particular, $\delta_{v_i} \leq \delta_{v_{i-1}} + c(v_{i-1}v_i)$ for every $i \in [k]$, where $v_0 = v_k$. Hence,

$$\sum_{i=1}^{k} \delta_{v_i} \leq \sum_{i=1}^{k} \delta_{v_{i-1}} + \sum_{i=1}^{k} c(v_{i-1}v_i).$$

Since the first two sums in the last inequality are equal, we obtain $0 \leq \sum_{i=1}^{k} c(v_{i-1}v_i) = c(Z)$; a contradiction. □

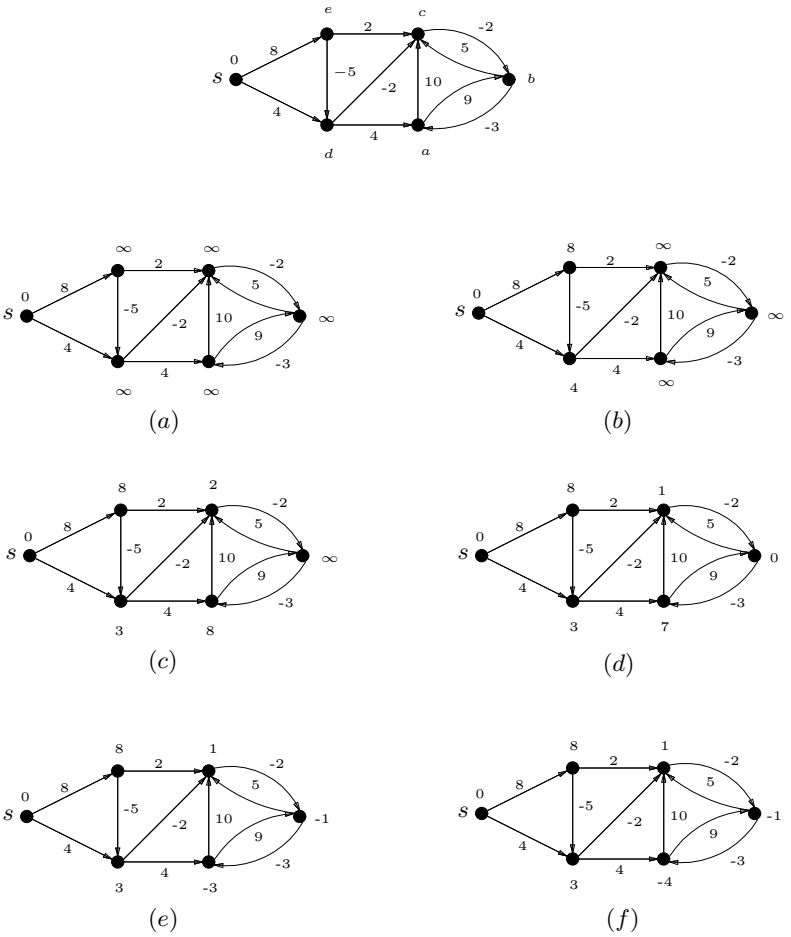


Figure 3.4 Execution of the Bellman-Ford-Moore algorithm. The vertex labellings and arc weights are given in the first digraph. The values of the parameter δ are given near the vertices of the digraphs (a)–(f). In the inner loop of the second step of the algorithm the arcs are considered in the lexicographic order: *ab*, *ac*, *ba*, *bc*, *cb*, *da*, *dc*, *ec*, *ed*, *sd*, *se*. (a) The situation after performing the first step of the algorithm. (b)–(f) The situation after each of the five successive executions of the inner loop in the second step of the algorithm.

3.3.5 The Floyd-Warshall Algorithm

The above algorithms can be run from all vertices to find *all* pairwise distances between the vertices of a strong digraph D. However, if D has negative weight arcs, but does not contain a negative cycle, we may only use the Bellman-Ford-Moore algorithm n times, which will result in $O(n^2m)$ time (see Exercise 3.19 for a faster method). The Floyd-Warshall algorithm will

find the required distances faster, in $O(n^3)$ time. According to Skiena [822], in practice, the algorithm even outperforms Dijkstra's algorithm applied from n vertices (when the weights in D are all non-negative) due to the simplicity of its code (and, thus, smaller hidden constants in the time complexity). The algorithm originates from the papers [324] by Floyd and [900] by Warshall. We assume that we are given a strong weighted digraph $D = (V, A, c)$ that has no negative cycle. In this subsection, it is convenient to assume that $V = [n]$.

Denote by δ_{ij}^m the length of a shortest (i,j)-path in $D\langle [m-1] \cup \{i,j\}\rangle$, for all $1 \leq m \leq n-1$. In particular, δ_{ij}^1 is the length of the path ij, if it exists. Observe that a shortest (i,j)-path in $D\langle [m] \cup \{i,j\}\rangle$ either does not include the vertex m, in which case $\delta_{ij}^{m+1} = \delta_{ij}^m$, or does include it, in which case $\delta_{ij}^{m+1} = \delta_{im}^m + \delta_{mj}^m$. Therefore,

$$\delta_{ij}^{m+1} = \min\{\delta_{ij}^m, \delta_{im}^m + \delta_{mj}^m\}. \tag{3.5}$$

Observe that $\delta_{ii}^m = 0$ for all $i \in [n]$, and, furthermore, for all pairs i, j such that $i \neq j$, $\delta_{ij}^1 = c(i,j)$ if $ij \in A$ and $\delta_{ij}^1 = \infty$, otherwise. Formula (3.5) is also correct when there is no (i,j)-path in $D\langle [m] \cup \{i,j\}\rangle$. Clearly, δ_{ij}^{n+1} is the length of a shortest (i,j)-path (in D). It is also easy to verify that $O(n^3)$ operations are required to compute δ_{ij}^{n+1} for all pairs i, j.

The above assertions can readily be implemented as a formal algorithm (the Floyd-Warshall algorithm, see Exercise 3.14). The Floyd-Warshall algorithm allows one to find the diameter and radius of a weighted digraph without cycles of negative weight in $O(n^3)$ time. Using the algorithm, we may check whether D has no negative cycle. For simplicity let us assume, as above, that D is strong. Then the verification can be based on the following theorem (see, e.g., Lawler's book [636]) whose proof is left to the interested reader as Exercise 3.15.

Theorem 3.3.10 *A weighted digraph D has a negative cycle if and only if $\delta_{ii}^m < 0$ for some $m, i \in [n]$.* □

3.4 Inequalities on Diameter

For a network representing a certain real-world system, it is desirable to have a small diameter as it increases the reliability of the system (see, e.g., Fiol, Yebra and Alegre [316]). However, networks representing real-world systems normally do not have many arcs to avoid too costly constructions. The objectives of minimizing the diameter and the size of a digraph clearly contradict each other. Therefore, it is important for a designer to know what kind of trade-off can be achieved. The inequalities of this section give some insight into this problem.

It is well-known that, in a connected undirected graph G, we have $\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2\,\mathrm{rad}(G)$. This inequality holds also for strong digraphs (for our definition of radius).

Proposition 3.4.1 *For a strong digraph $D = (V, A)$, we have* $\mathrm{rad}(D) \leq \mathrm{diam}(D) \leq 2\,\mathrm{rad}(D)$.

Proof: Clearly, $\mathrm{rad}(D) \leq \mathrm{diam}(D)$. Let x be a vertex of D such that $(\mathrm{dist}(x, V) + \mathrm{dist}(V, x))/2 = \mathrm{rad}(D)$, and let y, z be vertices of D such that $\mathrm{dist}(y, z) = \mathrm{diam}(D)$. Since $\mathrm{dist}(y, z) \leq \mathrm{dist}(y, x) + \mathrm{dist}(x, z) \leq 2\,\mathrm{rad}(D)$, we conclude that $\mathrm{diam}(D) \leq 2\,\mathrm{rad}(D)$. □

The following simple bound (called the **Moore bound**) on the order of a strong digraph is important in certain applications [316]. We leave its proof to the reader (Exercise 3.23).

Proposition 3.4.2 *Let n, Δ and d be the order, the maximum out-degree and the diameter, respectively, of a strong digraph D. Then $n \leq 1 + \Delta + \Delta^2 + \cdots + \Delta^d$.* □

The Moore bound is attained for $\Delta = 1$ by the cycle $\vec{C}_{d+1}$ and for $d = 1$ by the complete digraph on $\Delta + 1$ vertices. However, it is well-known (see Bridges and Toueg [181] and Plesník and Znám [752]) that this bound cannot be attained for $\Delta > 1$ and $d > 1$. Therefore,

$$n < \frac{\Delta^{d+1} - 1}{\Delta - 1}$$

for $\Delta > 1$ and $d > 1$. After simple algebraic transformations, we obtain the following:

Proposition 3.4.3 *Let n, Δ and d be the order, the maximum out-degree and the diameter, respectively, of a strong digraph D. If $\Delta > 1$ and $d > 1$, then $d \geq \lfloor \log_\Delta(n(\Delta - 1) + 1) \rfloor$.* □

The cases $d = 2, 3$ have received special consideration. For $\Delta = 2$, Miller and Fris [699] proved that there is no 2-regular digraph of diameter $d \geq 3$ and order $n = \Delta + \Delta^2 + \cdots + \Delta^d$. 3-regular digraphs of order $n = \Delta + \Delta^2 + \cdots + \Delta^d$, with $\Delta = 3$, have been studied by Baskoro, Miller, Plesník and Znám [126].

The following two theorems solve the problems of establishing lower and upper bounds for the diameter of a strong digraph. Theorem 3.4.4 was proved by Goldberg [418]; Theorem 3.4.5 was derived by Ghouila-Houri [402].

Theorem 3.4.4 *Let D be a strong digraph of order n and size m, $m \geq n+1$, and let $g(n, m) = \lceil \frac{2n-2}{m-n+1} \rceil$. Then* $\mathrm{diam}(D) \geq g(n, m)$. *This bound is the best possible.* □

Theorem 3.4.5 *Let D be a strong digraph of order n and size m. Then* $\mathrm{diam}(D) \leq n-1$, *if* $n \leq m \leq (n^2+n-2)/2$ *and, otherwise, we have* $\mathrm{diam}(D) \leq \lfloor n + \frac{1}{2} - \sqrt{2m - n^2 - n + \frac{17}{4}} \rfloor$, *otherwise.* □

Now we consider a more refined upper bound on the diameter of an eulerian digraph obtained by Soares [826].

Theorem 3.4.6 *Let $D = (V, A)$ be an eulerian digraph of order n, diameter d and minimum in-degree r, and let*

$$f(n, r, t) = \max\left\{2,\ t + 3\left(\frac{n-t}{r+1} - 1\right)\right\},$$

where $t = d \bmod 3$. Then $d \leq f(n, r, t)$.

Proof: We may assume that $d \geq 3$. Let $v, w \in V$ such that $\mathrm{dist}(v, w) = d$ and let $V_i = N^{+i}(v)$ for $i = 0, 1, \ldots, d$. Clearly, $V_0, V_1, \ldots, V_d$ is a partition of V. Consider three consecutive sets V_{j-1}, V_j, V_{j+1} of the partition. Recall that (X, Y) denotes the set of arcs with tail in X and head in Y. By the definition of the partition, we have

$$(V_j, V) = (V_j, V_j) \cup (V_j, V_{j+1}) \cup (V_j, W), \tag{3.6}$$

where $W = V_0 \cup V_1 \cup \ldots \cup V_{j-1}$. By Corollary 1.7.3, $|(V_j, W)| \leq |(V \setminus W, W)| = |(W, V \setminus W)|$. By the definition of the partition $(W, V \setminus W) = (V_{j-1}, V_j)$. Since the minimum out-degree of D is r, we have $|(V_j, V)| \geq r|V_j|$. Also, we have

$$|(V_j, V_j)| \leq |V_j|(|V_j| - 1),\ |(V_j, V_{j+1})| \leq |V_j||V_{j+1}|,\ |(V_{j-1}, V_j)| \leq |V_{j-1}||V_j|.$$

Therefore it follows from (3.6) that

$$r|V_j| \leq |V_j|(|V_j| - 1) + |V_j||V_{j+1}| + |V_{j-1}||V_j|.$$

Thus,

$$|V_{j-1}| + |V_j| + |V_{j+1}| \geq r + 1. \tag{3.7}$$

Since $d^+(v) \geq r$ and $d^-(w) \geq r$, we have $|V_0| + |V_1| \geq r+1$ and $|V_{d-1}| + |V_d| \geq r+1$. Adding these two inequalities to inequality (3.7) for $j = 3, 6, \ldots, 3\lfloor (d-3)/3 \rfloor$ and to the inequalities $|V_j| \geq 1$ for the remaining $t = d \bmod 3$ sets V_j, we obtain

$$n = \sum_{j=0}^{d} |V_j| \geq (r+1)\left(\frac{d-t}{3} + 1\right) + t,$$

which implies the desired inequality. □

Soares [826] showed that the bound of Theorem 3.4.6 is best possible by constructing a strong r-regular digraph D of order n with diameter $\lfloor f(n, r, t) \rfloor$ for all integers n and r such that $n - 1 > r \geq 2$.

For oriented digraphs the upper bound of Theorem 3.4.6 can be improved. Let D be an eulerian oriented graph of order n. Dankelmann [240] proved that that $\mathrm{diam}(D) \leq \frac{4n}{2\delta^0(D)+1} + 2$. For given n and δ^0, Knyazev [603] constructed strong δ^0-regular oriented graphs of order n and of diameter larger than $\frac{4n}{2\delta^0+1} - 4$. This shows that Dankelmann's upper bound is sharp modulo an additive constant.

For a strong oriented graph D, an obvious upper bound on its diameter is $\mathrm{diam}(D) \leq \mathrm{lp}(UG(D))$, where $\mathrm{lp}(G)$ denotes the length of a longest path in a graph G. This bound is sharp due to the following result by Gutin [455]. A short proof of this result was given by Bondy [166].

Theorem 3.4.7 *Let G be a connected bridgeless graph. Then*

$$\max\{\mathrm{diam}(D) : \ D \in \mathcal{S}(G)\} = \mathrm{lp}(G),$$

where $\mathcal{S}(G)$ is the set of strong orientations of G. □

Oriented graphs of diameter 2 and minimum size (for fixed order n) were discussed by Fűredi, Horak, Pareek and Zhu [368]. (If we consider digraphs instead of oriented graphs, the minimum size can be found trivially: it is attained by $\overleftrightarrow{K}_{1,n-1}$.) Let $f(n)$ be the minimum size of a strong oriented graph of diameter 2 and order n. The authors of [368] proved that

$$(1 - o(1))n \log_2 n \leq f(n) \leq n \log_2 n + O(n \log_2 \log_2 n).$$

They stated the following:

Conjecture 3.4.8 *We have $f(n) \geq n \log_2 n + (\frac{1}{2} + o(1))n \log_2 \log_2 n$.*

We will finish this section by the following conjecture of Shen [815].

Conjecture 3.4.9 *Let D be a strong digraph of order n, girth g and minimum out-degree at least r. Then* $\mathrm{diam}(D) \leq n - (r-1)(g-1) - 1$.

Shen [815] showed that the bound in Conjecture 3.4.9 cannot be decreased. The conjecture is trivial for either $r = 1$ or $g = 2$. Shen [815] proved the conjecture for $r = 2$. It is easy to see that Conjecture 3.4.9 implies the conjecture of Caccetta and Häggkvist (Conjecture 8.4.1).

3.5 Minimum Diameter of Orientations of Multigraphs

Readers may find the following complexity result surprising.

Theorem 3.5.1 (Chvátal and Thomassen) *[224] It is $\mathcal{NP}$-complete to decide whether an undirected graph admits an orientation of diameter 2.* □

For a bridgeless multigraph G, let $\mathrm{diam}_{\min}(G)$ denote the minimum diameter of an orientation of G. We will present a minor modification of the original proof of Theorem 3.5.1 by Chvátal and Thomassen [224]. The main difference is in the use of Lemma 3.5.2 (which is applied to two different results in this section). Define a bipartite tournament BT_s, with partite sets U, W, each of cardinality s, as follows. Let $U = \{u_1, u_2, \ldots, u_s\}$ and $W = \{w_1, w_2, \ldots, w_s\}$. The vertex u_i dominates only vertices $w_i, w_{i+1}, \ldots, w_{i+\lfloor s/2 \rfloor - 1}$ (the subscripts are taken modulo s) for every $i \in [s]$.

Lemma 3.5.2 *Let $s \geq 2$. The diameter* $\mathrm{diam}(BT_s)$ *equals 3. In particular,* $\mathrm{dist}(U, U) = \mathrm{dist}(W, W) = 2$.

Proof: Clearly, it suffices to show that $\mathrm{dist}(U, U) = \mathrm{dist}(W, W) = 2$. This follows from the fact that, for every $i \neq j$, we have $N^+(u_i) - N^+(u_j) \neq \emptyset$ and, hence, there is a vertex $w \in W$ such that $u_i \rightarrow w \rightarrow u_j$. □

Lovász [653] proved that it is $\mathcal{NP}$-hard to decide whether a hypergraph of rank[3] 3 is 2-colourable. By the result of Lovász, Theorem 3.5.1 follows from the next theorem.

Theorem 3.5.3 *Given a hypergraph H of rank 3 and order n, one can construct in polynomial time (in n) a graph G such that* $\mathrm{diam}_{\min}(G) = 2$ *if and only if H is 2-colourable.*

Proof: Let k be the integer satisfying $8 \leq k \leq 11$ and $n + k$ is divisible by 4. Let H_0 be a hypergraph obtained from H by adding k new vertices $v_1, \ldots, v_k$. Moreover, append three new edges $\{\{v_i, v_{i+1}\} : i = 1, 2, 3\}$ to H_0 if H has an odd number of edges, and add four new edges $\{\{v_i, v_{i+1}\} : i = 1, 2, 3, 4\}$ to H_0 otherwise. Observe that H_0 has an even number of edges, which is at least four. To construct G, take disjoint sets R and Q such that the elements of R (Q) are in a one-to-one correspondence with the vertices (the edges) of H_0. Let $G\langle R \rangle$ and $G\langle Q \rangle$ be complete graphs, and $p \in R$ and $q \in Q$ be adjacent if and only if the vertex corresponding to p belongs to the edge corresponding to q (in H_0).

Append four new vertices w_1, w_2, w_3, w_4 and join each of them to all the vertices in $R \cup Q$. Finally, add a new vertex x and join it to all the vertices in R. We show that the obtained graph G has the desired property. (Clearly, G can be constructed in polynomial time.)

Assume that G admits an orientation G^* of diameter 2. For a vertex $u \in R$, set $f(u) = 0$ if and only if $x \rightarrow u$ in G^*; otherwise, $f(u) = 1$. Since $\mathrm{dist}_{G^*}(x, q) = 2$ ($\mathrm{dist}_{G^*}(q, x) = 2$, respectively) for each $q \in Q$, every edge e of H contains a vertex y such that $f(y) = 0$ ($f(y) = 1$, respectively). Thus H is 2-colourable.

Now assume that H is 2-colourable. Then H_0 admits a 2-colouring which generates a partition $R = R_1 \cup R_2$ such that every edge of H_0 has a vertex

[3] Recall that the rank of a hypergraph is the cardinality of its largest edge.

corresponding to an element from R_i and $|R_i| \geq 4$ (for every $i = 1, 2$). An orientation G' of G of diameter 2 is defined as follows. Orient the edges in each complete graph $G\langle L\rangle \in \{G\langle R_1\rangle, G\langle R_2\rangle, G\langle Q\rangle\}$ such that the resulting tournament contains the bipartite tournament $BT_{|L|}$. Let A_i, B_i be the partite sets of the bipartite tournaments in $G\langle R_i\rangle$ $(i = 1, 2)$ and let A, B be the partite sets of the bipartite tournament in $G\langle Q\rangle$. The rest of the edges in G are oriented as follows:

$$\begin{array}{ll} x \to R_1 \to R_2 \to x, & R_1 \to Q \to R_2, \\ (A_1 \cup A_2) \to w_1 \to A, & B \to w_1 \to (B_1 \cup B_2), \\ (A_1 \cup A_2) \to w_2 \to B, & A \to w_2 \to (B_1 \cup B_2), \\ (B_1 \cup B_2) \to w_3 \to A, & B \to w_3 \to (A_1 \cup A_2), \\ (B_1 \cup B_2) \to w_4 \to B, & A \to w_4 \to (A_1 \cup A_2). \end{array}$$

Using Lemma 3.5.2, it is not difficult to verify that $\mathrm{diam}(G') = 2$. For example, to show that $\mathrm{dist}_{G'}(A_1, V(G')) \leq 2$ and $\mathrm{dist}_{G'}(V(G'), A_1) \leq 2$, it suffices to observe that $\mathrm{dist}_{G'}(A_1, A_1) = 2$ and

$$\begin{aligned} B_1 \cup R_2 \cup Q \cup \{w_1, w_2\} &\subseteq N^+(A_1), \\ \{x, w_3, w_4\} &\subseteq N^+(B_1 \cup R_2 \cup Q \cup \{w_1, w_2\}), \\ B_1 \cup \{x, w_3, w_4\} &\subseteq N^-(A_1), \\ N^-(B_1 \cup \{x, w_3, w_4\}) &\subseteq R_2 \cup Q \cup \{w_1, w_2\}. \end{aligned}$$

□

Chvátal and Thomassen [224] dealt with the following parameter which we call the **strong radius**. The strong radius of a strongly connected digraph $D = (V, A)$, $\mathrm{srad}(D)$, is equal to

$$\min\{\max\{\mathrm{dist}(v, V), \mathrm{dist}(V, v)\} : \ v \in V\}.$$

Chvátal and Thomassen [224] showed that it is $\mathcal{NP}$-hard to decide whether a graph admits a strongly connected orientation of strong radius 2. The strong radius is of interest because, in particular, $\mathrm{srad}(D) \leq \mathrm{diam}(D) \leq 2\,\mathrm{srad}(D)$ for every strongly connected digraph D (this follows from the fact that $\mathrm{rad}(D) \leq \mathrm{srad}(D)$ for every strong digraph D and Proposition 3.4.1). Following [224], we prove a sharp upper bound for the value of the strong radius of a strong orientation of a bridgeless connected multigraph. The first part of the proof of Theorem 3.5.4 is illustrated in Figure 3.5.

Theorem 3.5.4 *[224] Every bridgeless connected multigraph $G = (V, E)$ admits an orientation of strong radius at most* $(\mathrm{rad}(G))^2 + \mathrm{rad}(G)$.

Proof: We will show a slightly more general result. Let $u \in V$ be arbitrary and let $\mathrm{dist}_G(u, V) = r$, then there is an orientation L of G such that $\mathrm{dist}_L(u, V) \leq r^2 + r$ and $\mathrm{dist}_L(V, u) \leq r^2 + r$.

Since G is bridgeless, every edge uv is contained in some undirected cycle; let $k(v)$ denote the length of a shortest cycle through uv. It is not difficult to prove (see Exercise 3.24) that for every $v \in N(u)$, $k(v) \leq 2r + 1$.

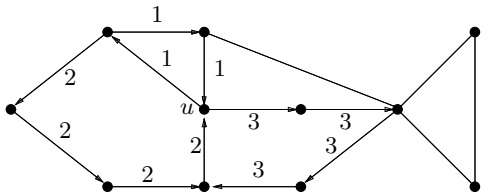


Figure 3.5 Constructing the orientation D of H in the proof of Theorem 3.5.4. The integers on arcs indicate the step number in the process of obtaining D.

We claim that there is a subgraph H of G and an orientation D of H with the following properties:

(a) $N_G(u) \subseteq V(H)$.
(b) For each $v \in N(u)$, D has a cycle C_v of length $k(v)$ containing either uv or vu.
(c) D is the union of the cycles C_v.

Observe that by this claim and $k(v) \leq 2r + 1$,

$$\max\{\text{dist}_D(u, V(D)), \text{dist}_D(V(D), u)\} \leq 2r. \tag{3.8}$$

We demonstrate the above claim by constructing H and D step by step. Let uv be an edge in G and let Z_v be an undirected cycle of length $k(v)$ through uv. Orient Z_v arbitrarily as a directed cycle and let C_v denote the cycle obtained this way. Set $H := Z_v$, $D := C_v$. This completes the first step. At step $i(\geq 2)$, we choose an edge uw such that $w \notin V(H)$ and an undirected cycle $Z = w_1w_2 \dots w_kw_1$ in G such that $w_1 = u$, $w_2 = w$ and $k = k(w)$. If no vertex in $Z_w - u$ belongs to H, then append the directed cycle $C_w = w_1w_2 \dots w_kw_1$ to D and the cycle Z to H. Go to the next step.

Otherwise, there is a vertex w_i $(2 \leq i \leq k)$ such that $w_i \in V(H)$ (and hence $w_i \in V(D)$). Suppose that w_i has the least possible subscript with this property. Since $w_i \in V(D)$, there is some neighbour v of u such that $w_i \in C_v$. (Recall that C_v is a directed cycle.) Let $C_v = v_1v_2 \dots v_tv_1$, where $u = v_1$, $v \in \{v_2, v_t\}$ and $w_i = v_j$ for some j. By considering the converse of D, if necessary, we may assume, without loss of generality, that $v = v_2$. Now we consider two cases.

Case 1: $w_k \neq v$. In this case, define the directed cycle $C_w = uw_2w_3 \dots w_iC_v[v_{j+1}, u]$ and observe that C_w has length $k(w)$. (Indeed, if C_w had more than $k(w)$ arcs, the path $C_w[w_i, u]$ would be longer than the path $P_2 = w_iw_{i+1} \dots w_ku$. In that case, the walk $Z_v[u, v_j]P_2[w_{i+1}, u]$ containing uv would be of length less than $k(v)$; a contradiction.) Let $Z_w := UG(C_w)$. Add C_w to D and Z_w to H. Go to the next step.

Case 2: $w_k = v$. In this case, define the directed cycle C_w as follows: $C_w = C_v[u, v_j]w_{i-1}w_{i-2} \dots w_2u$ and observe that C_w has length $k(w)$ (the proof of

the last fact is similar to the one given in Case 1). Let $Z_w := UG(C_w)$. Add C_w to D and Z_w to H. Go to the next step.

Since $V(G)$ is finite and we add at least one new vertex to H at each step, this process will terminate with the desired subgraph H and its orientation D. Thus, the claim is proved.

Consider the directed multigraph D. In G, contract all the vertices of D into a new vertex u^* (the operation of contraction for undirected multigraphs is similar to that for directed multigraphs) and call the resulting multigraph G^*. Note that G^* is bridgeless and that by the property (a) of the above claim, we obtain $\mathrm{dist}_{G^*}(u^*, V(G^*)) \leq r - 1$. By the induction hypothesis, there is an orientation L^* of G^* such that

$$\mathrm{dist}_{L^*}(u^*, V(L^*)) \leq r^2 - r \text{ and } \mathrm{dist}_{L^*}(V(L^*), u^*) \leq r^2 - r. \tag{3.9}$$

Consider an orientation L of G obtained by combining L^* with D and orienting the rest of the edges in G arbitrarily. By (3.8) and (3.9), we have

$$\mathrm{dist}_L(u, V(L)) \leq r^2 + r \text{ and } \mathrm{dist}_L(V(L), u) \leq r^2 + r.$$

□

The sharpness of the bound in Theorem 3.5.4 is proved in [224]. Theorem 3.5.4 immediately implies the following.

Corollary 3.5.5 *For every bridgeless connected multigraph G of radius r,* $\mathrm{diam}_{\min}(G) \leq 2r^2 + 2r$. □

Plesník [751] generalized Theorem 3.5.4 and Corollary 3.5.5 to orientations of weighted multigraphs.

Theorem 3.5.6 *Let G be a bridgeless connected multigraph in which every edge has weight between 1 and W. If the radius of G is r, then G admits an orientation of strong radius at most $r^2 + rW$ and of diameter at most $2r^2 + 2rW$.* □

Plesník [751] showed that the result of the previous theorem regarding the strong radius is sharp.

Chung, Garey and Tarjan [218] generalized Corollary 3.5.5 to mixed graphs. They proved the following.

Theorem 3.5.7 *Every bridgeless connected mixed graph G of radius r admits an orientation of diameter at most $8r^2 + 8r$. Such an orientation can be found in time $O(r^2(n + m))$.* □

3.6 Minimum Diameter Orientations of Some Graphs and Digraphs

Recall that an **orientation** of a digraph D is an oriented graph H obtained from D by deleting one arc in every 2-cycle of D. For a strong digraph D with bridgeless $UG(D)$, let $\mathrm{diam}_{\min}(D)$ denote the minimum diameter of an orientation of D. By Corollary 1.8.2, the assumption that $UG(D)$ is bridgeless implies that $\mathrm{diam}_{\min}(D) < \infty$. (If $UG(D)$ has a bridge, no orientation of D is strong.)

Many authors consider the following parameter $\rho(D) := \mathrm{diam}_{\min}(D) - \mathrm{diam}(D)$. It turns out that, for many interesting directed and undirected graphs G, $\rho(G) = 0$, 1 or 2 (a result which is quite different from the 'pessimistic' bound proved in Theorem 3.5.4). In this section, we discuss results on minimum diameter orientations of some special families of directed and undirected graphs G for which $\rho(G)$ is (very) small.

3.6.1 Generalizations of Tournaments

Recall that a digraph is semicomplete k-partite if it can be obtained from a complete k-partite graph by replacing each edge xy with either arc xy or arc yx or both xy and yx. Recall that a digraph D is quasi-transitive if the existence of arcs xy, yz $(x \neq z)$ in D implies that at least one of the arcs xz and zx is in D.

Observe that, by Corollary 1.8.2, a strong semicomplete k-partite digraph D, $k \geq 2$, has a strong orientation unless D is a semicomplete bipartite digraph with a partite set consisting of a single vertex. This justifies the consideration of the following two classes of digraphs. Let $\mathcal{D}_0$ be the set of strong quasi-transitive digraphs of order $n \geq 3$. Let $\mathcal{D}_1$ be the set of strong semicomplete bipartite digraphs with at least two vertices in each partite set.

For digraphs from the class $\mathcal{D}_0 \cup \mathcal{D}_1$ the following bound on the minimum diameter of an orientation was obtained by Gutin and Yeo [478].

Theorem 3.6.1 *If $D \in \mathcal{D}_i$ for $i \in \{0, 1\}$, then*

$$\mathrm{diam}_{\min}(D) \leq \max\{3 + 2i, \mathrm{diam}(D)\}.$$

Proof: Assume that this theorem is false and that D is a counterexample to the theorem with as few 2-cycles as possible. Let $D \in \mathcal{D}_i$ for $i \in \{0, 1\}$ and let $\gamma = 3 + 2i$. Let xyx be a 2-cycle in D. Clearly, the diameter of D increases by at least one when we delete either of the arcs xy or yx from D. Therefore, there exist vertices $s_{xy}, t_{xy}, s_{yx}, t_{yx}$ in D, such that $\mathrm{dist}_{D-xy}(s_{xy}, t_{xy}) > \max\{\gamma, \mathrm{diam}(D)\}$ and $\mathrm{dist}_{D-yx}(s_{yx}, t_{yx}) > \max\{\gamma, \mathrm{diam}(D)\}$. Let $P = p_0 p_1 \dots p_l$ be an (s_{xy}, t_{xy})-path in D of minimum length (note that $l \leq \mathrm{diam}(D)$) and let $Q = q_0 q_1 \dots q_m$ be an (s_{yx}, t_{yx})-path in D of minimum length (note that $m \leq \mathrm{diam}(D)$). Let ρ and η be defined

such that $xy = p_\rho p_{\rho+1}$ and $yx = q_\eta q_{\eta+1}$. We now consider the following cases, which exhaust all possibilities:

Case 1: $\rho+1 < l$, $\eta+1 < m$ **and** $D \in \mathcal{D}_0 \cup \mathcal{D}_1$**.** We first show that $p_{\rho+2}$ and $q_{\eta+2}$ are adjacent. This is clearly true if D is semicomplete bipartite as these two vertices belong to different partite sets of D. If D is quasi-transitive, then p_ρ and $p_{\rho+2}$ are adjacent. Therefore, $p_{\rho+2} \rightarrow p_\rho$ by the minimality of l. However, this implies that $p_{\rho+2}$ and $q_{\eta+2}$ are adjacent, as $p_{\rho+2} \rightarrow (p_\rho = q_{\eta+1}) \rightarrow q_{\eta+2}$.

If $p_{\rho+2} \rightarrow q_{\eta+2}$, then by $q_\eta = p_{\rho+1}$, $q_0 q_1 \dots q_\eta p_{\rho+2} q_{\eta+2} \dots q_m$ is a (q_0, q_m)-path of length $m \leq \operatorname{diam}(D)$ in $D - yx$, a contradiction. The case when $q_{\eta+2} \rightarrow p_{\rho+2}$ can be considered analogously.

Case 2: $\rho > 0$, $\eta > 0$ **and** $D \in \mathcal{D}_0 \cup \mathcal{D}_1$**.** This case can be transformed into Case 1 by considering the converse of D.

Case 3: $\rho = 0$, $\eta + 1 = m$ **and** $D \in \mathcal{D}_0$. We first prove that $l + m \geq 3$. Suppose that $l = m = 1$, i.e., $x = p_0 = q_1, y = p_1 = q_0$. Let $z_0 z_1 \dots z_k$ be a shortest (y, x)-path in $D - yx$. By the choice of x, y, we have $k \geq 4$. By Proposition 2.7.1, $z_k \rightarrow z_1$ and $z_2 \rightarrow z_0$. Hence, $z_k z_1 z_2 z_0$ is an (x, y)-path in $D - xy$ of length three, a contradiction with the choice of x. Thus, we may assume, without loss of generality, that $l \geq 2$.

Let $R = r_0 r_1 \dots r_t$ be a shortest path from q_0 to p_l in D. The path R can be chosen such that it does not contain yx. Indeed, if $y = r_j, x = r_{j+1}$ for some j, then $r_0 r_1 \dots r_j p_2 p_3 \dots p_l$ is not longer than R (as $p_1 p_2 \dots p_l$ is a shortest (p_1, p_l)-path in D). So, we may assume that R does not contain yx. It is not difficult to see that we may assume that R does not contain xy either.

By Proposition 2.7.1, we obtain immediately that $p_l \rightarrow p_0$ if $l \neq 3$ and $p_l \rightarrow p_1$ if $l = 3$. If $l = 3$, then we have $p_3 \rightarrow p_1$ and $p_0 \rightarrow p_1$. Therefore, by the minimality of l, $p_3 \rightarrow p_0$. Hence, we have shown that $p_l \rightarrow p_0$ for every $l \geq 2$.

We have $t > 2$, for otherwise $r_0 r_1 \dots r_t p_0$ would be a path from q_0 to q_m of length $t + 1 \leq 3$ in $D - yx$. Since $p_l \rightarrow p_0$ and $r_{t-1} \rightarrow r_t = p_l$, we conclude that r_{t-1} and p_0 are adjacent. If $r_{t-1} \rightarrow p_0$, then $r_0 r_1 \dots r_{t-1} p_0$ is a path from q_0 to q_m of length $t \leq \operatorname{diam}(D)$ in $D - yx$, a contradiction. If $p_0 \rightarrow r_{t-1}$, then $p_0 r_{t-1} p_l$ is a path of length two from p_0 to p_l in $D - xy$, a contradiction.

Case 4: $\eta = 0$, $\rho + 1 = l$ **and** $D \in \mathcal{D}_0$. This case can be transformed into Case 3 by considering the converse of D.

Case 5: $\rho = 0$, $\eta + 1 = m$ **and** $D \in \mathcal{D}_1$. Suppose that $l = m = 1$. Let $z_0 z_1 \dots z_k$ be a shortest (y, x)-path in $D - yx$. By the choice of x, y, $k \geq 6$. By the minimality of k, $z_3 \rightarrow z_0$ (z_0 and z_3 belong to different partite sets of D) and $z_k \rightarrow z_2$ (z_k and z_2 belong to different partite sets of D). Hence, $z_k z_2 z_3 z_0$ is an (x, y)-path in $D - xy$, a contradiction. So, we may assume, without loss of generality, that $m \geq 2$.

Let $R = r_0 r_1 \dots r_t$ be a shortest path from q_0 to p_l in D. As in Case 3, we may assume that R contains neither xy nor yx.

Suppose that $t = 0$, implying that $q_0 = p_l$ and $l, m \geq 2$. Assume that $l \geq 3$. If p_0 and p_l belong to different partite sets of D, then, by the minimality of l and the assumption that D is semicomplete bipartite, $p_l \rightarrow p_0$, which is impossible as $p_l p_0$ is a (q_0, q_m)-path of length one in $D - yx$, a contradiction. If p_0 and p_l belong to the same partite set of D, then $p_l \rightarrow p_1$ (by the minimality of l) and $p_l p_1 p_2 p_3 p_0$ is a (q_0, q_m)-path of length four in $D - yx$, a contradiction. So, $l = 2$. Analogously, we can prove that $m = 2$. Since $D - xy$ has a (p_0, p_2)-path and $p_2 = q_0 \rightarrow q_1 = p_1$, there is a (p_0, p_1)-path $S = s_0 s_1 \dots s_a$ in $D - xy$. Assume that S has minimum length and observe that $a \geq 5$, as $s_0 s_1 \dots s_a p_l$ is a (p_0, p_l)-path in $D - xy$. Furthermore, $s_3 \rightarrow s_0$ as s_0 and s_3 lie in different partite sets of D and S is of minimum length. Observe that if $p_2 \rightarrow s_3$, then $p_2 s_3 s_0$ is a (q_0, q_m)-path in $D - yx$ of length 2, and if $s_3 \rightarrow p_2$, then $s_0 s_1 s_2 s_3 p_2$ is a (p_0, p_l)-path in $D - xy$ of length 4. In both cases we obtain a contradiction. Hence, $t > 0$.

Suppose that $1 \leq t \leq 2$. Clearly r_0 and r_1 lie in different partite sets, so we may assume, without loss of generality, that r_0 and p_0 are adjacent (the case when r_1 and p_0 are adjacent can be considered analogously). Clearly p_0 dominates r_0 by the minimality of m. However, $p_0 r_0 \dots r_t$ is a (p_0, p_l)-path in $D - xy$ of length of $t + 1 \leq 3$, a contradiction. Hence, $t \geq 3$.

Clearly r_1 and r_2 lie in different partite sets, so we may assume, without loss of generality, that r_1 and p_0 are adjacent (the case when r_2 and p_0 are adjacent can be considered analogously). Clearly p_0 dominates r_1 by the minimality of m. However, the path $p_0 r_1 \dots r_t$ in $D - xy$ is of length $t \leq \operatorname{diam}(D)$.

Case 6: $\eta = 0$, $\rho + 1 = l$ and $D \in \mathcal{D}_1$. This case can be transformed into Case 5 by considering the converse of D. □

The upper bound of this theorem is sharp as one can see from the following examples given in [478]. Let T_k, $k \geq 3$, be a (transitive) tournament with vertices $x_1, x_2, \dots, x_k$ and arcs $x_i x_j$ for every $1 \leq i < j \leq k$. Let y be a vertex not in T_k, which dominates all vertices of T_k but x_k and is dominated by all vertices of T_k but x_1. The resulting semicomplete digraph D_{k+1} has diameter 2. However, the deletion of any arc of D_{k+1} between y and the set $\{x_2, x_3, \dots, x_{k-1}\}$ leaves a digraph with diameter 3. Indeed, if we delete yx_i, $2 \leq 2 \leq k - 1$, then a shortest (x_k, x_i)-path becomes of length 3.

Let H be a strong semicomplete bipartite digraph with the following partite sets V_1 and V_2 and arc set A: $V_1 = \{x_1, x_2, x_3\}$, $V_2 = \{y_1, y_2, y_3\}$ and

$$A = \{x_1 y_1, y_1 x_1, x_1 y_2, y_3 x_1, x_2 y_1, y_2 x_2, y_3 x_2, y_1 x_3, x_3 y_3, x_3 y_2\}.$$

Let $H' = H - x_1 y_1$ and $H'' = H - y_1 x_1$. It is easy to verify that $\operatorname{diam}(H) = 4$ (in particular, $\operatorname{dist}(y_2, y_3) = 4$) and that $\operatorname{diam}(H') = \operatorname{diam}(H'') = 5$ (a shortest (x_1, y_3)-path in H' and a shortest (y_2, x_1)-path in H'' are of length 5).

The digraph H can be used to generate an infinite family of semicomplete bipartite digraphs with the above property: replace, say, x_3 by a set of independent vertices.

The above theorem inspired the following conjecture by Gutin, Koh, Tay and Yeo [467].

Conjecture 3.6.2 *There is an absolute constant c such that for every strong semicomplete multipartite digraph D, we have* $\mathrm{diam}_{\min}(D) \leq \mathrm{diam}(D) + c$.

Recall that a digraph D is locally semicomplete if both of the digraphs $D\langle N^+(x)\rangle$ and $D\langle N^-(x)\rangle$ are semicomplete digraphs for every vertex $x \in V(D)$. The following theorem is an analog of Theorem 3.6.1 and was also proved by Gutin and Yeo [478].

Theorem 3.6.3 *If D is a strong locally semicomplete digraph of order $n \geq 3$, then*

$$\mathrm{diam}_{\min}(D) \leq \max\{5, \mathrm{diam}(D) + 1\}.$$

□

3.6.2 Extended Digraphs

Recall the notion of an extension of a digraph. The $(s_1, s_2, \ldots, s_n)$-**extension** (or just **extension**) $D(s_1, s_2, \ldots, s_n)$ of a digraph D with vertices labelled, say, $1, 2, \ldots, n$ is obtained from D by replacing every vertex i by a set of s_i independent (i.e., with no arc between them) vertices; more formally,

$$V(D(s_1, s_2, \ldots, s_n)) = \{(p_i, i) : \ 1 \leq p_i \leq s_i, \ i \in [n]\}$$

and $(p, i) \rightarrow (q, j)$ in $D(s_1, s_2, \ldots, s_n)$ if and only if $i \rightarrow j$ in D.

Observe that complete p-partite graph is an extension of K_p. The first result on the topic of this subsection was obtained by Šoltés [827].

Theorem 3.6.4 *If $n_1 \geq n_2 \geq 2$, then $\rho(K_{n_1,n_2}) = 1$ for $n_1 \leq \binom{n_2}{\lfloor n_2/2 \rfloor}$, and $\rho(K_{n_1,n_2}) = 2$, otherwise.* □

The original proof of Theorem 3.6.4 is rather long. A shorter proof of this result using the well-known Sperner's lemma is given by Gutin [450]; Gutin's proof is given in Chapter 2 of [91].

The exact values of $\rho(K_{n_1,n_2,\ldots,n_k})$ are unknown, but the following result obtained independently by Plesník [751] and Gutin [450] gives a sharp upper bound on $f(n_1, \ldots, n_k) = \mathrm{diam}_{\min}(K_{n_1,n_2,\ldots,n_k})$.

Theorem 3.6.5 *For every $k \geq 3$ and all positive integers $n_1, \ldots, n_k$, we have $2 \leq f(n_1, \ldots, n_k) \leq 3$.*

Proof: Obviously, $f(n_1, \ldots, n_k) \geq 2$.

If k is odd, let $R(n_1, n_2, \ldots, n_k)$ stand for a multipartite tournament with partite sets $V_1, \ldots, V_k$ of cardinalities $n_1, \ldots, n_k$ such that $V_i \rightarrow V_j$ if and only if $j - i \equiv 1, 2, \ldots, \lfloor k/2 \rfloor \pmod{k}$. If k is even, then $R(n_1, n_2, \ldots, n_k)$ is determined as follows: $R(n_1, n_2, \ldots, n_k) - V_k \cong R(n_1, n_2, \ldots, n_{k-1})$, $V_k \rightarrow V_i$ $(i = 1, 3, 5, \ldots, k-1)$, $V_j \rightarrow V_k$ $(j = 2, 4, 6, \ldots, k-2)$. We show that $\operatorname{diam}(R(n_1, n_2, \ldots, n_k)) \leq 3$ for every $k \geq 3$.

Case 1: k is odd, $k \geq 3$. It is sufficient to prove that $\operatorname{dist}(V_1, V_i) \leq 3$ for all $i \in [k]$. If $1 < j \leq \lfloor k/2 \rfloor + 1$, then $V_1 \rightarrow V_j$ by the definition. If $\lfloor \frac{k}{2} \rfloor + 1 < j \leq k$, then $V_{\lfloor k/2 \rfloor + 1} \rightarrow V_j$, hence $\operatorname{dist}(V_1, V_j) = 2$. Since $V_1 \rightarrow V_{\lfloor k/2 \rfloor + 1} \rightarrow V_{\lfloor k/2 \rfloor + 2} \rightarrow V_1$, we have $\operatorname{dist}(V_1, V_1) \leq 3$.

Case 2: k is even, $k \geq 4$. Since $R(n_1, \ldots, n_k) - V_k \cong R(n_2, \ldots, n_{k-1})$, we have $\operatorname{dist}(V_i, V_j) \leq 3$ for all $1 \leq i,\ j \leq k-1$. Moreover, $V_k \rightarrow V_i \rightarrow V_{i+1}$ for $i = 1, 3, 5, \ldots, k-3$ and $V_k \rightarrow V_{k-1}$. Therefore $\operatorname{dist}(V_k, V_t) \leq 2$ for $t \in [k-1]$. Analogously, $V_i \rightarrow V_{i+1} \rightarrow V_k$ for $i = 1, 3, 5, \ldots, k-3$ and $V_{k-1} \rightarrow V_1 \rightarrow V_2 \rightarrow V_k$. Hence $\operatorname{dist}(V_t, V_k) \leq 3$ for $t \in [k-1]$. Finally, $V_k \rightarrow V_1 \rightarrow V_2 \rightarrow V_k$. Therefore $\operatorname{dist}(V_k, V_k) \leq 3$. □

The following main result of this subsection was obtained by Gutin, Koh, Tay and Yeo [467].

Theorem 3.6.6 *Let H be a strong digraph of order $n \geq 3$ and let $D = H(s_1, s_2, \ldots, s_n)$ with $s_i \geq 2,\ 1 \leq i \leq n$. Then* $\operatorname{diam}(H) \leq \operatorname{diam}_{\min}(D) \leq \operatorname{diam}(H) + 2$.

Notice that Theorem 3.6.6 is a generalization of an analogous result for extensions of graphs obtained by Koh and Tay [615, 846].

The requirement $n \geq 3$ is important as one can see from Theorem 3.6.4 that $\operatorname{diam}(K_2) = 1$, but $\operatorname{diam}_{\min}(K_{s,2}) = 4$ for $s \geq 3$. Clearly, $\operatorname{diam}(H) \leq \operatorname{diam}(D')$ for every orientation D' of D. To prove the more difficult part of the inequality in Theorem 3.6.6, we will use the following lemma.

Lemma 3.6.7 *Let t_i, s_i be integers such that $2 \leq t_i \leq s_i$ for $1 \leq i \leq n$ and let H be a strong digraph with vertex set $[n]$, $n \geq 3$. If the digraph $D' = H(t_1, t_2, \ldots, t_n)$ admits an orientation F' in which every vertex $v = (p, i)$, such that i belongs to a cycle in H of length two, lies on a cycle C_v of length not exceeding m, then $D = H(s_1, s_2, \ldots, s_n)$ has an orientation F with diameter at most* $\max\{m, \operatorname{diam}(F')\}$.

Proof: Given an orientation F' of D', we define an orientation F of D as follows. We have $(p, i) \rightarrow (q, j)$ in F if and only if one of the following holds:

(a) $p < t_i$, $q < t_j$ and $(p, i) \rightarrow (q, j)$ in F'.
(b) $p < t_i$, $q \geq t_j$ and $(p, i) \rightarrow (t_j, j)$ in F'.
(c) $p \geq t_i$, $q < t_j$ and $(t_i, i) \rightarrow (q, j)$ in F'.
(d) $p \geq t_i$ and $q \geq t_j$ and $(t_i, i) \rightarrow (t_j, j)$ in F'.

Let $u = (p, i)$ and $v = (q, j)$ be a pair of distinct vertices in F. If $i \neq j$, then it is clear that $\text{dist}_F(u, v) \leq \text{diam}(F')$ (we can use obvious modifications of the corresponding paths in F'). We have the same result if $i = j$ but $p < t_i$ or $q < t_j$. Assume that $i = j$, $p \geq t_i$ and $q \geq t_j$. If i belongs to a cycle in H of length two, then using the cycle C_u we conclude that $\text{dist}_F(u, v) \leq m$. If i belongs to no cycle in H of length two, then since u, v dominate and are dominated by the same vertices and since $\text{dist}_F((1, i), (2, i)) \leq \text{diam}(F)$, we have $\text{dist}((p, i), (q, i)) \leq \text{diam}(F)$. □

Proof of Theorem 3.6.6: We prove that there exists an orientation D' of D such that $\text{diam}(D') \leq \text{diam}(H) + 2$. If $\text{diam}(H) = 1$, then this claim follows from Theorem 3.6.5. Thus, we may assume that $\text{diam}(H) \geq 2$.

Define an orientation F' of $H(t_1, t_2, \ldots, t_n)$, where every $t_i = 2$, as follows:

$$(1, i) \rightarrow (1, j) \rightarrow (2, i) \rightarrow (2, j) \rightarrow (1, i) \text{ if and only if } i < j. \tag{3.10}$$

Let $u = (p, i)$ and $v = (q, j)$ be a pair of distinct vertices in F'. We show that $\text{dist}_{F'}(u, v) \leq \text{diam}(H) + 2$. Suppose that $ik_1k_2 \ldots k_sj$ is a path of length $s + 1 = \text{dist}_H(i, j)$ in H. Then the path $Q = (p, i)(k_1^*, k_1)(k_2^*, k_2) \ldots (k_s^*, k_s)(j^*, j)$, where $x^* = 1$ or 2, is of length $\text{dist}_H(i, j)$ in F'. If $j^* = q$, then the last inequality follows. Otherwise, i.e., $j^* \neq q$, the path $Q(3 - k_s^*, k_s)(q, j)$ is of length $\text{dist}_H(i, j) + 2$ in F'. Thus, $\text{dist}_{F'}(u, v) \leq \text{diam}(H) + 2$. Hence, $\text{diam}(F') \leq \text{diam}(H)$. By (3.10), every vertex (p, i) of F', such that i lies on a cycle in H of length 2, belongs to a cycle of length 4. Now this theorem follows from Lemma 3.6.7. □

We finish this subsection by the following conjecture from [467]. It is unknown whether the conjecture is valid even for undirected graphs [615]. The conjecture is correct for $H = \overleftrightarrow{T}$, where T is a tree [846], and some other classes of digraphs, see [467].

Conjecture 3.6.8 *Let H be a strong digraph of order $n \geq 3$ and let $D = H(s_1, s_2, \ldots, s_n)$ with $s_i \geq 2$, $i \in [n]$, be of diameter at least three. Then* $\text{diam}_{\min}(D) \leq \text{diam}(H) + 1$.

3.6.3 Cartesian Products of Graphs

The **Cartesian product** of a family of undirected graphs $G_1, G_2, \ldots, G_n$, denoted by $G = G_1 \times G_2 \times \ldots \times G_n$ or $\prod_{i=1}^{n} G_i$, where $n \geq 2$, is the graph G having $V(G) = V(G_1) \times V(G_2) \times \ldots \times V(G_n) = \{(w_1, w_2, \ldots, w_n) : w_i \in V(G_i), i \in [n]\}$ and a pair of vertices $(u_1, u_2, \ldots, u_n)$ and $(v_1, v_2, \ldots, v_n)$ of G are adjacent if and only if there exists an $r \in [n]$ such that $u_rv_r \in E(G_r)$ and $u_i = v_i$ for all $i \in [n] \setminus \{r\}$. Let P_n (C_n, K_n) be the (undirected) path (cycle, complete graph) of order n and let T_n stand for a tree of order n. Roberts and Xu [781, 782, 783, 784] and Koh and Tan [606] evaluated the quantity

$\rho(P_k \times P_s)$. (We remark that Roberts and Xu [781, 782, 783, 784] considered objective functions other than ρ for orientations of the Cartesian products of undirected paths.) Koh and Tay [611] proved that most of those results can be extended as follows.

Theorem 3.6.9 *For $n \geq 2$, $k_1 \geq 3$, $k_2 \geq 6$ and $(k_1, k_2) \neq (3, 6)$, we have $\rho(P_{k_1} \times P_{k_2} \times \cdots \times P_{k_n}) = 0$.* □

This, in particular, generalizes the main result of McCanna [688] on n-cubes, i.e., the graphs $\prod_{i=1}^{n} P_2$. Koh and Tay [610] have obtained the values of $q(r, k) = \rho(C_{2r} \times P_k)$ for $r, k \geq 2$: $q(r, k) = 0$ if $k \geq 4$, $q(r, k) = 2$ if $k = 2$ and r is even, $q(r, k) = 1$ in the remaining cases.

They have also evaluated $\rho(K_m \times P_k)$, $\rho(K_m \times C_{2r+1})$ and $\rho(K_m \times K_n)$ [612], $\rho(K_m \times C_{2r})$ [614] and $\rho(T_m \times T_n)$ [616]. König, Krumme and Lazard [621] studied the Cartesian products of cycles. They proved the following interesting result.

Theorem 3.6.10 *Let p, q be integers with $p, q \geq 6$. If at least one of these two integers is even, then $\rho(C_p \times C_q) = 0$. If both p and q are odd, then $\rho(C_p \times C_q) = 1$.* □

König, Krumme and Lazard [621] evaluated $\rho(C_p \times C_q)$ in most cases when the minimum of p and q is smaller than 6. They also extended the $\rho(C_p \times C_q) = 0$ part of Theorem 3.6.10 to the Cartesian products of three or more cycles. These results are described in more detail in [846]. Some of the above results were extended by Koh and Tay [611], where the following theorem was proved.

Theorem 3.6.11 *For $m \geq 2$, $r \geq 0$, $k_1 \geq 3$, $k_2 \geq 6$ and $(k_1, k_2) \neq (3, 6)$, we have $\rho(\prod_{i=1}^{m} P_{k_i} \times \prod_{i=1}^{r} C_{n_i}) = 0$.* □

This result was further extended by Koh and Tay in [613]. For details, see [613] or Chapter 2 in [91].

3.6.4 Chordal Graphs

An undirected graph G is **chordal** if each cycle C of G of length at least 4 has a chord, i.e., an edge connecting two vertices of C that are not neighbours in C. Fomin, Matamala and Rapaport [327] proved the following:

Theorem 3.6.12 *Every connected chordal graph G with no bridge has an orientation of diameter at most $2\,\mathrm{diam}(G) + 1$.* □

Better bounds for $\mathrm{diam}_{\min}(G)$ can be obtained for special families of chordal graphs. An undirected graph G with $V(G) = \{v_i : i \in [n]\}$ is called an **interval graph** if there is a set $\{J_i : i \in [n]\}$ of intervals on the real line

such that v_iv_j is an edge if and only if $J_i \cap J_j \neq \emptyset$. If the corresponding set $\{J_i : i \in [n]\}$ of intervals can be chosen such that no interval is contained in another, then G is called a **proper interval graph**. The class of interval graphs is of great importance for graph theory and its applications [421]. It is easy to see that every interval graph is chordal.

Improving results of Fomin, Matamala, Prisner and Rapaport [326], Huang and Ye [542] proved the following:

Theorem 3.6.13 *We have* $\mathrm{diam}_{\min}(G) \leq \lceil \frac{3}{2}\mathrm{diam}(G) \rceil + 1$ *for each bridgeless connected interval graph* G. *If* G *is a 2-connected proper interval graph, then* $\mathrm{diam}_{\min}(G) \leq \lceil \frac{5}{4}\mathrm{diam}(G) \rceil + k$, *where* $k = 0$ *if* $\mathrm{diam}(G) \leq 3$ *and* $k = 1$, *otherwise.* □

It follows from examples in [326, 542] that the bounds of Theorem 3.6.13 are sharp.

3.7 Kings in Digraphs

In this section, we study r-kings in tournaments, semicomplete multipartite digraphs and other generalizations of tournaments. The main emphasis is on 4-kings in semicomplete multipartite digraphs. The notion of a 2-king and some results on 2-kings in tournaments will be generalized in Section 3.8.2.

3.7.1 2-Kings in Tournaments

Studying dominance in certain animal societies, the mathematical sociologist Landau [634] observed that every tournament has a 2-king. In fact, in every tournament T, each vertex x of maximum out-degree is a 2-king. Indeed, for a vertex $y \in T$, $y \neq x$, either $x \rightarrow y$ or there is an out-neighbour of x which is an in-neighbour of y. In both cases, $\mathrm{dist}(x, y) \leq 2$. Observe that if a tournament T has a vertex of in-degree zero, this vertex is the only r-king in T for every positive integer r. Moon [702] proved the following.

Theorem 3.7.1 *Every tournament with no vertex of in-degree zero has at least three 2-kings.*

Proof: Exercise 3.30. □

The following example shows that this bound on the number of 2-kings by Moon is sharp. Let T_n be a tournament with vertex set $\{x_1, x_2, \ldots, x_n\}$ and arc set $A = X \cup Y \cup \{x_{n-2}x_n\}$, where $X = \{x_ix_{i+1} : i \in [n-1]\}$ and $Y = \{x_jx_i : 1 \leq i < j-1 \leq n-1, (j,i) \neq (n, n-2)\}$. It is easy to verify that, for $n \geq 5$, $x_{n-3}, x_{n-2}, x_{n-1}$ are the only 2-kings in T_n (Exercise 3.32), see Figure 3.6.

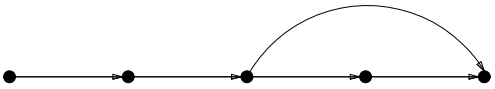


Figure 3.6 An example of a tournament with exactly three 2-kings. The arcs which are not shown are oriented from right to left.

Since the converse of a tournament is a tournament, the above two results can be reformulated for 2-serfs. (A vertex x is a **2 serf** if $\mathrm{dist}(V, x) \leq 2$.) The concepts of 2-kings and 2-serfs in tournaments were extensively investigated by both mathematicians and political scientists (the latter have studied so-called majority preferences). The interested reader is referred to Reid [774] for a comprehensive recent survey on the topic.

3.7.2 Kings in Semicomplete Multipartite Digraphs

It is easy to see that Proposition 3.1.1 implies that a multipartite tournament T has a finite out-radius if and only if T contains at most one vertex of in-degree zero (Exercise 3.34). Moreover, the following somewhat surprising assertion holds. If a multipartite tournament has finite out-radius, the out-radius is at most four. In other words, every multipartite tournament with at most one vertex of in-degree zero contains a 4-king. (Similar results hold for quasi-transitive digraphs and a certain class of digraphs that includes multipartite tournaments, see Subsection 3.7.3.) This result was proved independently by Gutin [446] and Petrović and Thomassen [749]. The bound is sharp as there exist infinitely many p-partite tournaments without 3-kings for every $p \geq 2$ [446]. Indeed, bipartite tournaments $\vec{C}_4[\overline{K}_q, \overline{K}_q, \overline{K}_q, \overline{K}_q]$ for $q \geq 2$ do not have 3-kings ($\mathrm{dist}(u, v) = 4$ for distinct vertices u, v from the same $\overline{K}_q$). It is clear that every multipartite tournament, for which the initial strong component is some $\vec{C}_4[\overline{K}_q, \overline{K}_q, \overline{K}_q, \overline{K}_q]$ ($q \geq 2$), has no 3-king either.

Thus, 4-kings are of particular interest in multipartite tournaments. In a number of papers (see, e.g., Gutin [450], Koh and Tan [607, 608, 609], Petrović [748] and the survey paper [774] by Reid) the authors investigate the minimum number of 4-kings in multipartite tournaments without vertices of in-degree zero. (If a multipartite tournament has exactly one vertex of in-degree zero, it contains exactly one 4-king, so this case is trivial.) In our view, the most interesting result in this direction was obtained by Koh and Tan in [607].

Theorem 3.7.2 *Let T be a k-partite tournament with no vertex of in-degree zero. If $k = 2$, T contains at least four 4-kings; it has exactly four 4-kings if its initial strong component consists of a cycle of length four. If $k \geq 3$, T contains at least three 4-kings; it has exactly three 4-kings if its initial strong component consists of a cycle of length three.* □

This theorem can be considered as a characterization of bipartite (p-partite, $p \geq 3$) tournaments with exactly k 4-kings for $k \in \{1,2,3,4\}$ ($k \in \{1,2,3\}$). The next theorem by Gutin and Yeo [474] goes further with respect to both exact number of 4-kings and the class of digraphs under consideration.

Theorem 3.7.3 *Let $D = (V, A)$ be a semicomplete multipartite digraph and let k be the number of 4-kings in D. Then*

1. *$k = 1$ if and only if D has exactly one vertex of in-degree zero.*
2. *$k = 2, 3$ or 4 if and only if the initial strong component of D has k vertices.*
3. *$k = 5$ if and only if either the initial strong component Q of D has five vertices or Q contains at least six vertices and possesses a path $P = p_0p_1p_2p_3p_4$ such that $\mathrm{dist}(p_0, p_4) = 4$ and $\{p_1, p_2, p_3, p_4\} \Rightarrow V - V(P)$.* □

We have seen that a vertex of maximum out-degree in a tournament is a 2-king. It is slightly more difficult to show that a vertex of maximum out-degree in a bipartite tournament is a 4-king (Exercise 3.33). With 4-kings in k-partite tournaments for $k \geq 3$, the situation is more complicated as can be seen from the next theorem by Goddard, Kubicki, Oellermann and Tian [412].

Theorem 3.7.4 *Let T be a strongly connected 3-partite tournament of order $n \geq 8$. If v is a vertex of maximum out-degree in T, then $\mathrm{dist}(v, V(T)) \leq \lfloor n/2 \rfloor$ and this bound is best possible.* □

In the rest of this subsection, we will prove the following theorem using an argument adapted from [474].

Theorem 3.7.5 *Every semicomplete multipartite digraph with at most one vertex of in-degree zero has a 4-king.*

For the proof we need the following lemmas:

Lemma 3.7.6 *If $P = p_0p_1 \ldots p_\ell$ is a shortest path from p_0 to p_ℓ in a semicomplete multipartite digraph D, and $\ell \geq 3$, then there is a (p_ℓ, p_0)-path of length at most 4 in $D\langle V(P)\rangle$.*

Proof: Since $\ell \geq 3$ and P is a shortest path we have $(\{p_0, p_1\}, p_\ell) = \emptyset$. If $p_\ell \rightarrow p_0$, we are done, so assume that p_ℓ and p_0 belong to the same partite set of D. This implies that $p_\ell \rightarrow p_1$. Analogously, $(p_0, \{p_2, p_3\}) = \emptyset$, which implies that either $p_\ell p_1 p_2 p_3 p_0$ or $p_\ell p_1 p_2 p_0$ is a (p_ℓ, p_0)-path of length at most 4 in $D\langle V(P)\rangle$. □

Lemma 3.7.7 *Let D be a semicomplete multipartite digraph and let Q be an initial strong component of D. If Q has at least two vertices, then D has only one initial strong component. Every vertex in Q, which is a 4-king in Q, is a 4-king in D.*

Proof: Assume that $|V(Q)| \geq 2$, but D has another initial strong component Q'. Since Q contains adjacent vertices, there is an arc between Q and Q', a contradiction.

Let x be a 4-king in Q and let $y \in V(D) - V(Q)$ be arbitrary. If x and y are adjacent, then clearly $x \rightarrow y$. Assume that x and y are not adjacent. Since Q is strong, it contains a vertex z dominated by x. Clearly, $x \rightarrow z \rightarrow y$. Hence $\text{dist}(x, y) \leq 2$ and x is a 4-king in D. □

Lemma 3.7.8 *Let D be a strong semicomplete multipartite digraph and let w be a vertex in D. For $i \geq 3$, if $N^{+i}(w) \neq \emptyset$, then* $\text{dist}(N^{+i}(w), N^{+i}[w]) \leq 4$.

Proof: Let $z \in N^{+i}(w)$ be arbitrary. Since a shortest path from w to z is of length $i \geq 3$, by Lemma 3.7.6, $\text{dist}(z, w) \leq 4$. Let $q \in N^{+i}[w] - \{w, z\}$ and let $r_0 r_1 \ldots r_j$ be a shortest (w, q)-path in D. If $1 \leq j \leq 3$, then, since z dominates at least one of the vertices r_0, r_1, either $z r_0 r_1 \ldots r_j$ or $z r_1 \ldots r_j$ is a (z, q)-path in D of length at most 4. If $j \geq 4$, then, since z dominates at least one of the vertices r_{j-3}, r_{j-2}, either $z r_{j-3} r_{j-2} r_{j-1} r_j$ or $z r_{j-2} r_{j-1} r_j$ is a (z, q)-path in D of length at most 4. □

Proof of Theorem 3.7.5: Let D be a semicomplete multipartite digraph with at most one vertex of in-degree zero. If D has a vertex x of in-degree zero, then clearly x is a 2-king in D. Thus, assume that D has no vertex of in-degree zero. Then, every initial strong component Q of D has at least two vertices. By Lemma 3.7.7, Q is unique and every 4-king in Q is a 4-king in D. It remains to show that Q has a 4-king. If every vertex in Q is a 4-king, then we are done. Otherwise, let w be a vertex in Q which is not a 4-king of Q. Then, $r = \text{dist}_Q(w, V(Q)) \geq 5$. By Lemma 3.7.8, $\text{dist}_Q(N_Q^{+r}(w), N_Q^{+r}[w]) \leq 4$, i.e., every vertex in $N_Q^{+r}(w)$ is a 4-king in Q (since $N_Q^{+r}[w] = V(Q)$). □

3.7.3 Kings in Generalizations of Tournaments

Bang-Jensen and Huang [104] considered kings in quasi-transitive digraphs. The main result of [104] is the following.

Theorem 3.7.9 *Let D be a quasi-transitive digraph. Then we have*

(1) D has a 3-king if and only if it has a finite out-radius[4].
(2) If D has a 3-king, then the following holds:

(a) Every vertex in D of maximum out-degree is a 3-king.
(b) If D has no vertex of in-degree zero, then D has at least two 3-kings.
(c) If the unique initial strong component of D contains at least three vertices, then D has at least three 3-kings. □

[4] See Proposition 3.1.1.

In the following family of quasi-transitive digraphs, every digraph has a 3-king but no 2-king: $\vec{C}_3[\overline{K}_{k_1}, \overline{K}_{k_2}, \overline{K}_{k_3}]$ for every $k_1, k_2, k_3 \geq 2$.

In [749], Petrović and Thomassen obtained the following.

Theorem 3.7.10 *Let G be an undirected graph whose complement is the disjoint union of complete graphs, paths and cycles. Then every orientation of G with at most one vertex of in-degree zero has a 6-king.* □

3.8 (k,l)-Kernels

Galeana-Sánchez and Li [382] introduced the concept of a (k,l)-kernel in a digraph. This concept generalizes several well-known notions of special independent sets of vertices such as a kernel and a quasi-kernel. In this section, we discuss (k,l)-kernels and their special important cases, kernels and quasi-kernels, and study some basic properties of kernels and quasi-kernels. The notion of a (k,l)-kernel has various applications, especially that of a $(2,1)$-kernel.

Let k and l be integers with $k \geq 2$, $l \geq 1$, and let $D = (V, A)$ be a digraph. A set $J \subseteq V$ is a **(k,l)-kernel** of D if

(a) for every ordered pair x, y of distinct vertices in J we have $\text{dist}(x,y) \geq k$,
(b) for each $z \in V - J$, there exists $x \in J$ such that $\text{dist}(z,x) \leq l$.

A **kernel** is a $(2,1)$-kernel and a **quasi-kernel** is a $(2,2)$-kernel. Galeana-Sánchez and Li [382] proved some results which relate (k,l)-kernels in a digraph D to those in its line digraph. In particular, they proved the following:

Theorem 3.8.1 *Let D be a digraph with $\delta^-(D) \geq 1$. Then the number of $(k,1)$-kernels in $L(D)$ is less than or equal to the number of $(k,1)$-kernels in D.* □

3.8.1 Kernels

We start with an equivalent definition of a kernel. A set K of vertices in a digraph $D = (V, A)$ is a kernel if K is independent and the first closed neighbourhood of K, $N^-[K]$, is equal to V. This notion was introduced by von Neumann in [723]; kernels have found many applications, for instance in game theory (a kernel represents a set of winning positions, cf. [723] and Chapter 14 in the book by Berge [144]), in logic [146] and in list edge-colouring of graphs (see Section 17.9). Chvátal (see [393], p. 204) proved that the problem to verify whether a given digraph has a kernel is $\mathcal{NP}$-complete. Several sufficient conditions for the existence of a kernel have been proved. Many of these conditions can be trivially extended to **kernel-perfect** digraphs, i.e., digraphs for which every induced subdigraph has a kernel. The notion

of kernel-perfect digraphs allows one to simplify certain proofs (due to the possibility of using induction, see the proof of Theorem 3.8.2) and is quite useful for applications (see Section 17.9).

Clearly, every symmetric digraph, i.e. digraph whose every arc belongs to a 2-cycle, is kernel-perfect (every maximal independent set is a kernel). It was proved by von Neumann and Morgenstern [723] that every acyclic digraph is kernel-perfect. Richardson [778] generalized this result as follows:

Theorem 3.8.2 *Every digraph with no odd cycle is kernel-perfect.*

The proof of Theorem 3.8.2, which we present here, is an adaptation of the one by Berge and Duchet [145]. A digraph which is not kernel-perfect is called **kernel-imperfect**. We say that a digraph D is **critical kernel-imperfect** if D is kernel-imperfect, but every proper induced subdigraph of D is kernel-perfect.

Lemma 3.8.3 *Every critical kernel-imperfect digraph is strong.*

Proof: Assume the converse and let $D = (V, A)$ be a non-strong critical kernel-imperfect digraph. Let T be a terminal strong component of D and let S_1 be a kernel of T. Since D has no kernel, the set $M = V - N^-[S_1]$ is non-empty. Hence the fact that D is critical kernel-imperfect implies that $D\langle M\rangle$ has a kernel S_2. The set $S_1 \cup S_2$ is independent since no arc goes from S_1 to S_2 (by the definition of a terminal strong component) and no arc goes from S_2 to S_1 (by the definition of M). Clearly, $N^-[S_1 \cup S_2] = V$. Hence, $S_1 \cup S_2$ is a kernel of D, a contradiction. □

Proof of Theorem 3.8.2: Let D be a kernel-imperfect digraph with no odd cycle and let D' be a critical kernel-imperfect subdigraph of D. By the lemma above, D' is strong. Since D' is strong and has no odd cycles, by Theorem 2.2.1, D' is bipartite. Let K be a partite set in D'. Since D' is strong, K is a kernel of D', a contradiction. □

This theorem has been strengthened in a number of papers. The conditions (a) and (b) of the following theorem are due to Duchet (see the papers by Berge [145]) and Galeana-Sánchez and Neumann-Lara [383], respectively). Galeana-Sánchez showed that for every $k \geq 2$, there are non-kernel-perfect digraphs for which every odd cycle has at least k chords [380].

Theorem 3.8.4 *A digraph D is kernel-perfect if at least one of the following conditions holds:*

(a) Every odd cycle has two arcs belonging to 2-cycles;
(b) Every odd cycle has two chords whose heads are consecutive vertices of the cycle. □

There were other attempts to strengthen Richardson's Theorem 3.8.2. In particular, Duchet (see [176]) conjectured that every digraph D, which is not an odd cycle and which does not have a kernel, contains an arc e such that $D-e$ has no kernel either. Apartsin, Ferapontova and Gurvich [42] found a counterexample to this conjecture. They proved that the circulant digraph[5] $C_{43}(1,7,8)$ has no kernel, but after deletion of any arc in this digraph a kernel will appear.

Observe that by the symmetry of $C_{43}(1,7,8)$ one needs only to show that $C_{43}(1,7,8)-(1,2)$, $C_{43}(1,7,8)-(1,8)$ and $C_{43}(1,7,8)-(1,9)$ have kernels. This task is left as Exercise 3.37. We note that $C_{43}(1,7,8)$ is the only known counterexample to the Duchet conjecture; Gurvich (private communication, December 1999) suspects that there is an infinite family of such circulant digraphs. It was also proved in [42] that $C_n(1,7,8)$ has a kernel if and only if $n \equiv 0 \pmod 3$ or $n \equiv 0 \pmod{29}$. The following problem seems quite natural:

Problem 3.8.5 *Characterize circulant digraphs which have kernels.*

A biorientation D of a graph G is called **normal**, if every subdigraph of D which is a semicomplete digraph has a kernel. An undirected graph G is **kernel-solvable** if every normal biorientation of G has a kernel. Boros and Gurvich [176] showed that a slight modification of the above conjecture of Duchet holds. They proved the following:

Theorem 3.8.6 *Let G be a connected non-kernel-solvable graph, which is not an odd cycle of length at least 5. Then there exists an edge e in G such that $G-e$ is not kernel-solvable either.* □

Berge and Duchet (see [678]) conjectured that a graph G is perfect[6] if and only if G is kernel-solvable. Boros and Gurvich [175] proved one direction of this conjecture, namely:

Theorem 3.8.7 *Every perfect graph is kernel-solvable.* □

The two original proofs of Theorem 3.8.7 are quite involved and lengthy. Using the notion of a fractional kernel, Aharoni and Holzman [8] found a much shorter proof of Theorem 3.8.7. The fact that every kernel-solvable graph is perfect follows [177] from the following important result, the Strong Perfect Graph Theorem proved by Chudnovsky, Robertson, Seymour and Thomas [216]. An induced cycle of odd length at least 5 is called an **odd hole**. An induced subgraph that is the complement of an odd hole is called an **odd anti-hole**.

[5] Circulant digraphs are introduced in Section 2.14.1.

[6] A graph G is perfect if, for every induced subgraph H of G, the chromatic number of H is equal to the order of the largest clique of H.

Theorem 3.8.8 *A graph G is perfect if and only if G has no odd hole and odd anti-hole.* □

3.8.2 Quasi-Kernels

We start with an equivalent definition of a quasi-kernel. A set Q of vertices in a digraph $D = (V, A)$ is a **quasi-kernel** if Q is independent and the second closed in-neighbourhood of Q, $N^{-2}[Q]$, is equal to V. The two results on 2-kings (or, more precisely, 2-serfs) in tournaments mentioned in the beginning of Section 3.7 have been extended to quasi-kernels in arbitrary digraphs as follows. The first theorem is by Chvátal and Lovász [222]. We give a very short proof by S. Thomassé (see [166]).

Theorem 3.8.9 *Every digraph D has a quasi-kernel.*

Proof: Let $V = V(D)$. Consider an ordering $x_1, \ldots, x_n$ of V and two spanning subdigraphs of D, $D_1 = (V, A_1)$ and $D_2 = (V, A_2)$, where $A_1 = \{x_i x_j : x_i x_j \in A(D), i < j\}$ and $A_2 = \{x_i x_j : x_i x_j \in A(D), j < i\}$. By Theorem 3.8.2, D_1 has a kernel K' and $D_2[K']$ has a kernel K''. Observe that K'' is a quasi-kernel of D. □

The second theorem is by Jacob and Meyniel [559].

Theorem 3.8.10 *If a digraph $D = (V, A)$ has no kernel, then D contains at least three quasi-kernels.*

Proof: By Theorem 3.8.9, D has a quasi-kernel Q_1. Since D has no kernel, we have $V \neq N^-[Q_1]$. Let Q_2 be a quasi-kernel of $D - N^-[Q_1]$. We will prove that $Q_2' = Q_2 \cup (Q_1 - N^-(Q_2))$ is a quasi-kernel of D. It is straightforward to see that Q_2' is independent and

$$V = (V - N^-[Q_1]) \cup N^-[Q_1 \cap N^-(Q_2)] \cup N^-[Q_1 - N^-(Q_2)].$$

By the definition of Q_2, every vertex of $V - N^-[Q_1]$ is the initial vertex of a path of length at most two terminating in Q_2. Since $N^-[Q_1 \cap N^-(Q_2)] \subseteq N^{-2}[Q_2]$, every vertex of $N^-[Q_1 \cap N^-(Q_2)]$ is the initial vertex of a path of length at most two terminating in Q_2. Since $N^-[Q_1 - N^-(Q_2)] \subseteq N^-[Q_1]$, a vertex of $N^-[Q_1 - N^-(Q_2)]$ either belongs to Q_1 or is the tail of an arc whose head is in $Q_1 - N^-(Q_2)$. Hence, Q_2 is a quasi-kernel.

Observe that $Q_1 \cap Q_2 = \emptyset$ and $Q_2 \neq \emptyset$. Thus, $Q_2' \neq Q_1$.

As Q_2' is not a kernel of D, we have $V \neq N^-[Q_2']$. Let Q_3 be a quasi-kernel of $D - N^-[Q_2']$ and let $Q_3' = Q_3 \cup (Q_2' - N^-(Q_3))$. As above, we can demonstrate that Q_3' is a quasi-kernel distinct from Q_2'. It remains to show that $Q_3' \neq Q_1$. Observe that $Q_3 \subseteq V - N^-[Q_2']$ and $Q_1 \subseteq N^-[Q_2']$. Thus, $Q_1 \cap Q_3 = \emptyset$. By this fact and since Q_3 is non-empty, we conclude that $Q_3' \neq Q_1$. □

Gutin, Koh, Tay and Yeo [468] characterized digraphs with exactly one and two quasi-kernels, and, thus, provided necessary and sufficient conditions for a digraph to have at least three quasi-kernels. In particular, they proved the following:

Theorem 3.8.11 *Every strong digraph of order at least three, which is not a 4-cycle, has at least three quasi-kernels.* □

3.9 Exercises

3.1. Formulate the shortest (s,t)-path problem as a linear programming problem with integer variables. Hint: use a variable for each arc.

3.2. Show how to check whether an undirected graph is bipartite in linear time using BFS. Does your method extend to strongly connected digraphs? That is, can you check whether a strong digraph is bipartite using BFS? Hint: consider the proof of Theorem 2.2.1.

3.3. Illustrate the shortest path algorithm for acyclic digraphs (Subsection 3.3.2) on the acyclic digraph in Figure 3.7.

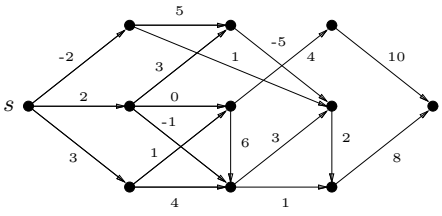


Figure 3.7 A weighted acyclic digraph.

3.4. **Finding the longest paths from a fixed vertex to all other vertices in a weighted acyclic digraph**. Develop a polynomial algorithm for finding the longest paths from a fixed vertex s to all other vertices in an arbitrary weighted acyclic digraph. Preferably your algorithm should run in linear time.

3.5. Find the longest paths from s to all other vertices in the acyclic digraph in Figure 3.7, e.g., using the algorithm that you designed in Exercise 3.4.

3.6. **Finding a longest path in a weighted acyclic digraph in linear time**. Show how to find a longest path in a weighted acyclic digraph D in linear time. Hint: use a variant of the dynamic programming approach taken in (3.3), or construct a superdigraph D' of D such that one can read out a longest path in D from a shortest path tree from some vertex s in D'.

3.7. Execute Dijkstra's algorithm on the digraph in Figure 3.8.

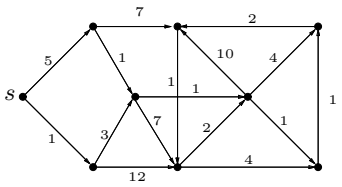


Figure 3.8 A digraph with non-negative weights on the arcs.

3.8. Complete the description of Dijkstra's algorithm in Subsection 3.3.3 such that not only the distances from s to the vertices of D are computed, but also the actual shortest paths are found.

3.9. Complete the description of the Bellman-Ford-Moore algorithm in Subsection 3.3.4 such that not only the distances from s to the vertices of D are computed, but also the actual shortest paths are found.

3.10. Execute the Bellman-Ford-Moore algorithm on the digraph in Figure 3.9. Perform the scanning of arcs in lexicographic order.

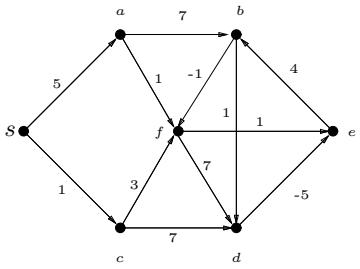


Figure 3.9 A digraph with weights on the arcs and no negative cycles.

3.11. **Negative cycle detection using the Bellman-Ford-Moore algorithm.** Prove Theorem 3.3.10.

3.12. Show how to detect a negative cycle in the digraph in Figure 3.10 using the extension of the Bellman-Ford-Moore algorithm.

3.13. Show by an example that Dijkstra's algorithm may not find the correct distances if it is applied to a weighted directed graph D where some arcs have negative weights, even if there is no negative cycle in D.

3.14. Show how to implement the Floyd-Warshall algorithm so that it runs in time $O(n^3)$.

3.15. Prove Theorem 3.3.10.

3.16. **Re-weighting the arcs of a digraph**. Let $D = (V, A, c)$ be a weighted digraph and let $\pi : V \to \mathcal{R}$ be a function on the vertices of D. Define a new weight function c^* by $c^*(u, v) = c(u, v) + \pi(u) - \pi(v)$ for all $v \in V$. Let dist*

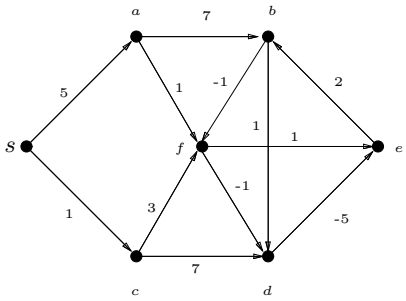


Figure 3.10 A weighted digraph with a negative cycle.

be the distance function with respect to $D^* = (V, A, c^*)$, and let P be an (x, y)-path in D. Prove that P is a shortest (x, y)-path in D (with respect to c) if and only if P is a shortest (x, y)-path in D^* (with respect to c^*). Hint: consider what happens to the length of a path after the transformation above.

3.17. Consider the weights introduced in Exercise 3.16. Show that the weight of a cycle in D is unchanged under the transformation from $D = (V, A, c)$ to $D^* = (V, A, c^*)$.

3.18. **Getting rid of negative weight arcs by re-weighting.** Let $D = (V, A, c)$ be a weighted digraph with some arcs of negative weight, but with no negative cycle. Let $D' = (V, A', c')$ be obtained from D by adding a new vertex s and all arcs of the form sv, $v \in V$, and setting $c'(s, v) = 0$ for all $v \in V$ and $c'(u, v) = c(u, v)$ for all $u, v \in V$. Let $\pi(v) = \mathrm{dist}_{D'}(s, v)$ for all $v \in V$. Define c^* by $c^*(u, v) = c(u, v) + \pi(u) - \pi(v)$ for all $u, v \in V$. Prove that $c^*(u, v) \geq 0$ for all $u, v \in V$.

3.19. **Johnson's algorithm for shortest paths.** Show that by combining the observations of Exercises 3.16-3.18, one can obtain an $O(n^2 \log n + nm)$ algorithm for the all pairs shortest path problem in digraphs with no negative cycles (Johnson [569]).

3.20. Let $M = [m_{ij}]$ be the adjacency matrix of a digraph $D = (V, A)$ with $V = [n]$ and let k be a natural number. Prove that there is an (i, j)-walk of length k in D if and only if the (i, j) entry of the kth power of M is positive.

3.21. Show how to compute the kth power of the adjacency matrix of a digraph of order n in time $O(P(n) \log k)$, where $P(n)$ is the time required to compute the product of two $n \times n$ matrices.

3.22. **Finding a shortest cycle in a digraph**. Describe a polynomial algorithm to find the shortest cycle in a digraph. Hint: use Exercise 3.20.

3.23. Prove Proposition 3.4.2.

3.24. **Short cycles through an edge**. Let $G = (V, E)$ be a 2-edge-connected graph and let $uv \in E$. Prove that G has a cycle of length at most $2\mathrm{dist}(u, V) + 1$ through the edge uv. Hint: use the (undirected) distance classes from u and v as well as the fact that uv is not a bridge.

3.25. We call a family $\mathcal{F}$ of subsets of $[n]$ an **antichain** if no set in $\mathcal{F}$ is contained in another. Prove Theorem 3.6.4 using the following lemma of Sperner. Let $\mathcal{F}$ be an antichain on $[n]$. Then $|\mathcal{F}| \leq \binom{n}{\lfloor n/2 \rfloor}$. The bound is attained by taking $\mathcal{F}$ to be the family of all subsets of size $\lfloor n/2 \rfloor$. (Gutin [450])

3.26. Prove that $\rho(C_p \times C_q) > 0$ when both p and q are odd $(p, q \geq 3)$ (D.B. West, see [621]).

3.27. Construct orientations of $P_3 \times P_6$ and $P_3 \times P_7$ of diameter 8.

3.28. For every odd number $n \geq 3$, give an example of a tournament T of order n, in which all vertices are 2-kings.

3.29. Let T be a tournament on 4 vertices. Show that T contains a vertex which is not a 2-king.

3.30. Prove Theorem 3.7.1 (Moon [704]).

3.31. Describe an infinite family of semicomplete digraphs, in which every member has exactly two 2-kings.

3.32. Prove that the tournament T_n in Subsection 3.7.1 has only three 2-kings for $n \geq 5$.

3.33. **4-kings in bipartite tournaments.** Prove that a vertex of maximum out-degree in a strong bipartite tournament is a 4-king. For all $s, t \geq 4$ construct strong bipartite tournaments with partite sets of cardinality s and t which do not have 3-kings. (Gutin [446])

3.34. Prove that a multipartite tournament T has a finite out-radius if and only if T contains at most one vertex of in-degree zero. Hint: use Proposition 3.1.1.

3.35. **3-kings in quasi-transitive digraphs**. Show that every quasi-transitive digraph of finite radius has a 3-king (Bang-Jensen and Huang [104]).

3.36. Give a direct proof that every acyclic digraph is kernel-perfect. Prove that an acyclic digraph has a unique kernel (von Neumann and Morgenstern [723]).

3.37. Prove that $C_{43}(\{1,7,8\}) - (1,2)$, $C_{43}(\{1,7,8\}) - (1,8)$ and $C_{43}(\{1,7,8\}) - (1,9)$ have kernels, where $C_{43}(\{1,7,8\})$ is a circulant digraph.

4. Flows in Networks

The purpose of this chapter is to describe basic elements of the theory and applications of network flows. This topic is probably the most important single tool for applications of digraphs and perhaps even of graphs as a whole. At the same time, from a theoretical point of view, flow problems constitute a beautiful common generalization of shortest path problems and problems such as finding internally (arc)-disjoint paths from a given vertex to another. The theory of flows is well understood and fairly simple. This, combined with the enormous applicability to real-life problems, makes flows a very attractive topic to study. From a theoretical point of view, flows are well understood as far as the basic questions, such as finding a maximum flow from a given source to a given sink or characterizing the size of such a flow, are concerned. However, the topic is still a very active research field and there are challenging open problems such as deciding whether an $O(nm)$ algorithm[1] exists for the general maximum flow problem.

Several books deal almost exclusively with flows; see, e.g., the books [13] by Ahuja, Magnanti and Orlin, [267] by Dolan and Aldous, the classical text [331] by Ford and Fulkerson and [710] by Murty. In particular, [13] and [710] contain a wealth of applications of flows. In this chapter we can only cover a very small part of the theory and applications of network flows, but we will try to illustrate the diversity of the topic and show several applications of a practical as well as theoretical nature. Many of the results given in this chapter will be used in several other chapters such as those on connectivity and hamiltonian cycles.

4.1 Definitions and Basic Properties

A **network** is a directed graph $D = (V, A)$ associated with the following functions on $V \times V$: a **lower bound** $l_{ij} \geq 0$, a **capacity** $u_{ij} \geq l_{ij}$ and a **cost** c_{ij} for each $(i, j) \in V \times V$. These parameters satisfy the following requirement:

[1] Here and everywhere in this chapter n is the number of vertices and m the number of arcs in the network under consideration.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_4,

$$\text{For every } (i,j) \in V \times V, \text{ if } ij \notin A, \text{ then } l_{ij} = u_{ij} = 0. \tag{4.1}$$

In order to simplify notation in this chapter we also make the assumption that

$$c_{ij} = -c_{ji} \qquad \forall (i,j) \in V \times V. \tag{4.2}$$

This assumption may seem restrictive but it is purely a technical convention to make some of the following definitions simpler (in particular, the definition of costs in the residual network in Subsection 4.1.2). When it comes to implementing algorithms for various flow problems involving costs, this assumption can easily be avoided (Exercise 4.2). Finally we assume that if there is no arc between i and j (in any direction, then $c_{ij} = 0$.

In some cases we also have a function $b : V \to \mathcal{R}$ called a **balance vector** which associates a real number with each vertex of D. We will always assume that

$$\sum_{v \in V} b(v) = 0. \tag{4.3}$$

We use the shorthand notation $\mathcal{N} = (V, A, l, u, b, c)$ to denote a network with corresponding digraph $D = (V, A)$ and parameters l, u, b, c. If there are no costs specified, or there is no prescribed balance vector, then we omit the relevant letters from the notation. Note that whenever we consider a network $\mathcal{N} = (V, A, l, u, b, c)$ we also have a digraph, namely, the digraph $D = (V, A)$ that we obtain from $\mathcal{N}$ by omitting all the functions l, u, b, c.

For a given pair of not necessarily disjoint subsets U, W of the vertex set of a network $\mathcal{N} = (V, A, l, u)$ and a function f on $V \times V$ we use the notation $f(U, W)$ as follows (here f_{ij} denotes the value of f on the pair (i, j)):

$$f(U,W) = \sum_{i \in U, j \in W} f_{ij}. \tag{4.4}$$

We will always make the realistic assumption that $n = O(m)$ which holds for all interesting networks. In fact, almost always, the networks on which we work will be connected as digraphs.

4.1.1 Flows and Their Balance Vectors

A **flow** in a network $\mathcal{N}$ is a function $x : A \to \mathcal{R}_0$ on the arc set of $\mathcal{N}$. We denote the value of x on the arc ij by x_{ij}. For convenience, we will sometimes think of x as a function of $V \times V$ and require that $x_{ij} = 0$ if $ij \notin A$ (see, e.g., the definition of residual capacity in (4.7)). An **integer flow** in $\mathcal{N}$ is a flow x such that $x_{ij} \in \mathcal{Z}_0$ for every arc ij. For a given flow x in $\mathcal{N}$ the **balance vector of** x is the following function b_x on the vertices:

$$b_x(v) = \sum_{vw \in A} x_{vw} - \sum_{uv \in A} x_{uv} \qquad \forall v \in V. \tag{4.5}$$

That is, $b_x(v)$ is the difference between the flow on arcs with tail v and the flow on arcs with head v. We classify vertices according to their balance values (with respect to x). A vertex v is a **source** if $b_x(v) > 0$, a **sink** if $b_x(v) < 0$ and otherwise v is **balanced** ($b_x(v) = 0$). When there is no confusion possible (in particular when there is only one flow in question) we may drop the index x on b and say that b is the balance vector of x.

A flow x in $\mathcal{N} = (V, A, l, u, b, c)$ is **feasible** if $l_{ij} \leq x_{ij} \leq u_{ij}$ for all $ij \in A$ and $b_x(v) = b(v)$ for all $v \in V$. If no balance vector is specified for the network, then a feasible flow x is only required to satisfy $l_{ij} \leq x_{ij} \leq u_{ij}$ for all $(i, j) \in A$.

The **cost** of a flow x in $\mathcal{N} = (V, A, l, u, c)$ is given by

$$c^T x = \sum_{ij \in A} c_{ij} x_{ij}. \tag{4.6}$$

See Figure 4.1 for an example of a feasible flow.

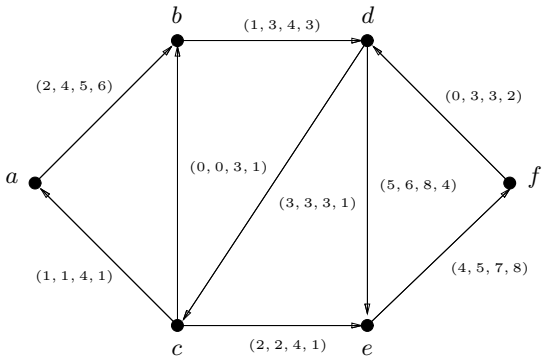


Figure 4.1 A network $\mathcal{N} = (V, A, l, u, c)$ with a feasible flow x specified. The specification on each arc ij is $(l_{ij}, x_{ij}, u_{ij}, c_{ij})$. The cost of the flow is 109.

We point out that whenever the lower bounds are all zero (an assumption that is not a restriction of the modelling power of flows as we shall see in Section 4.2) we will always assume that if iji is a 2-cycle of a network $\mathcal{N}$ and x is a flow in $\mathcal{N}$, then at least one of x_{ij}, x_{ji} is equal to zero. We call such a flow a **netto flow** in $\mathcal{N}$. The practical motivation for this restriction is that very often one uses flows to model items (water, electricity, telephone messages, etc.) that move from one place to another in time. Here it makes perfect sense to say that sending 3 units from i to j and 2 units from j to i is the same as sending 1 unit from i to j and nothing from j to i (we say

that 2 of the units **cancel out**). In some of the definitions below it is easier to work with netto flows.

The notion of flows generalizes that of paths in directed graphs. Indeed, if P is an (s,t)-path in a digraph $D=(V,A)$, then we can describe a feasible flow x in the network $\mathcal{N}=(V,A,l\equiv 0,u\equiv 1)$ by taking $x_{ij}=1$ if ij is an arc of P and $x_{ij}=0$ otherwise. This flow has balance vector

$$b_x(v)=\begin{cases}1 & \text{if } v=s\\ -1 & \text{if } v=t\\ 0 & \text{otherwise.}\end{cases}$$

We can also see that if there are weights on the arcs of D and we let $\mathcal{N}$ inherit these weights as costs on the arcs, then the cost of the flow defined above is equal to the length (weight) of P. Hence the shortest path problem is a special case of the minimum cost flow problem (which is studied in Section 4.10) with respect to the balance vector described above (here we implicitly used Theorem 4.3.1 for the other direction of going from a flow to an (s,t)-path in D). In a very similar way we can also see that flows generalize cycles in digraphs. It is an important and very useful fact about flows that in some sense one can also go the other way. As we shall see in Theorem 4.3.1, every flow in a network with n vertices and m arcs can be decomposed into no more than $n+m$ flows along simple paths and cycles. Furthermore, paths and cycles play a fundamental role in several algorithms for finding optimal flows where the optimality is with respect to measures we define later.

4.1.2 The Residual Network

The concept of a residual network was implicitly introduced by Ford and Fulkerson [331].

For a given flow x in a network $\mathcal{N}=(V,A,l,u,c)$, define the **residual capacity** r_{ij} from i to j as follows:

$$r_{ij}=(u_{ij}-x_{ij})+(x_{ji}-l_{ji}). \tag{4.7}$$

The **residual network** $\mathcal{N}(x)$ with respect to x is defined as $\mathcal{N}(x)=(V,A(x),\tilde{l}\equiv 0,r,c)$, where $A(x)=\{ij : r_{ij}>0\}$. That is, the cost function is the same[2] as for $\mathcal{N}$ and all lower bounds are zero. See Figure 4.2 for an illustration.

The arcs of the residual network have a natural interpretation. If $ij\in A$ and $x_{ij}=5<7=u_{ij}$, then we may increase x by up to 2 units on the arc ij at the cost of c_{ij} per unit. Furthermore, if we also have $l_{ij}=2$, then we can also choose to decrease x by up to 3 units along the arc ij. The cost of this decrease is exactly $c_{ji}=-c_{ij}$ per unit. Note that a decrease of flow along the

[2] Note that this differs from definitions in other texts such as [13], but we can do this since we made the assumption (4.2).

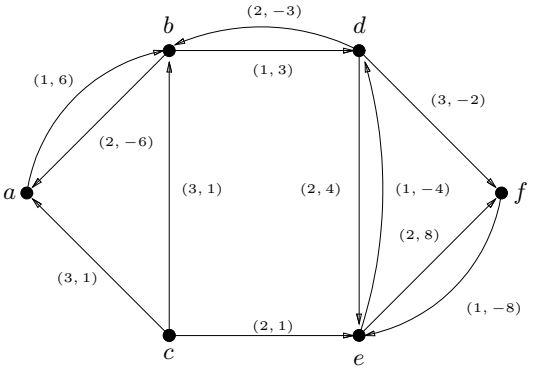


Figure 4.2 The residual network $\mathcal{N}(x)$ corresponding to the flow in Figure 4.1. The data on each arc is (r, c).

arc ij may also be thought of as sending flow in the opposite direction along the residual arc ji and then cancelling out.

4.2 Reductions Among Different Flow Models

The purpose of this section is to show that one can restrict the general definition of a flow network considerably and still retain its modeling generality. We also show that one can model networks with lower bounds, capacities and costs on the vertices by networks, where all these numbers are on arcs only.

4.2.1 Eliminating Lower Bounds

We start with the following easy observation which shows that within the general model the assumption that all lower bounds are zero does not limit the model.

Lemma 4.2.1 *Let $\mathcal{N} = (V, A, l, u, b, c)$ be a network.*

(a) Suppose that the arc $ij \in A$ has $l_{ij} > 0$. Let $\mathcal{N}'$ be obtained from $\mathcal{N}$ by making the following changes: $b(j) := b(j) + l_{ij}$, $b(i) := b(i) - l_{ij}$, $u_{ij} := u_{ij} - l_{ij}$, $l_{ij} := 0$. Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}'$ and vice versa. Furthermore, the costs of these two flows are related by $c^T x = c^T x' + l_{ij} c_{ij}$.

(b) There exists a network $\mathcal{N}_{l \equiv 0}$ in which all lower bounds are zero such that every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{l \equiv 0}$ and vice versa. Furthermore, the costs of these two flows are related by $c^T x = c^T x' + \sum_{ij \in A} l_{ij} c_{ij}$.

Proof: Part (a) is left to the reader as Exercise 4.3. Since we may eliminate lower bounds one arc at the time, (b) follows from (a) by induction on the number of arcs. □

It is also useful to observe that we can construct $\mathcal{N}'$ from $\mathcal{N}$ in time $O(n+m)$ and reconstruct the flow x from x' in time $O(m)$. Hence the time for eliminating lower bounds and reconstructing a flow in the original network is negligible since all algorithms on networks need $O(n+m)$ time just to input the network.

4.2.2 Flows with One Source and One Sink

Let s, t be distinct vertices of a network $\mathcal{N} = (V, A, l \equiv 0, u, c)$. An **$(s,t)$-flow** is a flow x satisfying the following for some $k \in \mathcal{R}_0$:

$$b_x(v) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise.} \end{cases}$$

The **value** of an (s,t)-flow x is denoted by $|x|$ and is defined by

$$|x| = b_x(s). \tag{4.8}$$

The next lemma combined with Lemma 4.2.1 shows that using only (s,t)-flows, one can model everything which can be modeled via flows in the general network model.

Lemma 4.2.2 *Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network. Let*[3] *$M = \sum_{\{v:b(v)>0\}} b(v)$ and let $\mathcal{N}_{st}$ be the network defined as follows: $\mathcal{N}_{st} = (V \cup \{s,t\}, A', l' \equiv 0, u', b', c')$, where*

(a) $A' = A \cup \{sr : b(r) > 0\} \cup \{rt : b(r) < 0\}$,
(b) $u'_{ij} = u_{ij}$ for all $ij \in A$, $u_{sr} = b(r)$ for all r such that $b(r) > 0$ and $u_{qt} = -b(q)$ for all q such that $b(q) < 0$,
(c) $c'_{ij} = c_{ij}$ for all $ij \in A$ and $c' = 0$ for all arcs leaving s or entering t,
(d) $b'(v) = 0$ for all $v \in V$, $b'(s) = M$, $b'(t) = -M$.

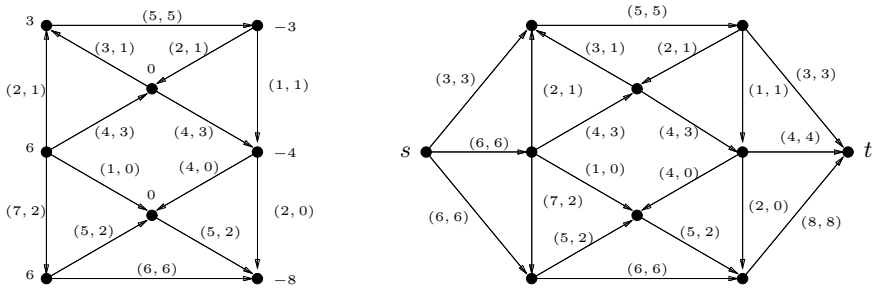
Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{st}$ and vice versa. Furthermore, the costs of x and x' are related by $c^T x = c'^T x'$. See Figure 4.3.

Proof: Exercise 4.4. □

It follows from Lemma 4.2.2 that given any network $\mathcal{N}$ in which all lower bounds are zero, we can check the existence of a feasible flow in $\mathcal{N}$ by constructing the corresponding network $\mathcal{N}_{st}$ and check whether this network has

[3] Recall that we also have $M = -\sum_{\{v:b(v)<0\}} b(v)$ by (4.3).

an (s,t)-flow x such that $|x| = M$ where M is defined in Lemma 4.2.2. This latter task is precisely the problem of finding the maximum value of a feasible (s,t)-flow in $\mathcal{N}_{st}$, a problem which we study extensively in Sections 4.5-4.7. See also Theorem 4.8.3.



(a) (b)

Figure 4.3 Part (a) shows a network $\mathcal{N}$ with a feasible flow with respect to the balance vector specified at each vertex. The numbers on each arc are (capacity, flow). Costs are omitted for clarity. Part (b) shows the network $\mathcal{N}_{st}$ as defined in Lemma 4.2.2 and a feasible flow x' in $\mathcal{N}_{st}$.

4.2.3 Circulations

A **circulation** is a flow x with $b_x(v) = 0$ for all $v \in V$. Combining our next result with Lemmas 4.2.1 and 4.2.2 shows that one can also model everything that can be modeled in the general (flow) network model by the seemingly much more restricted circulations. Note that we cannot completely exclude lower bounds in this reduction (see Exercise 4.5).

Lemma 4.2.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with distinct vertices s, t and let the balance vector of $\mathcal{N}$ satisfy $b(v) = 0$ for all $v \in V - \{s,t\}$, $b(s) = M$, $b(t) = -M$, for some $M \in \mathcal{R}_0$. Let $\mathcal{N}^* = (V, A \cup \{ts\}, l'', u'', c'')$ be the network obtained from $\mathcal{N}$ by adding a new arc ts with lower bound $l_{ts} = M$, capacity $u_{ts} = M$ and cost $c''_{ts} = 0$, keeping the lower bound, capacity and cost of each original arc and posing no restriction on the balance vector of $\mathcal{N}^*$. Then every feasible (s,t)-flow x in $\mathcal{N}$ corresponds to a feasible circulation x'' in $\mathcal{N}^*$ and vice versa. Furthermore, the costs of x and x'' are related by $c^T x = c''^T x''$.*

Proof: Exercise 4.5. □

The concept of a circulation is a very useful tool for applications to questions concerning sub(di)graphs of (di)graphs as we show in Section 4.11.

4.2.4 Networks with Bounds and Costs on the Vertices

In some applications of flows one is not interested in imposing lower bounds and capacities on arcs, but rather on vertices. One such example is when one is looking for a cycle subdigraph that contains all vertices of a certain subset X and possibly other vertices (see Section 4.11). Another example is when one is looking for a path factor which covers all vertices of a digraph (see Section 13.5). We show below how to model networks with lower bounds, capacities and costs on vertices (and possibly also on arcs) by standard networks where all functions, other than the balance vectors, are on the arcs. First we introduce a useful transformation of any digraph to a bipartite digraph which we will use not only for the problem above but also several other places in the book.

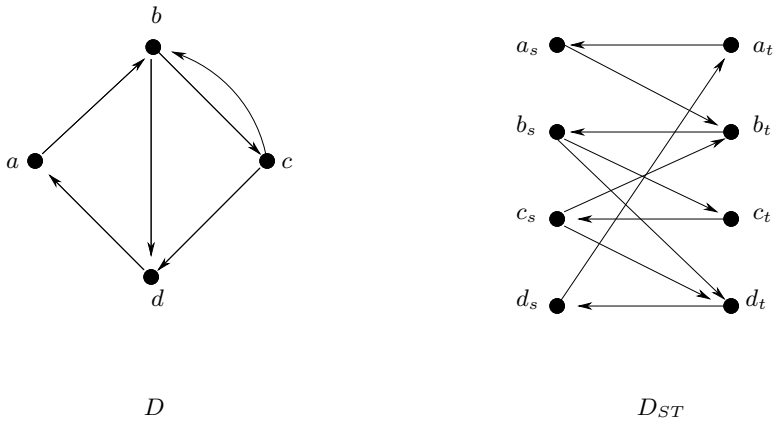


Figure 4.4 The vertex splitting procedure.

Given a digraph $D = (V, A)$, construct a new digraph D_{ST} as follows. For each vertex $v \in V$, D_{ST} contains two new vertices v_s, v_t and the arc $v_t v_s$. For each arc $xy \in A(D)$, $A(D_{ST})$ contains the arc $x_s y_t$. See Figure 4.4. We say that the digraph D_{ST} is obtained from D by the **vertex splitting procedure**.

Now suppose that $\mathcal{N} = (V, A, l, u, b, c, l^*, u^*, c^*)$ is a network with a prescribed balance vector b, lower bounds, capacities and costs l, u, c on the arcs (the case when there are no such specifications can easily be modeled by taking $l \equiv 0, u \equiv \infty, c \equiv 0$) and lower bounds, capacities and costs l^*, u^*, c^* on the vertices. To be precise we have to define the meaning of these new parameters. There is some freedom in such a definition, but for the applications we will need, it suffices to use the definition that $l^*(v)$ is the minimum and $u^*(v)$ the maximum amount of flow that may pass through v and the cost of sending one such unit through v is $c^*(v)$. By 'passing through' we

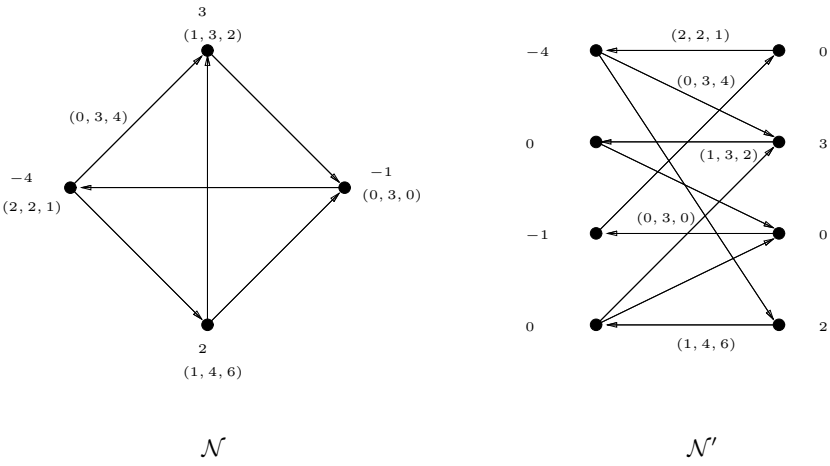
mean the obvious thing when $b(v) = 0$ and if $b(v) > 0$ ($b(v) < 0$) we think of $l^*(v), u^*(v), c^*(v)$ as bounds and costs per unit on the total amount of flow out of (in to) v.

Let D_{ST} be the digraph obtained from $D = (V, A)$ by performing the vertex splitting procedure. Define a new network based on the digraph D_{ST} by adding lower bounds, capacities and costs as follows:

(a) For every arc $i_s j_t$ (corresponding to an arc ij of A) we let $h'(i_s j_t) = h(ij)$, where $h \in \{l, u, c\}$.
(b) For every arc $i_t i_s$ (corresponding to a vertex i of V) we let $h'(i_t i_s) = h^*(i)$, where $h^* \in \{l^*, u^*, c^*\}$.

Finally we define the function b' as follows:

If $b(i) = 0$, then $b'(i_s) = b'(i_t) = 0$;
If $b(i) > 0$, then $b'(i_t) = b(i)$ and $b'(i_s) = 0$;
If $b(i) < 0$, then $b'(i_t) = 0$ and $b'(i_s) = b(i)$.



$\mathcal{N}$ $\mathcal{N}'$

Figure 4.5 The construction of $\mathcal{N}'$ from $\mathcal{N}$. The specification is the balance vector and (l, u, c). For clarity only one arc of $\mathcal{N}$ has a description of bounds and cost.

See Figure 4.5 for an example of the construction. It is not difficult to show the following result.

Lemma 4.2.4 *Let $\mathcal{N}$ and $\mathcal{N}'$ be as described above. Then every feasible flow in $\mathcal{N}$ corresponds to a feasible flow in $\mathcal{N}' = (V(D_{ST}), A(D_{ST}), l', u', b', c')$ and vice versa. Furthermore, the costs of these flows are the same.*

Proof: Exercise 4.6. □

4.3 Flow Decompositions

In this section we consider a network $\mathcal{N} = (V, A, l \equiv 0, u)$ and denote by $D = (V, A)$ the underlying digraph of $\mathcal{N}$. By a path or cycle in $\mathcal{N}$ we mean a directed path or cycle in D. We will show that every flow in a network can be decomposed into a small number of very simple flows in the same network. Besides being a nice elementary mathematical result, this also has very important algorithmic consequences as will be clear from the succeeding sections.

A **path flow** $f(P)$ along a path P in $\mathcal{N}$ is a flow with the property that there is some number $k \in \mathcal{R}_0$ such that $f(P)_{ij} = k$ if ij is an arc of P and otherwise $f(P)_{ij} = 0$. Analogously, we can define a **cycle** flow $f(W)$ for any cycle W in D. The **arc sum** of two flows x, x', denoted $x + x'$, is simply the flow obtained by adding the two flows arc-wise.

Theorem 4.3.1 *Every flow x in $\mathcal{N}$ can be represented as the arc sum of some path and cycle flows $f(P_1), f(P_2), \ldots, f(P_\alpha), f(C_1), \ldots, f(C_\beta)$ with the following two properties:*

(a) Every directed path P_i, $1 \leq i \leq \alpha$, with positive flow connects a source vertex to a sink vertex.
(b) $\alpha + \beta \leq n + m$ and $\beta \leq m$.

Proof: Let x be a non-zero flow in $\mathcal{N}$. Suppose first that $b_x(i_0) > 0$ for some $i_0 \in V$. Since $b_x(i_0) > 0$ it follows from (4.5) that there is some arc $i_0 i_1$ leaving i_0 with $x_{i_0 i_1} > 0$. If $b(i_1) < 0$, then we have found a path from i_0 to the sink i_1. Otherwise $b(i_1) \geq 0$ and it follows from (4.5) and the fact that $x_{i_0 i_1} > 0$ that i_1 has some arc $i_1 i_2$ leaving it with $x_{i_1 i_2} > 0$. Continuing this way, we either find a path P from i_0 to a sink vertex i_k such that x is positive on all arcs on P, or eventually some vertex that was examined previously must be reached for the second time. In the latter case we have detected a cycle $C = i_r i_{r+1} \ldots i_{p-1} i_p i_r$ such that x is positive on all arcs of C. Now we change the flow x as follows:

(i) If we detected a path P from i_0 to a sink i_k, then let $\delta = \min\{x_{i_q i_{q+1}} : i_q i_{q+1} \in A(P)\}$ and define μ by $\mu = \min\{b_x(i_0), -b_x(i_k), \delta\}$. Let $f(P)$ be the path flow of value μ along P. Decrease x by μ units along P.
(ii) Otherwise we have detected a cycle C. Let $\mu = \min\{x_{i_q i_{q+1}} : i_q i_{q+1} \in A(C)\}$ and let $f(C)$ be a cycle flow of value μ along C. Decrease x by μ units along C.

If no arc carries positive flow after the changes made above, we are done. Otherwise we repeat the process above. If every vertex v becomes balanced with respect to the current flow x (i.e., $b_x(v) = 0$) before x is identically zero, then just start from a vertex i_0 which has an arc $i_0 i_1$ with positive flow. From now on only cycle flows will be extracted in the subroutine described above.

Since each of these iterations either results in a vertex becoming balanced with respect to the current flow, or in an arc ij losing all its flow, i.e., x_{ij} becomes zero, the total number of iterations, extracting either a path flow or a cycle flow from the current flow, is at most $n + m$. It follows from the description above that (a) and the first part of (b) holds. The second part of (b) follows from the fact that each time we extract a cycle flow at least one arc loses all its flow. □

The proof above immediately implies an algorithm for finding such a decomposition in time $O(m^2)$ if one uses DFS to find the next path or cycle flow to extract. However, if we use an appropriate data structure and a little care, this complexity can be improved.

Lemma 4.3.2 *Given an arbitrary flow x in $\mathcal{N}$ one can find a decomposition of x into at most $n + m$ path and cycle flows, at most m of which are cycle flows, in time $O(nm)$.*

Proof: Exercise 4.7. □

The following useful fact is an easy consequence of Theorem 4.3.1.

Corollary 4.3.3 *Let $\mathcal{N}$ be a network. Every circulation in $\mathcal{N}$ can be decomposed into no more than m cycle flows.* □

4.4 Working with the Residual Network

Suppose $\mathcal{N}$ is a network and x, x' are feasible flows in $\mathcal{N}$. What can we say about the relation between x and x'? Clearly one can be obtained from the other by changing the flow along each arc appropriately, but we can reveal much more interesting relations as we shall see below. In fact, it turns out that if x is feasible in $\mathcal{N}$ and x' is any other feasible flow in $\mathcal{N}$, then x' can be expressed in terms of x and some feasible flow in the residual network $\mathcal{N}(x)$. The other direction holds as well: if x is feasible in $\mathcal{N}$ and y is feasible in $\mathcal{N}(x)$, then we can 'add' y to x and obtain a new feasible flow in $\mathcal{N}$. These two properties imply that in order to study flows in a network $\mathcal{N}$ it suffices to find one feasible flow x and then work in the residual network $\mathcal{N}(x)$. We assume below that all lower bounds are zero. Recall that, according to the results in Section 4.2, this restriction does not limit our modeling power.

The first result shows that if x is a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ and $\tilde{x}$ is a feasible flow in $\mathcal{N}(x)$, then one can **'add'** $\tilde{x}$ to x and obtain a new feasible flow in $\mathcal{N}$. Here 'adding' is arc-wise and should be interpreted as defined below. Recall that we may assume we are dealing with netto flows.

Definition 4.4.1 *Let x be a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let $\tilde{x}$ be a feasible flow in $\mathcal{N}(x)$. Define the flow $x^* = x \oplus \tilde{x}$ as follows: Start by letting $x^*_{ij} := x_{ij}$ for every $ij \in A$ and then for every arc ij in $\mathcal{N}(x)$ such that $\tilde{x}_{ij} > 0$ we modify x^* as follows (see Figure 4.6).*

(a) If $x_{ji} = 0$, *then* $x^*_{ij} := x_{ij} + \tilde{x}_{ij}$.
(b) If $x_{ij} = 0$ *and* $x_{ji} < \tilde{x}_{ij}$, *then* $x^*_{ij} := \tilde{x}_{ij} - x_{ji}$ *and* $x^*_{ji} := 0$.
(c) If $x_{ji} \geq \tilde{x}_{ij}$, *then* $x^*_{ji} := x_{ji} - \tilde{x}_{ij}$.

Note that by (4.7), if $0 < x_{ji} < \tilde{x}_{ij}$, then $ij \in A$. Using that x is a netto flow it is easy to check that the resulting flow x^* is also a netto flow.

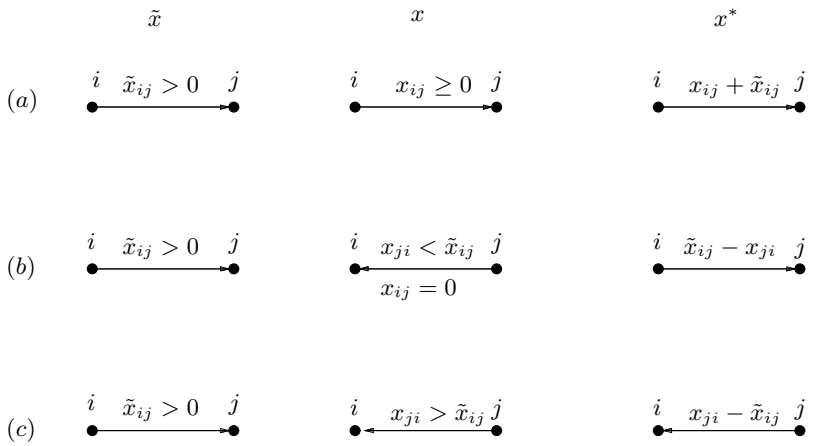


Figure 4.6 The three different cases in Definition 4.4.1. The three columns shows the flows $\tilde{x}$, x and x^*, respectively. An arc between i and j is shown unless the corresponding flow on that arc is zero.

Theorem 4.4.2 *Let* x *be a feasible flow in* $\mathcal{N} = (V, A, l \equiv 0, u, c)$ *with balance vector* b_x *and* $\tilde{x}$ *is a feasible flow in* $\mathcal{N}(x) = (V, A(x), r, c)$ *with balance vector* $b_{\tilde{x}}$. *Then* $x^* = x \oplus \tilde{x}$ *is a feasible flow in* $\mathcal{N}$ *with balance vector* $b_x + b_{\tilde{x}}$ *and the cost of* x^* *is given by* $c^T x^* = c^T x + c^T \tilde{x}$.

Proof: Let us first show that $0 \leq x^*_{ij} \leq u_{ij}$ for every $ij \in A$. We started the construction of x^* by letting $x^*_{ij} := x_{ij}$ for every arc. Hence it suffices to consider pairs (i, j) for which $\tilde{x}_{ij} > 0$. We consider the three possible cases (a)-(c) in Definition 4.4.1. In Case (a) we have $x^*_{ji} = 0$ and

$$\begin{aligned} 0 < x^*_{ij} = x_{ij} + \tilde{x}_{ij} &\leq x_{ij} + r_{ij} \\ &= x_{ij} + (u_{ij} - x_{ij} + x_{ji}) \\ &= u_{ij}, \end{aligned}$$

since we have $x_{ji} = 0$ in Case (a). In Case (b) we will have $x^*_{ji} = 0$ and

$$\begin{aligned} 0 \leq x^*_{ij} = \tilde{x}_{ij} - x_{ji} &\leq r_{ij} - x_{ji} \\ &= (u_{ij} - x_{ij} + x_{ji}) - x_{ji} \\ &= u_{ij}, \end{aligned}$$

since we have $x_{ij} = 0$ in Case (b). In Case (c) it is easy to see that we get $x^*_{ij} = 0$ and that $0 \leq x^*_{ji} < u_{ji}$.

Consider the balance vector of the resulting flow. We wish to prove that x^* has balance vector $b_x + b_{\tilde{x}}$, that is, for every $i \in V$,

$$b_{x^*}(i) = \sum_{ij \in A} x^*_{ij} - \sum_{ji \in A} x^*_{ji} = b_x(i) + b_{\tilde{x}}(i). \tag{4.9}$$

This can be proved directly from the definitions of the balance expressions for x and $\tilde{x}$. However, this approach is rather tedious and there is a simple inductive proof using Theorem 4.3.1. If $\tilde{x}$ is just a cycle flow in $\mathcal{N}(x)$, then it is easy to see (Exercise 4.12) that the balance vector of x^* equals that of x. Similarly, if $\tilde{x}$ is just a path flow of value δ along a (p, q)-path, for some distinct vertices $p, q \in V$, then $b_{x^*}(v) = b_x(v)$ for vertices v which are either internal vertices on P or not on P and $b_{x^*}(p) = b_x(p) + \delta$, $b_{x^*}(q) = b_x(q) - \delta$. In the general case, when $\tilde{x}$ is neither a path flow nor a cycle flow in $\mathcal{N}(x)$ we consider a decomposition of $\tilde{x}$ into path and cycle flows in $\mathcal{N}(x)$ according to Theorem 4.3.1. Using the observation above and Theorem 4.3.1 (implying that when adding all balance vectors of the paths and cycles in a decomposition, we obtain the balance vector of $\tilde{x}$) it is easy to prove by induction on the number of paths and cycles in the decomposition that (4.9) holds.

We leave it to the reader to prove using the same approach as above that the cost of x^* is given by $c^T x^* = c^T x + c^T \tilde{x}$ (see Exercise 4.12). □

The next theorem shows that the difference between any two feasible flows in a network can be expressed as a feasible flow in the residual network with respect to any of those flows.

Theorem 4.4.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be a network and let x and x' be feasible netto flows in $\mathcal{N}$ with balance vectors b_x and $b_{x'}$. There exists a feasible flow $\bar{x}$ in $\mathcal{N}(x)$ with balance vector $b_{\bar{x}} = b_{x'} - b_x$ such that $x' = x \oplus \bar{x}$. Furthermore, the costs of these flows satisfy $c^T \bar{x} = c^T x' - c^T x$.*

Proof: Let x, x' be feasible netto flows in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and define a flow in $\mathcal{N}(x)$ as follows. For every arc $pq \in \mathcal{N}(x)$ we let $\bar{x}_{pq} := 0$ and then for every arc $ij \in A$ such that either $x_{ij} > 0$ or $x'_{ij} > 0$ holds, we modify $\bar{x}$ as follows:

(a) If $x_{ij} > x'_{ij}$, then $\bar{x}_{ji} := x_{ij} - x'_{ij} + x'_{ji}$.
(b) If $x'_{ij} > x_{ij}$, then $\bar{x}_{ij} := x'_{ij} - x_{ij} + x_{ji}$.

Using that x and x' are feasible netto flows in $\mathcal{N}$, one can verify that $\bar{x}$ is a feasible netto flow in $\mathcal{N}(x)$ (Exercise 4.13). It also follows easily from Definition 4.4.1 that $x' = x \oplus \bar{x}$. Now the last two claims regarding balance vector and cost follow from Theorem 4.4.2. □

The following immediate corollary of Theorem 4.4.3 and Corollary 4.3.3 will be useful when we study minimum cost flows in Section 4.10.

Corollary 4.4.4 *If x and x' are feasible flows in the network $\mathcal{N} = (V, A, l \equiv 0, u, c)$ such that $b_x = b_{x'}$, then there exists a collection of at most m cycles $W_1, W_2, \ldots, W_k$ in $\mathcal{N}(x)$ and cycle flows $f(W_1), \ldots, f(W_k)$ in $\mathcal{N}(x)$ such that the following holds:*

(a) $x' = x \oplus (f(W_1) + \ldots + f(W_k)) = (\ldots((x \oplus f(W_1)) \oplus f(W_2)) \oplus \ldots) \oplus f(W_k)$;
(b) $c^T x' = c^T x + \sum_{i=1}^{k} c^T f(W_i)$. □

4.5 The Maximum Flow Problem

In this and the next section we study (s,t)-flows in networks with all lower bounds equal to zero. That is, we consider networks of the type $\mathcal{N} = (V, A, l \equiv 0, u)$ where $s, t \in V$ are special vertices and we are only interested in flows x which satisfy $b_x(s) = -b_x(t)$ and $b_x(v) = 0$ for all other vertices. We call s the **source** and t the **sink** of $\mathcal{N}$. By Theorem 4.3.1, every (s,t)-flow x can be decomposed into a number of path flows along (s,t)-paths and some cycle flows whose values do not affect the value of the flow x. Based on this observation we also say that x is a flow **from s to t**.

Recall from (4.8) that the value $|x|$ of an (s,t)-flow is $|x| = b_x(s)$. We are interested in determining the maximum value k for which $\mathcal{N}$ has a feasible (s,t)-flow of value[4] k. Such a flow is called a **maximum flow** in $\mathcal{N}$. The problem of finding a maximum flow from s to t in a network with a specified source s and sink t is known as the MAXIMUM FLOW PROBLEM [331].

An **(s,t)-cut** is a set of arcs of the form $(S, \overline{S})$ where $S, \overline{S}$ form a partition of V such that $s \in S, t \in \overline{S}$. The **capacity** of an (s,t)-cut $(S, \overline{S})$ is the number $u(S, \overline{S})$, that is, the sum of the capacities of arcs with tail in S and head in $\overline{S}$ (recall (4.4)). Cuts of this kind are interesting in relation to the maximum flow problem as we shall see below.

Lemma 4.5.1 *For every (s,t)-cut $(S, \overline{S})$ and every (s,t)-flow x, we have*

$$|x| = x(S, \overline{S}) - x(\overline{S}, S). \tag{4.10}$$

Proof: Starting from the definition of $|x|$ and the fact that $b_x(v) = 0$ for all $v \in S - s$ we obtain

[4] Observe that there always exists a feasible flow in $\mathcal{N}$ since we have assumed $l \equiv 0$.

$$\begin{aligned}|x| &= b_x(s) + \sum_{i \in S-s} b_x(i) \\ &= \sum_{i \in S} (\sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji}) \\ &= x(S, V) - x(V, S) \\ &= x(S, S) + x(S, \overline{S}) - x(\overline{S}, S) - x(S, S) \\ &= x(S, \overline{S}) - x(\overline{S}, S),\end{aligned}$$

where we also used (4.4). □

Since a feasible flow x satisfies $x \leq u$, every feasible (s,t)-flow must satisfy

$$x(S, \overline{S}) \leq u(S, \overline{S}) \text{ for every } (s,t)\text{-cut } (S, \overline{S}). \tag{4.11}$$

A **minimum (s,t)-cut** is an (s,t)-cut $(S, \overline{S})$ with

$$u(S, \overline{S}) = \min\{u(S', \overline{S'}) : (S', \overline{S'}) \text{ is an } (s,t)\text{-cut in } \mathcal{N}\}.$$

It follows from (4.11) and Lemma 4.5.1 that the capacity of any (s,t)-cut provides an upper bound for the value $|x|$ for any feasible flow x in the network. We also obtain the following useful consequence.

Lemma 4.5.2 *If a flow x has value $|x| = u(S, \overline{S})$ for some (s,t)-cut $(S, \overline{S})$, then $x(\overline{S}, S) = 0$, x is a maximum (s,t)-flow and $(S, \overline{S})$ is a minimum (s,t)-cut.* □

Suppose x is an (s,t)-flow in $\mathcal{N}$ and P is an (s,t)-path in $\mathcal{N}(x)$ such that $r_{ij} \geq \epsilon > 0$ for each arc ij on P. Let x'' be the (s,t)-path flow of value ϵ in $\mathcal{N}(x)$ which is obtained by sending ϵ units of flow along the path P. By Theorem 4.4.2, we can obtain a new flow $x' = x \oplus x''$ of value $|x| + \epsilon$ in $\mathcal{N}$, implying that x is not a maximum flow in $\mathcal{N}$. We call a path P in $\mathcal{N}(x)$ as above an **augmenting path** with respect to x. The **capacity** $\delta(P)$ of an augmenting path P is given by

$$\delta(P) = \min\{r_{ij} : ij \text{ is an arc of } P\}. \tag{4.12}$$

We call an arc ij of P for which $x_{ij} < u_{ij}$ a **forward arc** of P and an arc ij of P for which $x_{ji} > 0$ a **backward arc** of P.

When we 'add' the path flow x'' to x according to Definition 4.4.1 we say that we **augment along P** by ϵ units. It follows from the definition of $\delta(P)$ and Theorem 4.4.2 that $\delta(P)$ is the maximum value by which we can augment x along P and still have a feasible flow in $\mathcal{N}$ after the augmentation.

Now we are ready to prove the following fundamental result, due to Ford and Fulkerson, relating minimum (s,t)-cuts and maximum (s,t)-flows.

Theorem 4.5.3 (Max-Flow Min-Cut theorem) *[331] Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. For every feasible (s,t)-flow x in $\mathcal{N}$ the following are equivalent:*

(a) The flow x is a maximum (s,t)-flow.
(b) There is no (s,t)-path in $\mathcal{N}(x)$.
(c) There exists an (s,t)-cut $(S, \overline{S})$ such that $|x| = u(S, \overline{S})$.

Proof: We show that (a)⇒(b)⇒(c)⇒(a).

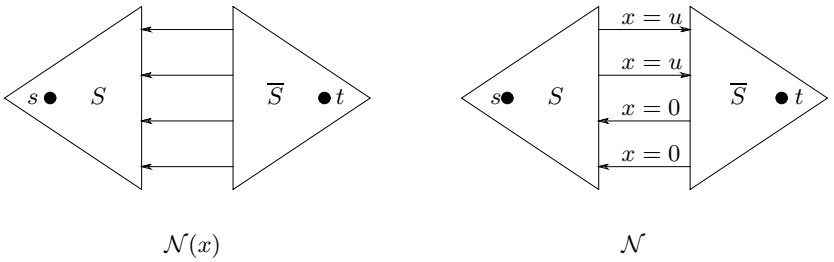


Figure 4.7 Illustration of part (b)⇒(c) in the proof of Theorem 4.5.3. The set S consists of those vertices that are reachable from s in $\mathcal{N}(x)$. The left part shows the situation in the residual network where we have $\overline{S} \Rightarrow S$ and the right part shows the corresponding situation in $\mathcal{N}$.

(a)⇒(b): Suppose x is a maximum flow in $\mathcal{N}$ and that $\mathcal{N}(x)$ contains an (s,t)-path P. Let $\delta(P) > 0$ be the capacity of P and let x' be the (s,t)-path flow in $\mathcal{N}(x)$ which sends $\delta(P)$ units of flow along P. By Theorem 4.4.2, $x \oplus x'$ is a feasible flow in $\mathcal{N}$ of value $|x| + \delta(P) > |x|$, contradicting the maximality of x. Hence (a)⇒(b).

(b)⇒(c): Suppose that $\mathcal{N}(x)$ contains no (s,t)-path. Let

$$S = \{y \in V : \mathcal{N}(x) \text{ contains an } (s,y)\text{-path}\}.$$

By the definition of S, there is no arc from S to $\overline{S}$ in $\mathcal{N}(x)$. Thus the definition of $\mathcal{N}(x)$ implies that for every arc $ij \in (S, \overline{S})$ we have $x_{ij} = u_{ij}$ and for every arc $ij \in (\overline{S}, S)$ we have $x_{ij} = 0$ (see Figure 4.7). This implies that we have $|x| = x(S, \overline{S}) - x(\overline{S}, S) = u(S, \overline{S}) - 0 = u(S, \overline{S})$. Hence we have proved that (b)⇒(c).

(c)⇒(a): This follows directly from Lemma 4.5.2. □

4.5.1 The Ford-Fulkerson Algorithm

The proof of Theorem 4.5.3 suggests the following simple method for finding a maximum (s,t)-flow in a network where all lower bounds are zero. Start

with $x \equiv 0$. This is a feasible flow since $0 = l_{ij} \leq u_{ij}$ for all arcs $ij \in A$. Try to find an (s,t)-path P in $\mathcal{N}(x)$. If there is such a path P, then augment x by $\delta(P)$ units along P. Continue this way until there is no (s,t)-path in $\mathcal{N}(x)$ where x is the current flow. This method, due to Ford and Fulkerson [331], is called the **Ford-Fulkerson (FF) algorithm**.

Strictly speaking this is not really an algorithm if we do not specify how we wish to search for an augmenting (s,t)-path. It can be shown (see Exercise 4.17) that when the capacities are allowed to take non-rational values and there is no restriction on the choice of augmenting paths (other than the restriction that one has to augment as much as possible along the current path), then the process above may continue indefinitely and without even converging to the right value of a maximum flow (see Exercise 4.17). In real-life applications this problem cannot occur since all numbers represented in computers are rational approximations of real numbers and in this case the algorithm will always terminate (Exercise 4.18).

Theorem 4.5.4 *If $\mathcal{N} = (V, A, l \equiv 0, u)$ has all capacities integers, then the Ford-Fulkerson algorithm finds a maximum (s,t)-flow in time $O(m|x^*|)$, where x^* is a maximum (s,t)-flow.*

Proof: The following generic process called the **labelling algorithm** will find an augmenting path in $\mathcal{N}(x)$ in time $O(n+m)$ if one exists[5]. Start with all vertices unlabelled except s and every vertex unscanned. In the general step we pick a labelled but unscanned vertex v and scan all its out-neighbours while labelling (by backwards pointers showing where a vertex got labelled from) those vertices among the out-neighbours of v that are unlabelled. If t becomes labelled this way, the process stops and an augmenting path, determined by the backwards pointers, is returned. If all vertices are scanned and t was not labelled, the process stops and the set of labelled vertices S and its complement $\overline{S}$ correspond to a minimum (s,t)-cut (recall the proof of Theorem 4.5.3).

Each time we augment along an augmenting path, the value of the current flow increases by at least one, since the capacities in the residual network are all integers (this is clear in the first iteration and easy to establish by induction for the rest of the iterations of the algorithm). Hence there can be no more than $|x^*|$ iterations of the above search for a path and the complexity follows. □

To see that the seemingly very pessimistic estimate in Theorem 4.5.4 for the time spent by the algorithm may in fact be realized, consider the network in Figure 4.8 and the sequence of augmenting paths specified there. The reader familiar with the literature on flows may see that our example is different from the classical example in books on flows. The reason for this is

[5] We could also use path finding algorithms such as BFS and DFS, but the original algorithm by Ford and Fulkerson uses only the generic labelling approach. See also Section 4.6.

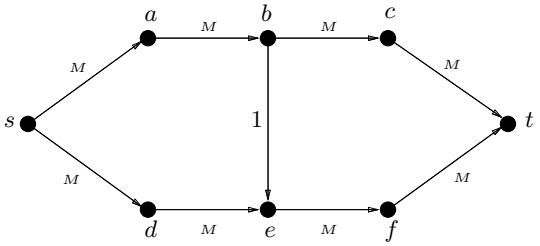


Figure 4.8 A possibly bad network for the Ford-Fulkerson algorithm. The number M denotes a large integer. If we choose augmenting paths of the form $sabeft$ with augmenting capacity 1 in odd-numbered iterations and augmenting paths of the form $sdebct$ with augmenting capacity 1 in even-numbered iterations, then a maximum flow x of value $2M$ will be found only after $2M$ augmentations. Clearly, if instead we augment first along $sabct$ and then along $sdeft$, each time by M units, we can find a maximum flow after just two augmentations.

that if we interpret the Ford-Fulkerson algorithm precisely as it is described in [331, page 18] (see also the proof of Theorem 4.5.4), then the algorithm will not behave badly on the usual example, whereas it still will do so on the example in Figure 4.8.

The value of the maximum flow in the example in Figure 4.8 is $2M$. This shows that the complexity of the Ford-Fulkerson algorithm is not bounded by a polynomial in the size of the input (recall from Chapter 1 that we assume that numbers are represented in binary notation). It is worth noting though that Theorem 4.5.4 implies that if all capacities are small integers, then we get a very fast algorithm which, due to its simplicity, is easy to implement. The following is an easy but very important consequence of the proof of Theorem 4.5.3:

Theorem 4.5.5 (Integrality theorem for maximum (s,t)-flows) *[331] Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. If all capacities are integers, then there exists an integer maximum (s,t)-flow in $\mathcal{N}$.*

Proof: This follows from our description of the Ford-Fulkerson algorithm. We start with $x \equiv 0$ and every time we augment the flow we do this by adding an integer-valued path flow of value $\delta(P) \in \mathcal{Z}_+$. Hence the new (s,t)-flow is also an integer flow. It follows from the fact that all capacities are integers that in a finite number of steps we will reach a maximum flow (by Lemma 4.5.1, $|x|$ cannot exceed the capacity of any cut). Now the claim follows by induction on the number of augmentations needed before we have a maximum flow. □

An (s,t)-flow in a network $\mathcal{N}$ is **maximal** if every (s,t)-path in $\mathcal{N}$ uses at least one arc pq such that $x_{pq} = u_{pq}$ (such an arc is called **saturated**). That is, either x is maximum or after augmenting along an augmenting path P the resulting flow x' has $x'_{ij} < x_{ij}$ for some arc[6]. This is equivalent to saying that

[6] Recall that we always work with netto flows.

every augmenting path with respect to x contains at least one backward arc when P is considered as an oriented path in $\mathcal{N}$. It is important to distinguish between a maximal flow and a maximum flow. An (s,t)-flow x is maximal if it is either maximum, or in order to augment it to a flow with a higher value, we must reduce the flow in some arc. See also Figure 4.9.

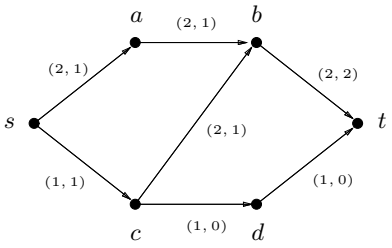


Figure 4.9 A network $\mathcal{N}$ with flow x which is maximal but not maximum as the path $P = sabcdt$ is an (s,t)-path in $\mathcal{N}(x)$. Note that the arc bc is a backward arc of P. The data on each arc is (capacity, flow).

4.5.2 Maximum Flows and Linear Programming

We digress for a short while to give some remarks on the relation between maximum flows and linear programming. First observe that the maximum flow problem (with lower bounds all equal to zero) is equivalent to the following linear programming problem:

$$\begin{aligned} &\text{maximize} && k \\ &\text{subject to} \\ &&& b_x(v) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise.} \end{cases} \\ &&& 0 \le x_{ij} \le u_{ij} \qquad \text{for every } ij \in A. \end{aligned}$$

The matrix T of the constraints of this linear program is given by $T = \begin{bmatrix} S \\ I \end{bmatrix}$, where S is the vertex-arc incidence matrix[7] of the underlying directed graph of the network (recall the definition of b_x) and I is the $m \times m$ identity matrix. The matrix S has the property that every column contains exactly $+1$ and exactly one -1. This implies that S is **totally unimodular**, i.e., each

[7] The **vertex-arc incidence matrix** $S = [s_{ij}]$ of a digraph $D = (V, A)$ has rows labelled by the vertices of V and columns labelled by the arcs of A and the entry s_{v_i,a_j} equals 1 if the arc a_j has tail v_i, -1 if a_j has head v_i and 0, otherwise.

square submatrix of S has determinant 0, 1 or -1 (see, e.g., the book [229] by Cook, Cunningham, Pulleyblank and Schrijver). Hence it follows from Exercise 4.19 that the matrix T is also totally unimodular. Therefore the integrality theorem for maximum flows (Theorem 4.5.5) follows immediately from the Hoffman-Kruskal characterization of total unimodularity (see [229, Theorem 6.25]).

Since the maximum flow problem is just a linear programming problem, it follows that one can find a maximum flow using any method for solving general linear programming problems. In particular, by the total unimodularity of T, the Simplex algorithm will always return an integer maximum flow, provided that all capacities are integers. However, due to the special nature of the problem, more efficient algorithms can be found when we exploit the structure of flow problems. Finally, we remark that the max-flow min-cut theorem can be derived from the duality theorem for linear programming (see, e.g., the book [742] by Papadimitriou and Steiglitz).

4.6 Polynomial Algorithms for Finding a Maximum (s, t)-Flow

In this section and in some of the exercises we describe a number of polynomial algorithms for the maximum flow problem. For a survey of complexities of various maxflow algorithms see, e.g., Chapter 10 of Schrijver's book [803]. The following problem has been folklore for many years.

Problem 4.6.1 *Is there an $O(mn)$ algorithm for the maximum flow problem?*

The Ford-Fulkerson algorithm can be modified in various ways to ensure that it becomes a polynomial algorithm. We describe two such modifications (see also Exercises 4.25 and 4.26). After doing so we describe a different approach in which we do not augment the flow by just one path at the time. For the first two subsections we need the following definition.

Definition 4.6.2 *A* **layered network** *is a network $\mathcal{N} = (V, A, l \equiv 0, u)$ with the following properties:*

(a) There is a partition $V = V_0 \cup V_1 \cup V_2 \cup \ldots \cup V_k \cup V_{k+1}$ such that $V_0 = \{s\}, V_{k+1} = \{t\}$ and
(b) every arc of A goes from a **layer** *V_i to the next layer V_{i+1} for some $i = 0, 1, \ldots, k$.*

See Figure 4.10 for an example of a layered network.

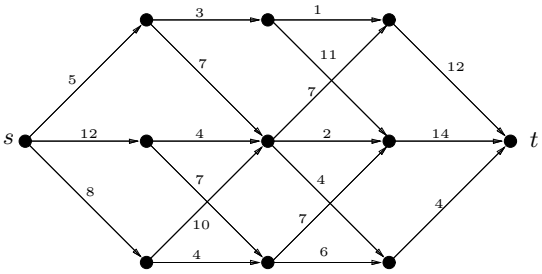


Figure 4.10 A layered network with source s and sink t. The numbers on the arcs indicate the capacities.

4.6.1 Augmenting Along Shortest Augmenting Paths

Edmonds and Karp [288] observed that in order to modify the Ford-Fulkerson algorithm so as to get a polynomial algorithm, it suffices to choose the augmenting paths as shortest paths with respect to the number of arcs on the path.

Let x be a feasible (s,t)-flow in a network $\mathcal{N}$. Denote by $\delta_x(s,t)$ the length of a shortest (s,t)-path in $\mathcal{N}(x)$. If no such path exists, we let $\delta_x(s,t) = \infty$.

Suppose that there is an augmenting path in $\mathcal{N}(x)$ and let P be a shortest such path. Let r be the number of arcs in P. Define the network $\mathcal{LN}(x)$ as the network one obtains from $\mathcal{N}(x)$ by taking the vertices from the distance classes $V_0, V_1, \ldots, V_r$, i.e., $V_i = \{v : \text{dist}_{\mathcal{N}(x)}(s,v) = i\}$, and all arcs belonging to $(V_i, V_{i+1})_{\mathcal{N}(x)}$ for $i = 0, 1, \ldots, r-1$ along with their residual capacities r_{ij}. Observe that, by the definition of distance classes, $\mathcal{LN}(x)$ contains all the shortest augmenting paths with respect to x in $\mathcal{N}(x)$.

The crucial fact that makes augmenting along shortest paths a good approach is the following lemma.

Lemma 4.6.3 *[288] Let x be a feasible (s,t)-flow in $\mathcal{N}$ and let x' be obtained from x by augmenting along a shortest path in $\mathcal{N}(x)$. Then*

$$\delta_{x'}(s,t) \geq \delta_x(s,t). \tag{4.13}$$

Proof: Suppose this is not the case for some x, x' where x' is obtained from x by augmenting along a shortest path P in $\mathcal{N}(x)$. By the remark above, $\mathcal{LN}(x)$ contains all the shortest augmenting paths (with respect to x) in $\mathcal{N}(x)$. Let $r = \delta_x(s,t)$. By our assumption, $\mathcal{N}(x')$ contains an (s,t)-path P' whose length is less than r. Thus P' must use an arc ij such that $ij \notin A(\mathcal{N}(x))$. However, every arc that is in $\mathcal{N}(x')$ but not in $\mathcal{LN}(x)$ is of the form ji where ij is an arc of P, or is inside a layer of $\mathcal{LN}(x)$. It follows that P' has at least $r+1$ arcs, contradicting the assumption. □

Note that even if $\mathcal{N}(x')$ contains no (s,t)-path of length $\delta_x(s,t)$, it may still contain a path of length $\delta_x(s,t)+1$, since we may use an arc which was inside a layer of $\mathcal{LN}(x)$.

Theorem 4.6.4 (Edmonds, Karp) *[288] If we always augment along shortest augmenting paths, then the Ford-Fulkerson algorithm has complexity* $O(nm^2)$.

Proof: By Lemma 4.6.3, the length of the current augmenting path increases monotonically throughout the execution of the algorithm. It follows from the proof of Lemma 4.6.3 that if the length of the next augmenting path does not go up, then that path is also a path in $\mathcal{LN}(x)$. Note also that at least one arc from some layer V_i to the next disappears after each augmentation[8]. Hence the number of iterations in which the length of the current augmenting path stays constant is at most m. Since the length can increase at most $n-2$ times (the length of an (s,t)-path is at least 1 and at most $n-1$) and we can find the next augmenting path in time $O(n+m)$ using BFS we obtain the desired complexity. □

Zadeh [926] constructed networks with n vertices and m arcs for which the Edmonds-Karp algorithm requires $\Omega(nm)$ augmentations to find a maximum flow. Hence the estimate on the worst case complexity is tight.

4.6.2 Maximal Flows in Layered Networks

Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$ be a layered network with $V_0 = \{s\}$ and $V_k = \{t\}$. By the definition of a maximal flow in Section 4.5, an (s,t)-flow x in $\mathcal{L}$ is maximal if there is no (s,t)-path of length k in the residual network $\mathcal{L}(x)$. That is, every augmenting path with respect to x (if there is any) must use at least one arc pq such that $p \in V_j, q \in V_i$ for some $j \geq i$.

We saw above that if we always augment along shortest augmenting paths, then the length of a shortest augmenting path is monotonically increasing. Hence if we have a method to find a maximal flow in a layered network in time $O(p(n,m))$, then we can use that method to obtain an $O(np(n,m))$ algorithm for finding a maximum (s,t)-flow in any given network.

The method of Edmonds and Karp above achieves a maximal flow in time $O(m^2)$. It was observed by Dinic [262] (who also independently and earlier discovered the method of using shortest augmenting paths) that a maximal flow in a layered network can be obtained in time $O(nm)$, thus resulting in an $O(n^2m)$ algorithm for maximum flow.

The idea is to search for a shortest augmenting path in a depth-first search manner. We modify slightly the standard DFS algorithm (see Section 1.9) as

[8] Recall that in each augmentation we augment by $\delta(P)$ units along the current augmenting path P.

shown below. The vector π is used to remember the arcs of the augmenting path detected if one is found.

Dinic's algorithm (one phase)
Input: A layered network $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$.
Output: A maximal flow x in $\mathcal{L}$.

1. **Initialization:** $x_{ij} := 0$ for every arc ij in A, let $v := s$ be the current vertex and let $A' := A$.
2. **Searching step:** If there is no arc with tail v in A' (from v to the next layer among the remaining arcs), then if $v = s$, go to Step 5; otherwise go to Step 4;
   If there is an arc $vw \in A'$, then let $\pi(w) := v$ and let $v := w$. If $v \neq t$, repeat Step 2.
3. **Augmentation step:** Using the π labels find the augmenting path P detected and augment x along P by $\delta(P)$ units. Delete all arcs ij of A' for which $x_{ij} = u_{ij}$. Erase all labels on vertices ($\pi(i) := nil$ for all $i \in V$). Let $v := s$ and go to Step 2.
4. **Arc deletion step:** (The search above has revealed that there is no (v,t)-path in the current digraph $D' = (V, A')$. Furthermore, $v \neq s$.) Delete all arcs with head or tail v from A', let $v := \pi(v)$ and go to Step 2.
5. **Termination:** Return the maximal flow x.

Theorem 4.6.5 *Dinic's algorithm (one phase of) correctly determines a maximal flow in a given layered network $\mathcal{L}$ in time $O(nm)$.*

Proof: Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$. Each time the current flow is augmented in the algorithm it is changed along an augmenting path of length k. We only delete an arc from A' when it is no longer present in the residual network $\mathcal{L}(x)$ where x is the current flow. Hence no deleted arc could be used in an augmenting path of length k with respect to the current flow. Furthermore, when the algorithm terminates there is no (s,t)-path in the current digraph $D' = (V, A')$. Here A' consists of those arcs from one layer to the next which are still not filled to capacity by the current x. It follows that the algorithm terminates with a maximal flow.

The complexity follows from the fact that we perform at most $O(n)$ steps between each deletion of an arc which is either saturated (via the actual augmenting path P) or enters a vertex for which we deleted all arcs having that vertex as the head or tail (see Step 4). □

4.6.3 The Push-Relabel Algorithm

The flow algorithms we have seen in the previous sections have the common feature that they all increase the flow along one augmenting path at a time. Very often, when searching for an augmenting path, one finds a path P containing an arc rq whose capacity is relatively small compared to the capacity

of the prefix $P[s,r]$ of that path. This means that along $P[s,r]$ we were able to augment by a large amount of flow, but due to the smaller capacity of the arc rq we only augment by that smaller amount and start all over again. In Dinic's algorithm this could be taken into account by not starting all over again, but instead backtracking until a new forward arc can be found in the layered network. However, we are still limited to finding one path at a time. Now we present a different approach, which allows one to work with more than one augmenting path at a time. We will discuss an algorithm of Goldberg and Tarjan [415, 416] called the **push-relabel algorithm**. The basic idea is to try to push as much flow towards t as possible, by first sending the absolute maximum possible, namely, $\sum_{sr\in A} u_{sr}$, out of s and then trying to push this forward to t. At some point no more flow can be sent to t and the algorithm returns the excess flow back to s again. This very vague description will be made precise below (the reader should compare this with the so-called MKM algorithm described in Exercise 4.25).

Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. A feasible flow x in $\mathcal{N}$ is called a **preflow** if $b_x(v) \leq 0$ for all $v \in V - s$. Note that every (s,t)-flow x is also a preflow since we have $b_x(v) = 0$ if $v \in V - \{s,t\}$ and $b_x(t) = -b_x(s) \leq 0$. Hence preflows generalize (s,t)-flows, an observation that we shall use below. Preflows were originally introduced by Karzanov [586]. Let x be a preflow in a network $\mathcal{N}$. A **height function with respect to x** is a function $h : V \to \mathcal{Z}_0$ which satisfies

$$\begin{aligned} &h(s) = n, \quad h(t) = 0; \\ &h(p) \leq h(q) + 1 \qquad \text{for every arc } pq \text{ of } \mathcal{N}(x). \end{aligned} \tag{4.14}$$

The following useful lemma is an immediate consequence of Theorem 4.3.1(a).

Lemma 4.6.6 *Let x be a preflow in a network $\mathcal{N} = (V, l \equiv 0, u)$ with source s and sink t and let v be a vertex such that $b_x(v) < 0$. Then $\mathcal{N}(x)$ contains a (v,s)-path.*

Proof: By the definition of a preflow, s is the only vertex r for which we have $b_x(r) > 0$. Hence, by Theorem 4.3.1(a), every decomposition of x into path and cycle flows contains an (s,v)-path P. Now it follows that $\mathcal{N}(x)$ contains a (v,s)-path, since every arc of P has positive flow in $\mathcal{N}$ and hence gives rise to an oppositely oriented arc in $\mathcal{N}(x)$. □

Now we are ready to describe the (generic) push-relabel algorithm. During the execution of the algorithm, a vertex $v \in V$ is called **active** if $b_x(v) < 0$. An arc pq of $\mathcal{N}(x)$ is **admissible** if $h(p) = h(q) + 1$. The algorithm uses two basic operations **push** and **lift**.

push(pq): Let p be a vertex with $b_x(p) < 0$ and let pq be an admissible arc in $\mathcal{N}(x)$. The operation **push**(pq) changes x_{pq} to $x_{pq} + \rho$, where $\rho = \min\{-b_x(p), r_{pq}\}$.

lift(p): Let p be a vertex with $b_x(p) < 0$ and $h(p) \leq h(q)$ for every arc pq in $\mathcal{N}(x)$. The operation **lift**(p) changes the height of p as follows:

$$h(p) := \min\{h(z) + 1 : pz \text{ is an arc of } \mathcal{N}(x)\}.$$

By the remark after the proof of Lemma 4.6.6, the number $h(p)$ is well-defined.

Lemma 4.6.7 *Let x be a preflow in $\mathcal{N}$ and let h be defined as in (4.14). If $p \in V$ satisfies $b_x(p) < 0$, then at least one of the operations* **push**(pq), **lift**(p) *can be applied.*

Proof: Suppose $b_x(p) < 0$, but we cannot perform a push from p. Then there is no admissible arc with tail p and hence we have $h(p) \leq h(q)$ for every arc pq in $\mathcal{N}(x)$. It follows from Lemma 4.6.6 that there is at least one arc out of p in $\mathcal{N}(x)$ and hence we can perform the operation **lift**(p). □

The generic push-relabel algorithm
Input: A network $\mathcal{N} = (V, l \equiv 0, u)$ with source s and sink t.
Output: A maximum (s,t)-flow in $\mathcal{N}$.
Preprocessing step:

(a) For each $p \in V$ let $h(p) := \text{dist}_{\mathcal{N}}(p,t)$;
(b) Let $h(s) := n$;
(c) Let $x_{sp} := u_{sp}$ for every arc out of s in $\mathcal{N}$;
(d) Let $x_{ij} := 0$ for all other arcs in $\mathcal{N}$.

Main loop:
While there is an active vertex $p \in V - t$ do the following:
if $\mathcal{N}(x)$ contains an admissible arc pq, then **push**(pq) else **lift**(p).

Theorem 4.6.8 *The generic push-relabel algorithm correctly determines a maximum (s,t)-flow in $\mathcal{N}$ in time $O(n^2m)$.*

Proof: We first show that the function h remains a height function throughout the execution of the algorithm. Initially this is the case since we use exact distance labels and there are no arcs out of s in $\mathcal{N}(x)$ (Exercise 4.20). Observe that for every vertex p, $h(p)$ is only changed when we perform the operation **lift**(p) and then it is changed so as to preserve the condition (4.14). Furthermore, the operation **push**(pq) may introduce a new arc qp in $\mathcal{N}(x)$, but this arc will satisfy $h(q) = h(p) - 1$ and hence does not violate (4.14). It follows that h remains a height function throughout the execution of the algorithm.

It is easy to see that x remains a preflow throughout the execution of the algorithm, since only a push operation affects the current x and by definition a push operation preserves the preflow condition.

Now we prove that if the algorithm terminates, then it does so with a maximum flow x. Suppose that the algorithm has terminated. This means that no vertex $v \in V$ has $b_x(v) < 0$. Thus it follows from the definition of a preflow that x is an (s,t)-flow. To prove that x is indeed a maximum flow, it suffices to show that there is no (s,t)-path in $\mathcal{N}(x)$. This follows immediately from the fact that h remains a height function throughout the execution of the algorithm. By (4.14), every arc pq in $\mathcal{N}(x)$ has $h(p) \leq h(q) + 1$ and we always have $h(s) = n, h(t) = 0$. Since no (s,t)-path has more than $n-1$ arcs, there is no (s,t)-path in $\mathcal{N}(x)$ and hence, by Theorem 4.5.3, x is a maximum (s,t)-flow.

To prove that the algorithm terminates and to determine its complexity, it is useful to distinguish between two kinds of pushes. An execution of the operation **push**(pq) is a **saturating push** if the arc pq is filled to capacity after the push and hence pq is not an arc of $\mathcal{N}(x)$ immediately after that push. A push which is not saturating is an **unsaturating push**.

We now establish a number of claims from which the complexity of the algorithm follows.

(A) **The total number of lifts is $O(n^2)$:** By Lemma 4.6.6, every vertex p with $b_x(p) < 0$ has a path to s in $\mathcal{N}(x)$. Hence, we have $h(p) \leq 2n-1$, by (4.14). Since the height of a vertex p increases by at least one every time the operation **lift**(p) is performed, no vertex can be lifted more than $2n-2$ times and the claim follows.

(B) **The total number of saturating pushes is $O(nm)$:** Let us consider a fixed arc pq and find an upper bound for the number of saturating pushes along this arc in the algorithm. When we perform a saturating push along pq, we have $h(p) = h(q) + 1$ and the arc pq disappears from the residual network. It can only appear again in the current residual network after flow has been pushed from q to p in some later execution of the operation **push**(qp). At that time we have $h(q) = h(p) + 1$. This and the fact that h remains a height function and never decreases at any vertex, implies that before we can perform a new saturating push along pq, $h(p)$ has increased by at least two. We argued above that we always have $h(p) \leq 2n-1$ and now we conclude that there are at most $O(n)$ saturating pushes along any given arc. Thus the total number of saturating pushes is $O(nm)$.

(C) **The total number of unsaturating pushes is $O(n^2m)$:** Let $\Phi = \sum_{b_x(v)<0} h(v)$. Then $\Phi \geq 0$ during the whole execution of the algorithm and since $h(v) < 2n$ at any time during the execution we have $\Phi \leq 2n^2$ after the preprocessing step. Let us examine what happens to the value of Φ after performing the different kinds of operations. A lift will increase Φ by at most $2n-1$. Hence, by (A), the total contribution to Φ from lifts is $O(n^3)$. A saturating push from p to q can increase Φ by at most $h(q) \leq 2n-1$ (it may also decrease Φ if p becomes balanced, but we are not concerned about that here). Hence, by (B), the total contribution to

Φ from saturating pushes is $O(n^2m)$. An unsaturating push from p to q will decrease Φ by at least one, since p becomes balanced and $h(p) = h(q)+1$ (if q was balanced before, then Φ decreases by one and otherwise it decreases by $h(p)$).
It follows from the considerations above that the total increase in Φ during the execution of the algorithm is $O(n^2m)$. Now it follows from the fact that Φ is never negative that the total number of unsaturating pushes is $O(n^2m)$. □

It is somewhat surprising that the simple approach above results in an algorithm of such a low complexity. The complexity bound is valid no matter which vertex we choose to push from or lift. This indicates the power of the approach. However, the algorithm does have its drawbacks. If no control is supplied to direct the algorithm (as to which vertices to push from or lift), then a large amount of time may be spent without any effect on the final maximum flow. In Exercise 4.21 the reader is asked to give an example showing that a large amount of useless work may be performed if no extra guidance is given to the choice of pushes. There are several approaches which can improve the performance of the push-relabel algorithm, we mention just two of these. For details see, e.g., the book by Ahuja, Magnanti and Orlin [13].

(a) If we examine the active vertices in a first-in first-out (FIFO) order, then we obtain an $O(n^3)$ algorithm [416].
(b) If we always push from a vertex p which has the largest height $h(p)$ among all active vertices, then we obtain an $O(n^2\sqrt{m})$ algorithm [207, 416].

Cheriyan and Maheshwari [207] have shown by examples that the worst case bounds for the FIFO and maximum height variants are tight. For another way to improve the performance of the generic algorithm in practice, see Exercise 4.22.

4.7 Unit Capacity Networks and Simple Networks

In this section we consider two special cases of networks, both of which occur in applications and for which, due to their special structure, one can obtain faster algorithms for finding a maximum flow. All networks considered in this section are assumed to have a source s and a sink t.

4.7.1 Unit Capacity Networks

A **unit capacity network** is a network $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$, i.e., all arcs have capacity equal to one. Unit capacity networks are important in several applications of flows to problems such as finding a maximum matching

in a bipartite graph (Subsection 4.11.1), finding an optimal path cover of an acyclic digraph (Section 13.5) and finding cycle subdigraphs covering specified vertices (Subsection 13.8).

Lemma 4.7.1 *If $\mathcal{N}$ is a unit capacity network without cycles of length 2 and x is a feasible (s,t)-flow, then $\mathcal{N}(x)$ is also a unit capacity network.*

Proof: Exercise 4.39. □

Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a unit capacity network with source s and sink t. Since the value of a minimum (s,t)-cut in $\mathcal{N}$ is at most $n-1$ (consider the cut $(s, V-s)$), we see from Theorem 4.5.4 that the Ford-Fulkerson algorithm will find a maximum (s,t)-flow in time $O(nm)$. The purpose of this section is to show that one can obtain an even faster algorithm. Our exposition is based on an idea due to Even and Tarjan [308].

Lemma 4.7.2 *Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u \equiv 1)$ be a layered unit capacity network with $V_0 = \{s\}$ and $V_k = \{t\}$. One can find a maximal (s,t)-flow in $\mathcal{L}$ in time $O(m)$.*

Proof: It suffices to see that the capacity of each augmenting path is 1 and no two augmenting paths of the same length can use the same arc. Hence it follows that Dinic's algorithm will find a maximal flow in time $O(m)$. □

Lemma 4.7.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a unit capacity network and let x^* be a maximum (s,t)-flow in $\mathcal{N}$. Then*

$$\mathrm{dist}_{\mathcal{N}}(s,t) \leq 2n/\sqrt{|x^*|}. \tag{4.15}$$

Proof: Let $\omega = \mathrm{dist}_{\mathcal{N}}(s,t)$ and let $V_0 = \{s\}, V_1, V_2, \ldots, V_\omega$ be the first ω distance classes from s. Since $\mathcal{N}$ contains no multiple arcs, the number of arcs from V_i to V_{i+1} is at most $|V_i||V_{i+1}|$ for $i = 0, 1, \ldots, \omega - 1$. Since the arcs in (V_i, V_{i+1}) correspond to the arcs across an (s,t)-cut in $\mathcal{N}$, we have $|x^*| \leq |V_i||V_{i+1}|$ for $i = 0, 1, \ldots, \omega - 1$. Thus $\max\{|V_i|, |V_{i+1}|\} \geq \sqrt{|x^*|}$ for $i = 0, 1, \ldots, \omega - 1$. Now we easily see that

$$n = |V| \geq \sum_{i=0}^{\omega} |V_i| \geq \sqrt{|x^*|}\lfloor \frac{\omega+1}{2} \rfloor, \tag{4.16}$$

implying that $\omega \leq 2n/\sqrt{|x^*|}$. □

Theorem 4.7.4 *[308] For unit capacity networks the complexity of Dinic's algorithm is $O(n^{\frac{2}{3}}m)$.*

Proof: Let $\mathcal{N}$ be a unit capacity network with source s and sink t. We assume for simplicity that $\mathcal{N}$ has no 2-cycles. The case when $\mathcal{N}$ does have a 2-cycle can be handled similarly (Exercise 4.41). Let q be the number of

phases performed by Dinic's algorithm before a maximum (s,t)-flow is found in $\mathcal{N}$. Let $0 \equiv x^{(0)}, x^{(1)}, \ldots, x^{(q)}$ denote the (s,t)-flows in $\mathcal{N}$ which have been calculated after the successive phases of the algorithm. Thus $x^{(0)}$ is the starting flow which is the zero flow and $x^{(i)}$ denotes the flow after phase i of the algorithm. Let $\tau = \lceil n^{\frac{2}{3}} \rceil$ and let $K = |x^{(q)}|$ denote the value of a maximum (s,t)-flow in $\mathcal{N}$.

By Lemmas 4.7.1 and 4.7.2, it suffices to prove that the total number of phases, q, is $O(n^{\frac{2}{3}})$. This is clear in the case when $K \leq \tau$, since we augment the flow by at least one unit after each phase. So suppose that $K > \tau$. Choose j such that $|x^{(j)}| < K - \tau$ and $|x^{(j+1)}| \geq K - \tau$. By Theorems 4.4.2 and 4.4.3 the value of a maximum flow in $\mathcal{N}(x^{(j)})$ is $K - |x^{(j)}| > \tau$.

Applying Lemmas 4.7.1 and 4.7.3 to $\mathcal{N}(x^{(j)})$, we see that $\text{dist}_{\mathcal{N}(x^{(j)})}(s,t) \leq 2n^{\frac{2}{3}}$. Using Lemma 4.6.3 and the fact that each phase of Dinic's algorithm results in a maximal flow, we see that $j \leq 2n^{\frac{2}{3}}$. Thus, since at most τ phases remain after phase j we conclude that the total number of phases q is $O(n^{\frac{2}{3}})$. □

4.7.2 Simple Networks

A **simple network** is a network $\mathcal{N} = (V, A, l \equiv 0, u)$ with special vertices s, t in which every vertex in $V - \{s,t\}$ has precisely one arc entering or precisely one arc leaving. For an example see Figure 4.11.

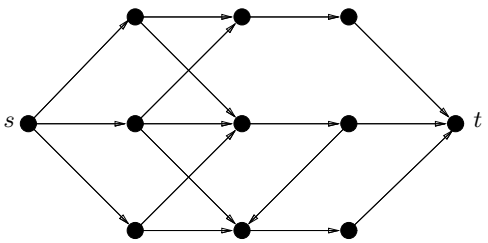


Figure 4.11 A simple network. Capacities are not shown.

Below we assume that the simple network in question does not have any 2-cycles. It is easy to see that this is not a serious restriction (Exercise 4.42).

Lemma 4.7.5 *Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a simple unit capacity network on n vertices and let x^* be a maximum (s,t)-flow in $\mathcal{N}$. Then*

$$\text{dist}_{\mathcal{N}}(s,t) \leq n/|x^*|. \tag{4.17}$$

Proof: Let $\omega = \text{dist}_{\mathcal{N}}(s,t)$ and $V_0 = \{s\}, V_1, V_2, \ldots, V_\omega$ be the first ω distance classes from s. Every unit of flow from s to t passes through the layer V_i for

$i = 1, 2, \ldots, \omega - 1$. Furthermore, since $\mathcal{N}$ is a simple unit capacity network, at most one unit of flow can pass through each $v \in V$. Thus $|V_i| \geq |x^*|$, for $i = 1, 2, \ldots, \omega - 1$, and hence

$$|V| > \sum_{i=1}^{\omega-1} |V_i| \geq (\omega - 1)|x^*|,$$

implying that $\omega \leq |V|/|x^*|$. □

Lemma 4.7.6 *If $\mathcal{N}$ is a simple unit capacity network, then $\mathcal{N}(x)$ is also a simple unit capacity network.*

Proof: Exercise 4.40. □

Using Lemmas 4.7.5 and 4.7.6 one can prove the following result due to Even and Tarjan. We leave the details as Exercise 4.43.

Theorem 4.7.7 *[308] For simple unit capacity networks Dinic's algorithm has complexity $O(\sqrt{n}m)$.* □

We point out that Dinic's algorithm will also find a maximum (s, t)-flow in time $O(\sqrt{n}m)$ in a simple network even if not all capacities are one, provided that the network has the property that at most one unit of flow can pass through any vertex $v \in V - \{s, t\}$. In particular a vertex may be the tail of an arc with capacity ∞ provided that it is the head of at most one arc and this arc (if it exists) has capacity one. We use this extension of Theorem 4.7.7 in Section 4.11.

4.8 Circulations and Feasible Flows

We now return to the general flow model and consider the problem of determining whether a feasible flow exists with respect to the given lower bounds and capacities on the arcs and a prescribed balance vector. As we showed in Section 4.2, in order to study the general case, it suffices to study circulations since we may use Lemmas 4.2.1-4.2.3 to transform the general case to the case of circulations. Note that in this section we always assume that all the data of the network are integers (that is, l and u are integers).

We need the following very simple observation. The proof is analogous to that of Lemma 4.5.1.

Lemma 4.8.1 *If x is a circulation in $\mathcal{N}$, then for every partition $S, \overline{S}$ of V we have $x(S, \overline{S}) = x(\overline{S}, S)$.* □

The example in Figure 4.12 gives us a starting point for detecting what can prevent the existence of a feasible circulation.

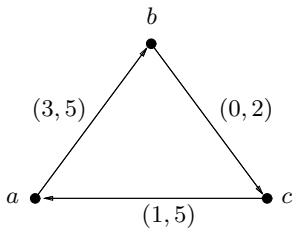


Figure 4.12 A network with no feasible circulation. The specification on the arcs is (l, u).

Let $\mathcal{N}$ be the network in Figure 4.12 and let $S = \{b\}$ and $\overline{S} = \{a, c\}$. Then $l(\overline{S}, S) = 3 > 2 = u(S, \overline{S})$. Now using Lemma 4.8.1 we see that if x is a feasible flow in $\mathcal{N}$, we must have

$$2 = u(S, \overline{S}) \geq x(S, \overline{S}) = x(\overline{S}, S) \geq l(\overline{S}, S) = 3,$$

implying that there is no feasible flow in $\mathcal{N}$. More generally, our argument shows that if $\mathcal{N} = (V, A, l, u)$ is a network for which some partition $S, \overline{S}$ of V satisfies $l(\overline{S}, S) > u(S, \overline{S})$, then $\mathcal{N}$ has no feasible circulation. Hoffman [531] proved that the converse holds as well.

Before we prove Theorem 4.8.2, we remark that Theorem 4.4.2 remains valid for networks with non-zero lower bounds provided that we modify the definition of $x \oplus \tilde{x}$ slightly (see Exercise 4.30).

Theorem 4.8.2 (Hoffman's circulation theorem) *[531] A network $\mathcal{N} = (V, A, l, u)$ with non-negative lower bounds on the arcs has a feasible circulation if and only if the following holds for every proper subset S of V:*

$$l(\overline{S}, S) \leq u(S, \overline{S}). \tag{4.18}$$

Proof: Let $\mathcal{N} = (V, A, l, u)$ be a network. We argued above that if x is a feasible circulation in $\mathcal{N}$, then for every partition $(S, \overline{S})$ of V we have $l(\overline{S}, S) \leq u(S, \overline{S})$.

To prove the converse we assume that (4.18) holds for all $S \subset V$ and give an algorithmic proof showing how to construct a feasible circulation starting from the all-zero circulation. Clearly $x \equiv 0$ is a circulation in $\mathcal{N}$ and if $l \equiv 0$, then we are done. So we may assume that $l_{ij} > x_{ij}$ for some $ij \in A$.

We try to find a (j, i)-path in $\mathcal{N}(x)$. If such a path P exists, then we let $\delta(P) > 0$ be the minimum residual capacity of an arc on P. Let $\epsilon = \min\{\delta(P), l_{ij} - x_{ij}\}$. By Theorem 4.4.2 (which, as remarked earlier, also holds when some lower bounds are non-zero), we can increase the current flow x by ϵ units along the cycle iP and obtain a new circulation.

We claim that we can continue this process until the current circulation x has $l_{ij} \leq x_{ij} \leq u_{ij}$ for all arcs $ij \in A$, that is, we can obtain a feasible circulation in $\mathcal{N}$ (observe that the procedure above preserves the inequality

$x \leq u$). Suppose this is not the case and that at some point we have $x_{st} < l_{st}$ for some arc st and there is no (t,s)-path in $\mathcal{N}(x)$. Define T as follows:

$$T = \{r : \text{ there exists a } (t,r)\text{-path in } \mathcal{N}(x)\}.$$

It follows from the definition of the residual network $\mathcal{N}(x)$ (in particular (4.7)) that in $\mathcal{N}$ we have $x_{ij} = u_{ij}$ for all arcs ij with $i \in T$ and $j \in \overline{T}$ and $x_{qr} \leq l_{qr}$ for all arcs qr with $q \in \overline{T}$ and $r \in T$. Using that $s \in \overline{T}$ and $x_{st} < l_{st}$ we obtain that

$$u(T,\overline{T}) = x(T,\overline{T}) = x(\overline{T},T) < l(\overline{T},T),$$

contradicting the assumption that (4.18) holds. This and the fact that all data are integers shows that the algorithm we described above will indeed find a feasible circulation in $\mathcal{N}$. □

It is not difficult to turn the proof above into a polynomial algorithm which, given a network $\mathcal{N} = (V,A,l,u)$, either finds a feasible circulation x in $\mathcal{N}$, or a subset S violating (4.18) (Exercise 4.29).

We conclude with a remark on finding feasible flows with respect to arbitrary balance vectors in general networks. This problem is relevant as a starting point for many algorithms on flows. It follows from the results in Section 4.2 and the fact that the push-relabel algorithm can be turned into an $O(n^3)$ algorithm (using the FIFO implementation) that the following holds.

Theorem 4.8.3 *There exists an $O(n^3)$ algorithm for finding a feasible flow in a given network $\mathcal{N} = (V,A,l,u,b)$. Furthermore, if l,u,b are all integer functions, then an integer feasible flow can be found in time $O(n^3)$.* □

Using Lemma 4.2.2 and Theorem 4.8.2 one can derive the following feasibility theorem for flows by Gale (Exercise 4.44):

Theorem 4.8.4 *[378] There exists a feasible flow in the network $\mathcal{N} = (V,A,l \equiv 0,u,b)$ if and only if*

$$\sum_{s \in S} b(s) \leq u(S,\overline{S}) \qquad \text{for every } S \subset U. \tag{4.19}$$

□

4.9 Minimum Value Feasible (s,t)-Flows

Let $\mathcal{N} = (V,A,l,u)$ be a network with source s, sink t and non-negative lower bounds on the arcs. A **minimum** feasible (s,t)-flow in $\mathcal{N}$ is a feasible (s,t)-flow whose value is minimum possible among all feasible (s,t)-flows. Although at first glance this problem may seem somewhat artificial, it turns

out that for many applications it is actually a minimum feasible flow that is sought (see, e.g., Sections 6.7 and 13.5).

To estimate the value of a minimum (s,t)-flow, let us define the **demand** $\gamma(S,\overline{S})$ of an (s,t)-cut $(S,\overline{S})$ as the number

$$\gamma(S,\overline{S}) = l(S,\overline{S}) - u(\overline{S},S). \tag{4.20}$$

Let x be a feasible flow. Then, by Lemma 4.5.1, for every (s,t)-cut $(S,\overline{S})$ we have

$$\begin{aligned} |x| &= x(S,\overline{S}) - x(\overline{S},S) \\ &\geq l(S,\overline{S}) - u(\overline{S},S) \\ &= \gamma(S,\overline{S}). \end{aligned} \tag{4.21}$$

Hence the demand of any (s,t)-cut provides a lower bound for the value of a minimum feasible (s,t)-flow. The next result shows that the minimum value of an (s,t)-flow is exactly the maximum demand of an (s,t)-cut.

Theorem 4.9.1 (Min-Flow Max-Demand theorem) *Suppose x is a minimum feasible (s,t)-flow in a network $\mathcal{N} = (V,A,l,u)$ with non-negative lower bounds on the arcs. Then*

$$|x| = \max\{\gamma(S,\overline{S}) : s \in S, t \in \overline{S}\}. \tag{4.22}$$

Furthermore we can find a minimum feasible (s,t)-flow by two applications of any algorithm for finding a maximum (s,t)-flow.

Proof: Suppose x is a feasible (s,t)-flow in $\mathcal{N}$. If $|x| = 0$, then x is clearly a minimum (s,t)-flow (since all lower bounds are non-negative). Hence we may assume that $|x| > 0$. Suppose that y is a feasible (t,s)-flow in $\mathcal{N}(x)$. By Theorem 4.4.2[9], $x \oplus y$ is a feasible flow in $\mathcal{N}$ of value $|x| - |y|$. Now suppose that y is a maximum (t,s)-flow in $\mathcal{N}(x)$. Apply Theorem 4.5.3 to y and $\mathcal{N}(x)$ and let $(T,\overline{T})$ be a minimum (t,s)-cut in $\mathcal{N}(x)$. The capacity of $(T,\overline{T})$ is by definition equal to $r(T,\overline{T})$, where r is the capacity function of $\mathcal{N}(x)$. By the choice of $(T,\overline{T})$ and the definition of the residual capacities we have

$$\begin{aligned} |y| &= r(T,\overline{T}) \\ &= \sum_{ij \in (T,\overline{T})} (u_{ij} - x_{ij}) + \sum_{qp \in (\overline{T},T)} (x_{qp} - l_{qp}) \\ &= u(T,\overline{T}) - l(\overline{T},T) + x(\overline{T},T) - x(T,\overline{T}) \\ &= u(T,\overline{T}) - l(\overline{T},T) + |x|, \end{aligned} \tag{4.23}$$

[9] As we remarked in the last section, this theorem is also valid in the general case of non-zero lower bounds.

by Lemma 4.5.1. Rearranging this, we obtain that $|x|-|y| = l(\overline{T}, T)-u(T, \overline{T})$. This implies that the flow $x \oplus y$ (whose value is $|x|-|y|$) is a minimum feasible (s,t)-flow and proves (4.22).

It remains to prove the second claim on how to find a minimum (s,t)-flow. It follows from the argument above that once we have any feasible (s,t)-flow, we can find a minimum (s,t)-flow by just one max flow calculation. On the other hand it follows from Lemmas 4.2.1 and 4.2.2 that we can find a feasible (s,t)-flow in $\mathcal{N}$ (if any exists) by performing the two transformations suggested in those lemmas and then using a max flow algorithm to check whether there is a feasible flow in the last network constructed (now feasibility is with respect to the value of $b(s)$ and all lower bounds are zero). □

4.10 Minimum Cost Flows

We now turn to networks with costs on the arcs and study the following problem called the MINIMUM COST FLOW PROBLEM: Given a network $\mathcal{N} = (V, A, l, u, b, c)$ find a feasible flow of minimum cost[10]. By the results in Section 4.2, without loss of generality, we may treat the problem only for networks with lower bound zero on all arcs and furthermore assume that we are looking for either an (s,t)-flow of value $b(s)$ or a circulation of minimum cost. However, for different applications, different flow models may be more convenient than others. Hence, except for always assuming that the lower bounds are zero, we will treat the general case, and hence all the special cases also, below.

We mentioned in Section 4.2 that the shortest path problem is a special case of the minimum cost flow problem. To see this, let $D = (V, A, c)$ be an arc weighted digraph with special vertices s, t and assume that D has no cycle of negative weight. Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1, c)$ be the network obtained from D by adding a lower bound of zero and a capacity of 1 to each arc of D and interpreting the weight of an arc in D as its cost in $\mathcal{N}$. We claim that a shortest (s,t)-path in D corresponds to a minimum cost integer (s,t)-flow of value 1 in $\mathcal{N}$. Clearly, any (s,t)-path P of weight M in D can be transformed into an (s,t)-flow of cost M just by sending one unit of flow along P in $\mathcal{N}$. Thus it suffices to prove that every (s,t)-flow x of value 1 and cost M can be transformed into an (s,t)-path in D of weight at most M. By Theorem 4.3.1, we may decompose x into a path flow of value 1 along an (s,t)-path P' and a number of cycle flows. All these cycles have non-negative cost since D has no negative cycle. Hence it follows that P' has cost at most M. It follows from our observations above that every minimum cost (s,t)-flow of value 1 in $\mathcal{N}$ can be decomposed into an (s,t)-path of the same cost and some cycle flows along cycles of cost zero.

[10] Recall that the cost of a flow is given by $\sum_{ij \in A} x_{ij} c_{ij}$.

In Exercise 4.47 the reader is asked to show that the maximum flow problem is also a special case of the minimum cost flow problem. However, the minimum cost flow problem is interesting not only because it generalizes these two problems, but also because a large number of practical applications can be formulated as minimum cost flow problems. The very comprehensive book by Ahuja, Magnanti and Orlin [13] contains a large number of such applications. We will discuss one of these in a reformulated form below.

A small cargo company uses a ship with a capacity to carry at most r units of cargo. The ship sails on a long route (say from Southampton to Alexandria) with several stops at ports in between. At these ports cargo may be unloaded and new cargo loaded. At each port there is an amount b_{ij} of cargo which is waiting to be shipped from port i to port $j > i$ (ports are numbered after the order in which the ship visits them). Let f_{ij} denote the income for the company from transporting one unit of cargo from port i to port j. The goal for the cargo company is to plan how much cargo to load at each port so as to maximize the total income while never exceeding the capacity of the ship. We illustrate how to model this problem, which we call the SHIP LOADING PROBLEM, as a minimum cost flow problem. The motivation for describing this application is that it shows not only that sometimes it is easier to work with the general model, but also that allowing negative costs on the arcs may simplify the formulation.

Let n be the number of stops including the starting port and the terminal port. Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be the network defined as follows:

$$V = \{v_1, v_2, \ldots, v_n\} \cup \{v_{ij} : 1 \leq i < j \leq n\},$$

$$A = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n\} \cup \{v_{ij}v_i, v_{ij}v_j : 1 \leq i < j \leq n\}.$$

The capacity of the arc v_iv_{i+1} is r for $i = 1, 2, \ldots, n-1$ and all other arcs have capacity ∞. The cost of the arc $v_{ij}v_i$ is $-f_{ij}$ for $1 \leq i < j \leq n$. All other arcs have cost zero (including those of the form $v_{ij}v_j$). The balance vector of v_{ij} is b_{ij} for $1 \leq i < j \leq n$ and the balance vector of v_i is $-(b_{1i} + b_{2i} + \ldots + b_{i-1i})$ for $i = 1, 2, \ldots, n$. (See Figure 4.13.)

We claim that this network models the ship loading problem. Indeed, suppose that $t_{12}, t_{13}, \ldots, t_{1n}, t_{23}, \ldots, t_{n-1n}$ are cargo numbers, where $t_{ij} (\leq b_{ij})$ denotes the amount of cargo the ship will transport from port i to port j and that the ship is never loaded above capacity. The total income from these cargo loads is $I = \sum_{1 \leq i < j \leq n} t_{ij} f_{ij}$. Let x be the flow in $\mathcal{N}$ defined as follows. The flow on an arc of the form $v_{ij}v_i$ is t_{ij}, the flow on an arc of the form $v_{ij}v_j$ is $b_{ij} - t_{ij}$ and the flow on an arc of the form v_iv_{i+1}, $i = 1, 2, \ldots, n-1$, is the sum of those t_{ab} for which $a \leq i$ and $b \geq i + 1$. It follows from the fact that t_{ij}, $1 \leq i < j \leq n$, are legal cargo numbers that x is feasible with respect to the balance vector and the capacity restriction. It is also easy to see that the cost of x is $-I$.

Conversely, suppose that x is a feasible flow in $\mathcal{N}$ of cost J. We claim that we get a feasible cargo assignment s_{ij}, $1 \leq i < j \leq n$ with income $-J$

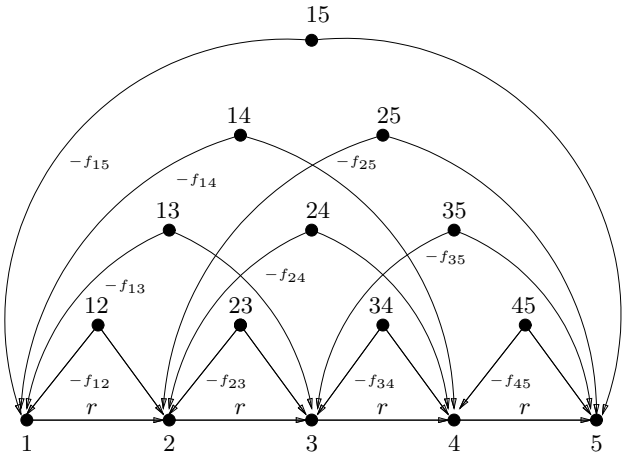


Figure 4.13 The network for the ship loading problem with three intermediate stops. For readability vertices are named by numbers only. The costs (capacities) are only shown when non-zero (not infinite). The balance vectors are as specified in the description in the text, i.e., the balance vector of the vertex 34 is b_{34} and the balance vector of the vertex 4 is $-(b_{14} + b_{24} + b_{34})$.

by letting s_{ij} be the value of x on the arc $v_{ij}v_i$. This is easy to check and we leave the details to the reader. It follows that a minimum cost flow in $\mathcal{N}$ corresponds to an optimal loading of the ship and vice versa.

Below we consider the minimum cost flow problem in some detail. Further applications are given in Section 4.11. See also Section 4.10.3 for two important special cases of the minimum cost flow problem.

We use the notion of the **cost** of a path or a cycle in a network. This is simply the sum of the costs of all arcs in the path or cycle. An **augmenting** path (cycle) with respect to a given flow x in a network $\mathcal{N}$ is a path (cycle) in $\mathcal{N}(x)$. Whenever we speak about an augmenting cycle or path W we use $\delta(W)$ to denote the minimum residual capacity of an arc on W in $\mathcal{N}(x)$. Furthermore, for every $\beta \leq \delta(W)$ we denote by $x' := x \oplus \beta W$ the flow we obtain from x by augmenting along W with β units.

Whenever we say that a flow x is **optimal** in a network $\mathcal{N}$, we mean by this that x is a minimum cost flow among all flows in $\mathcal{N}$ with balance vector b_x.

4.10.1 Characterizing Minimum Cost Flows

Recall from Theorem 4.5.3 that, when we consider maximum (s,t)-flows, we can verify optimality by showing that there is no (s,t)-path in the residual network with respect to the current flow. It turns out that we can also use the residual network to check whether a given feasible flow in a network $\mathcal{N} = (V, A, l, u, c)$ has minimum cost among all flows with the same balance vector.

Suppose first that x is feasible in $\mathcal{N}$ and that there is some cycle W in $\mathcal{N}(x)$ such that the cost $c(W)$ of W is negative. Let δ denote the minimum residual capacity of an arc on W and let x' be the cycle flow in $\mathcal{N}(x)$ which sends δ units around W. Then it follows from Theorem 4.4.2 that $x \oplus x'$ is a flow in $\mathcal{N}$ with the same balance vector as x and cost $c^T x + c^T x' = c^T x + \delta c(W) < c^T x$. Thus if $\mathcal{N}(x)$ contains a cycle of negative cost, then x is not a minimum cost feasible flow in $\mathcal{N}$ with respect to the balance vector b_x.

The interesting thing is that the other direction holds as well. Indeed, suppose x is feasible in $\mathcal{N} = (V, A, l, u, b, c)$ and that $\mathcal{N}(x)$ contains no cycle of negative cost. Let y be an arbitrary feasible flow in $\mathcal{N}$. Since we have specified a balance vector b for $\mathcal{N}$, it follows from Corollary 4.4.4 that there exists a collection of at most m cycles $W_1, W_2, \ldots, W_k$ in $\mathcal{N}(x)$ and cycle flows $f(W_1), \ldots, f(W_k)$ in $\mathcal{N}(x)$ such that $c^T y = c^T x + \sum_{i=1}^{k} c(W_i)\delta_i$, where $\delta_i > 0$ is the amount of flow that $f(W_i)$ sends along W_i in $\mathcal{N}(x)$. Since $\mathcal{N}(x)$ has no negative cost cycle, $c(W_i) \geq 0$ for $i = 1, 2, \ldots, k$ and we see that[11] $c^T y \geq c^T x$. Thus we have established the following important optimality criterion for the minimum cost flow problem.

Theorem 4.10.1 *Let x be a feasible flow in the network $\mathcal{N} = (V, A, l, u, b, c)$. Then x is a minimum cost feasible flow in $\mathcal{N}$ if and only if $\mathcal{N}(x)$ contains no directed cycle of negative cost.* □

It is natural to ask how useful this optimality criterion is. First observe that, using the Bellman-Ford-Moore algorithm (Subsection 3.3.4), we can check whether an arbitrary given network contains a negative cycle in time $O(nm)$. Thus we obtain the following algorithm, due to Klein [598], for finding a minimum cost feasible flow in a network.

The cycle canceling algorithm
Input: A network $\mathcal{N} = (V, A, l, u, b, c)$.
Output: A minimum cost feasible flow in $\mathcal{N}$.

1. Find a feasible flow x in $\mathcal{N}$.
2. Search for a negative cycle in $\mathcal{N}(x)$.
3. If such a cycle W is found, then augment x by $\delta(W)$ units along W and go to Step 2.
4. Return x.

Just as is the case for the Ford-Fulkerson algorithm, the cycle canceling algorithm may never terminate if the capacities are non-rational numbers. It is easy to modify the example in Exercise 4.17 to show this. However, if all lower bounds and capacities are integers (or just rational numbers), then this is indeed an algorithm, although not always a very fast one. See Figure 4.14 for an illustration of the algorithm.

[11] In fact, our argument shows that $c^T y = c^T x$ if and only if y can be obtained from x by 'adding' zero or more cycle flows, each of cost zero, in $\mathcal{N}(x)$.

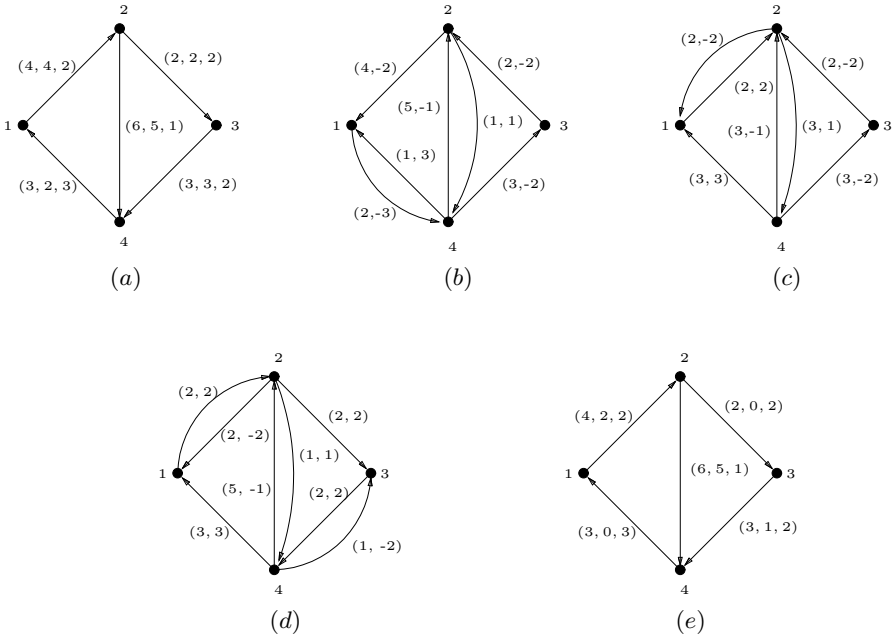


Figure 4.14 An illustration of the cycle canceling algorithm. (a) A network $\mathcal{N}$ with a feasible flow x with respect to the balance vector $(b(1), b(2), b(3), b(4)) = (2, 3, 1, -6)$. The data on the arcs are (capacity, flow, cost). (b) The residual network $\mathcal{N}(x)$. The data on the arcs are (residual capacity, cost). (c) The residual network after augmenting by 2 units along the cycle 1421. (d) The residual network after augmenting by 2 units along the cycle 2432. (e) The final optimal flow.

Let U and C denote the maximum capacity of $\mathcal{N}$ and the maximum numerical value among all costs of $\mathcal{N}$.

Theorem 4.10.2 *If all lower bounds, capacities, costs and balance vectors of the input network $\mathcal{N}$ are integers, then the cycle canceling algorithm finds an optimum flow in time $O(nm^2CU)$.*

Proof: By Theorem 4.8.3, we can find a feasible flow x in $\mathcal{N}$ in time $O(n^3)$. Hence Step 1 can be performed within the promised time bound, since we assume that all networks in this chapter have $m = \Omega(n)$. The maximum possible cost of a feasible flow in $\mathcal{N}$ is mUC and the minimum possible cost is $-mUC$. Since we decrease the cost of the current flow by at least one in Step 3, it follows that after at most $O(mUC)$ executions of Step 3 we obtain a minimum cost feasible flow. Now the complexity follows from the fact that Step 2 can be performed in time $O(nm)$, using the Bellman-Ford-Moore algorithm. □

Furthermore, just as it was the case for maximum flows, we have a nice integrality property.

Theorem 4.10.3 (Integrality theorem for minimum cost flows) *If all lower bounds, capacities and balance vectors of the network $\mathcal{N}$ are integers, then there exists an integer minimum cost flow.*

Proof: This is an easy consequence of the proof of Theorem 4.10.2. By Theorem 4.8.3, we may assume that the flow x after Step 1 is an integer flow. Now the claim follows easily by induction of the number of augmentations made by the cycle canceling algorithm, since in each augmentation we change the current flow by an integer amount along the arcs of the augmenting cycle. □

For arbitrary networks with integer-valued data, the complexity of the cycle canceling algorithm is not very impressive and the algorithm is clearly not polynomial since its running time is exponential in both the maximum capacity and the maximum (absolute value of the) cost. It is easy to construct examples for which the algorithm, without some guidance as to how the next negative cycle should be chosen, may use $O(mUC)$ augmentations before it arrives at an optimum flow (Exercise 4.52). However, for several applications, such as when we are looking for certain structures in digraphs, both U and C are small and then the algorithm is quite attractive due to its simplicity (see, e.g., some of the results in Section 4.11).

The problem of finding a strongly polynomial algorithm[12] for the minimum cost flow problem was posed by Edmonds and Karp [288] in 1972 and remained open until Tardos [842] found the first such algorithm in 1985. We mentioned above that if we use just any negative cycle in Step 3, then the cycle canceling algorithm may use a non-polynomial number of iterations. Goldberg and Tarjan showed that the following variant of the algorithm is already strongly polynomial [417]. The **mean cost** of a cycle W is the number $c(W)/|A(W)|$.

Theorem 4.10.4 *[417] If we always augment along a cycle of minimum mean cost (as negative mean cost as possible) in Step 3, then the cycle canceling algorithm has complexity $O(n^2m^3 \log n)$, even if some arcs have non-rational data.* □

The correctness of the algorithm, provided that it terminates, follows from Theorem 4.10.1, since there is no negative cycle in the current residual network at termination. Due to space considerations, we will not prove the complexity part of the theorem here. We refer the interested reader to [13, 710] for nice accounts for the complexity of this algorithm. It is interesting to

[12] A graph algorithm is **strongly polynomial** if (counting each arithmetic operation as constant time) the number of operations is bounded by a polynomial in n and m.

note that although the proof of the complexity statement of Theorem 4.10.4 is quite non-trivial, it uses just the basic definitions of flows along with some new concepts which are used to make the proof smoother.

4.10.2 Building up an Optimal Solution

The cycle canceling algorithm starts from a (generally) non-optimal but feasible flow and continues through a sequence of feasible flows until an optimal flow is found (provided the algorithm ever terminates). In this subsection we describe another approach, due to Jewell [565] and Busacker and Gowen [184], in which we start from a (generally) infeasible flow which is optimal[13] and continue through a sequence of optimal but infeasible flows until a feasible and optimal flow is reached.

Theorem 4.10.5 (The buildup theorem) *[565, 184] Suppose that x is a minimum cost feasible flow in a network $\mathcal{N} = (V, A, l \equiv 0, u, c)$ with respect to the balance vector $b = b_x$ and let P be a minimum cost (p, q)-path in $\mathcal{N}(x)$. Let $\alpha \leq \delta(P)$ and let $f(P)$ be the path flow of value α in $\mathcal{N}(x)$. Then the flow $x' := x \oplus f(P)$ is a minimum cost feasible flow in $\mathcal{N}$ with respect to the balance vector b' given by*

$$b'(v) = \begin{cases} b(v) & \text{if } v \notin \{p, q\} \\ b(p) + \alpha & \text{if } v = p \\ b(q) - \alpha & \text{if } v = q. \end{cases}$$

Proof: By Theorem 4.10.1, it is sufficient to prove that there is no negative cycle in $\mathcal{N}(x')$. Since x is optimal, there is no negative cycle in $\mathcal{N}(x)$. Suppose that $\mathcal{N}(x')$ contains a negative cycle W. By the definition of x', every arc in $\mathcal{N}(x')$ is either an arc of $\mathcal{N}(x)$ or the opposite of an arc on P. Consider the directed multigraph H that we obtain from $A(P) \cup A(W)$, considered as a multiset, by deleting all arcs a such that both a and the opposite arc are in $A(P) \cup A(W)$. It is easy to see that if we add the arc qp to H, then we obtain a directed multigraph M such that each connected component of M is eulerian. Hence, by Exercise 4.8, we can decompose $A(H)$ into a (p, q)-path P' and a number of cycles $W_1, W_2, \ldots, W_k$. It follows from our remark above and the way we defined H that all arcs of $P', W_1, W_2, \ldots, W_k$ are arcs of $\mathcal{N}(x)$. By (4.2) opposite arcs have costs which cancel and hence, using that $c(W) < 0$, we obtain

$$\begin{aligned} c(P) &> c(P) + c(W) \\ &= c(P') + \sum_{i=1}^{k} c(W_i) \\ &\geq c(P'), \end{aligned}$$

[13] Recall that optimality is with respect to flows with the same balance vector.

since the cost of each W_i must be non-negative because W_i is a cycle in $\mathcal{N}(x)$. Thus we see that P' is a (p,q)-path with a cost smaller than that of P, contradicting the minimality of P. Hence W cannot exist and the proof is complete. □

Based on Theorem 4.10.5 we can construct an algorithm, called the **buildup algorithm** [565, 184], for finding an optimal feasible flow in a network $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$. The algorithm described below only works if there are no negative cycles in the starting network. This restriction poses no practical problems since, according to Exercise 4.49, we may reduce the general minimum cost flow problem to the case when all costs are non-negative. Under the assumption that $\mathcal{N}$ has no negative cycles, the flow $x \equiv 0$ is an optimal circulation in $\mathcal{N}$. At any time during the execution of the buildup algorithm the sets U_x, Z_x are defined with respect to the current flow x as follows:

$$U_x = \{v | b_x(v) < b(v)\}, \qquad Z_x = \{v | b_x(v) > b(v)\}.$$

Observe that $U_x = \emptyset$ if and only if $Z_x = \emptyset$.

The buildup algorithm
Input: A network $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$.
Output: A minimum cost feasible flow in $\mathcal{N}$ with respect to b or a proof that the problem is infeasible.

1. Let $x_{ij} := 0$ for every $ij \in A$;
2. If $U_x = \emptyset$, then go to Step 8;
3. If there is no (U_x, Z_x)-path in $\mathcal{N}(x)$, go to Step 9;
4. Let p and q be chosen such that $p \in U_x, q \in Z_x$ and $\mathcal{N}(x)$ contains a (p,q)-path;
5. Find a minimum cost (p,q)-path P in $\mathcal{N}(x)$;
6. Let $\epsilon = \min\{\delta(P), b(p) - b_x(p), b_x(q) - b(q)\}$ ($\delta(P)$ is the residual capacity of P);
7. Let $x := x \oplus \epsilon P$; modify U_x, Z_x and go to Step 2;
8. Return x;
9. Return 'no feasible solution'.

See Figure 4.15 for an illustration of the algorithm.

Theorem 4.10.6 *[565, 184] Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ have all data integers and no negative costs. The buildup algorithm correctly determines a minimum cost feasible flow x in $\mathcal{N}$ or detects that no feasible flow exists in $\mathcal{N}$. The algorithm can be performed in time $O(n^2 mM)$, where $M = \max_{v \in V} |b(v)|$. Furthermore, if there is a feasible flow in $\mathcal{N}$, then the algorithm will find an integer optimal feasible flow in $\mathcal{N}$.*

Proof: Exercise 4.50. □

The following result shows that when we consider minimum cost (s,t)-flows, the cost of successive augmenting (s,t)-paths forms a monotonically

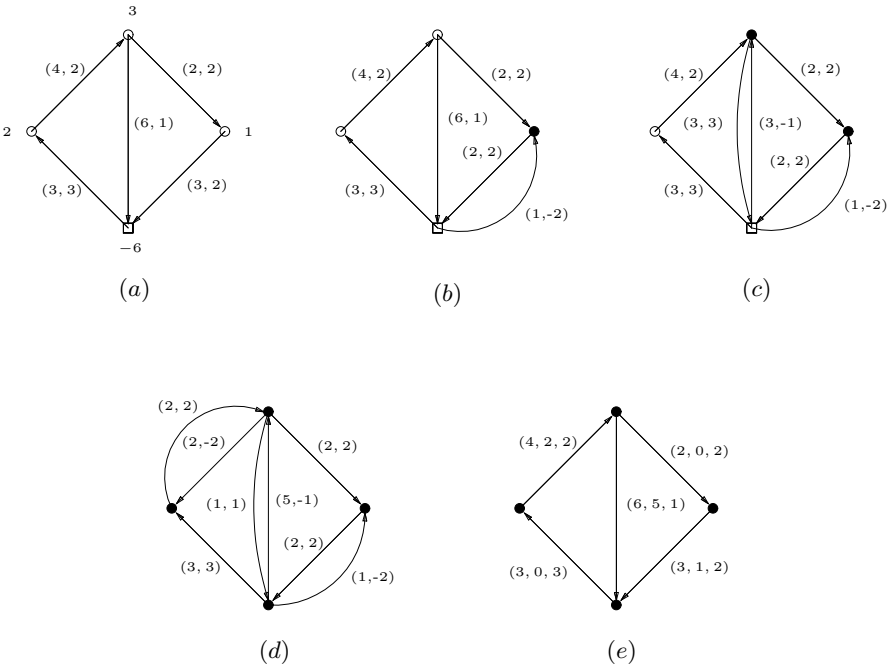


Figure 4.15 The buildup algorithm performed on the network from Figure 4.14(a). Parts (a)-(d) show the current residual network with respect to the flow x, starting from $x \equiv 0$ in (a). For each arc (u, c) is specified and in (a) $b(v)$ is specified for each vertex. White circles correspond to the set U_x and white boxes correspond to Z_x. Black circles represent vertices that have reached the desired balance value. Part (e) shows the final optimal flow.

increasing function. One can make a more general statement (Exercise 4.51), but for simplicity we consider only (s, t)-flows here.

Proposition 4.10.7 *Let $\mathcal{N}$ be a network with distinct vertices s, t and let x be an optimal (s, t)-flow in $\mathcal{N}$. Suppose x' is obtained from x by augmenting by one unit along a minimum cost (s, t)-path P in $\mathcal{N}(x)$ and that x'' is obtained from x' by augmenting by one unit along a minimum cost (s, t)-path P' in $\mathcal{N}(x')$. Then*

$$c^T x - c^T x' \geq c^T x' - c^T x''. \tag{4.24}$$

Proof: Let x, x', x'' and P, P' be as described in the proposition. Analogously to the way we argued in the proof of Theorem 4.10.5, we can show that the directed multigraph H' obtained from the multiset of arcs from $A(P) \cup A(P')$ by deleting arcs that are opposite in the two paths can be decomposed into two (s, t)-paths Q, R and some cycles $W_1, \ldots, W_p$ such that all arcs of these paths and cycles are in $\mathcal{N}(x)$. Since x is optimal each cycle W_i, $i = 1, 2, \ldots, p$,

has non-negative cost by Theorem 4.10.1. Using that P is a minimum cost (s,t)-path in $\mathcal{N}(x)$, we conclude that each of R, Q has cost at least $c(P)$ implying that $c(P') \geq c(P)$. Hence (4.24) holds. □

4.10.3 The Assignment and the Transportation Problem

In this section we briefly discuss two special cases of the minimum cost flow problem, both of which occur frequently in practical applications. For a more detailed discussion see, e.g., [91, Section 3.12].

In the ASSIGNMENT PROBLEM, the input consists of a set of persons $P_1, P_2, \ldots, P_n$, a set of jobs $J_1, J_2, \ldots, J_n$ and an $n \times n$ matrix $M = [M_{ij}]$ whose entries are non-negative integers. Here M_{ij} is a measure for the skill of person P_i in performing job J_j (the lower the number the better P_i performs job J_j). The goal is to find an assignment π of persons to jobs so that each person gets exactly one job and the sum $\sum_{i=1}^{n} M_{i\pi(i)}$ is minimized. Given any instance of the assignment problem, we may form a complete bipartite graph $B = (U, V; E)$ where $U = \{P_1, P_2, \ldots, P_n\}$, $V = \{J_1, J_2, \ldots, J_n\}$ and E contains the edge P_iJ_j with the weight M_{ij} for each $i \in [m]$, $j \in [n]$. Now the assignment problem is equivalent to finding a minimum weight perfect matching in B. Clearly we can also go the other way and hence the assignment problem is equivalent to the WEIGHTED BIPARTITE MATCHING PROBLEM. It is also easy to see from this observation that the assignment problem is a (very) special case of the minimum cost flow problem. In fact, if we think of M_{ij} as a cost and orient all edges from U to V in the bipartite graph above, then what we are seeking is an integer-valued flow of minimum cost so that the value of the balance vector equals 1 for each P_i, $i = 1, 2, \ldots, m$, and equals -1 for each J_j, $j = 1, 2, \ldots, n$.

Inspecting the description of the buildup algorithm above, it is not hard to see that the following holds (Exercise 4.53).

Theorem 4.10.8 *The buildup algorithm solves the assignment problem for a bipartite graph on n vertices in time $\mathcal{O}(n^3)$.* □

In the TRANSPORTATION PROBLEM we are given a set of production plants $S_1, S_2, \ldots, S_m$ that produce a certain product to be shipped to a set of retailers $T_1, T_2, \ldots, T_n$. For each pair (S_i, T_j) there is a real-valued cost c_{ij} of transporting one unit of the product from S_i to T_j. Each plant produces a_i, $i = 1, 2, \ldots, m$, units per time unit and each retailer needs b_j, $j = 1, 2, \ldots, n$, units of the product per time unit. We assume below that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ (this is no restriction of the model as shown in Exercise 4.54). The goal is to find a transportation schedule for the whole production (i.e., how many units to send from S_i to T_j for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$) in order to minimize the total transportation cost.

Again the transportation problem is easily seen to be a special case of the minimum cost flow problem. Consider a bipartite network $\mathcal{N}$ with partite sets

$S = \{S_1, S_2, \ldots, S_m\}$ and $T = \{T_1, T_2, \ldots, T_n\}$ and all possible arcs from S to T, where the capacity of the arc S_iT_j is ∞ and the cost of sending one unit of flow along S_iT_j is c_{ij}. Now it is easy to see that an optimal transportation schedule corresponds to a minimum cost flow in $\mathcal{N}$ with respect to the balance vectors

$$b(S_i) = a_i, i = 1, 2, \ldots, m, \text{ and } b(T_j) = -b_j, j = 1, 2, \ldots, n.$$

Again we could solve the transportation problem by the buildup algorithm but in this case we would not be guaranteed a polynomial running time since the running time would depend on the required balance values. Applying Theorem 4.10.4, we obtain a strongly polynomial algorithm for the problem. Clearly one would expect the existence of an algorithm of better complexity for the transportation problem (being a restricted version of the general minimum cost flow problem). Such an algorithm was given by Kleinschmidt and Schannath.

Theorem 4.10.9 *[600] The transportation problem with m suppliers and n consumers can be solved in time $\mathcal{O}(\min\{n, m\}(n + m)^2 \log(n + m))$.* □

For much more material on the assignment and transportation problems, including a survey of various complexities, the reader may consult Chapters 17 and 21 of Schrijver's book [803].

4.11 Applications of Flows

In this section we illustrate the applicability of flows to a large spectrum of problems both of theoretical and practical nature. For further applications see, e.g., Chapters 5, 13 and 17. Since we will need these results in later chapters the main focus is on finding certain substructures in digraphs.

4.11.1 Maximum Matchings in Bipartite Graphs

Let $G = (V, E)$ be an undirected graph. Recall that a matching in G is a set of edges from E, no two of which share a vertex, and a **maximum matching** of G is a matching of maximum cardinality among all matchings of G. Matching problems occur in many practical applications such as the following scheduling problem. We are given a set $T = \{t_1, t_2, \ldots, t_r\}$ of tasks (such as handling a certain machine) to be performed and a set $P = \{p_1, p_2, \ldots, p_s\}$ of persons, each of which is capable of performing some of the tasks from T. The goal is to find a maximum number of tasks such that each task can be performed by some person who does not at the same time perform any other task and no task is performed by more than one person. This can be formulated as a matching problem as follows. Let $B = (P, T; E)$ be the bipartite graph whose vertex set is $P \cup T$ and such that for each i, j such that

$1 \leq i \leq s$, $1 \leq j \leq r$, E contains the edge $p_i t_j$ whenever person p_i can perform task t_j. Now it is easy to see that the answer to the problem above is a matching in B which covers the maximum possible number of vertices in T (see also Exercise 4.55). Finding a maximum matching in an arbitrary (not necessarily bipartite) graph is quite complicated and it was a great breakthrough when Edmonds [282] found a polynomial algorithm. For the case of bipartite graphs we describe a simple algorithm based on flows.

Theorem 4.11.1 *For bipartite graphs the maximum matching problem is solvable in time* $O(\sqrt{n}m)$.

Proof: Let $B = (X, Y; E)$ be an undirected bipartite graph with bipartition (X, Y). Construct a network $\mathcal{N}_B = (X \cup Y \cup \{s, t\}, A, l \equiv 0, u)$ as follows (see Figure 4.16):

$A = \{ij : i \in X, j \in Y \text{ and } ij \in E\} \cup \{si : i \in X\} \cup \{jt : j \in Y\}$, $u_{ij} = \infty$ for all $ij \in (X, Y)$, $u_{si} = 1$ for all $i \in X$ and $u_{jt} = 1$ for all $j \in Y$.

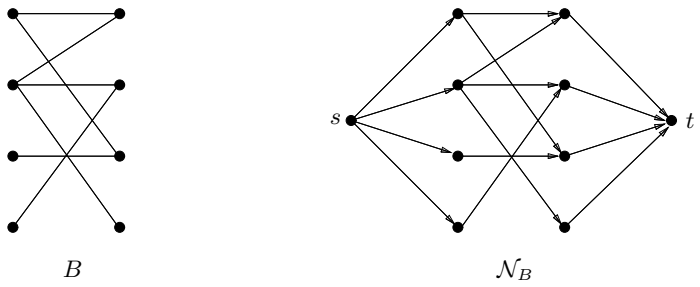


Figure 4.16 A bipartite graph and the corresponding network. Capacities are one on all arcs of the form sv, ut and ∞ on all arcs corresponding to edges of B.

We claim that the value of a maximum (s, t)-flow in $\mathcal{N}_B$ equals the size of a maximum matching in B. To see this, suppose that x is an integer flow in $\mathcal{N}$ of value k. Let $M = \{ij : i \in X, j \in Y \text{ and } x_{ij} > 0\}$. For each $i \in X$ the flow on the arc x_{si} is either zero or one. Furthermore, if $x_{si} = 1$, then it follows from the fact that x is integer valued and $b_x(i) = 0$ that precisely one arc from i to Y has non-zero flow. Similarly , for each $j \in Y$, if $x_{jt} = 1$, then precisely one arc from X to j has non-zero flow. It follows that M is a matching of size k in B and hence, by Theorem 4.5.5, the size of a maximum matching in B is at least the value of a maximum flow in $\mathcal{N}_B$.

On the other hand, if $M' = \{q_i r_i : q_i \in X, r_i \in Y, i = 1, 2, \ldots, h\}$ is a matching in B, then we obtain a feasible (s, t)-flow of value h in $\mathcal{N}_B$ by sending one unit of flow along each of the internally disjoint paths $sq_i r_i t$, $i = 1, 2, \ldots, h$. This shows that the opposite inequality also holds and the claim follows.

It follows from the arguments above that, given a maximum integer flow x, we can obtain a maximum matching M of B by taking precisely those arcs of the form u_iv_i, $u_i \in X, v_i \in Y$, which have flow value equal to 1. Note that $\mathcal{N}_B$ is a simple network. Hence the complexity claim follows from the fact that we can find a maximum flow in $\mathcal{N}$ in time $O(\sqrt{n}m)$, using the algorithm of Theorem 4.7.7 (recall that this complexity is also valid for simple networks where not all capacities are 1, provided that at most one unit of flow can pass through any vertex distinct from s, t). □

In the case of dense graphs a slightly faster algorithm of complexity $O(n^{1.5}\sqrt{m/\log n})$ was given by Alt, Blum, Mehlhorn and Paul in [39]. It is still possible to obtain fast algorithms for finding a maximum matching in general graphs, see, e.g., Tarjan's book [845]. However, it does not seem possible to formulate the maximum matching problem for an arbitrary graph as an instance of the maximum flow problem in some network. In [604] a generalization of flows which contains the maximum matching problem for general graphs as a special case was studied by Kocay and Stone.

A **vertex cover** of an undirected graph $G = (V, E)$ is a subset $U \subseteq V$ such that every edge $e \in E$ has at least one of its end-vertices in U. Since no two edges of a matching share a vertex, it follows that for every vertex cover U in G, the size of U is at least the size of a maximum matching. For general graphs there does not have to be equality between the size of a maximum matching and the size of a minimum vertex cover. For instance, if G is just a 5-cycle, then the size of a maximum matching is 2 and no vertex cover has less than 3 vertices. We now prove the following result, due to König [619], which shows that for bipartite graphs equality does hold. The proof illustrates the power of the max-flow min-cut theorem.

Theorem 4.11.2 (König's theorem) *[619] Let $B = (X, Y; E)$ be an undirected bipartite graph with bipartition (X, Y). The size of a maximum matching in B equals the size of a minimum vertex cover in B.*

Proof: Let $\mathcal{N}_B = (V \cup \{s, t\}, A, l \equiv 0, u)$ be defined as in the proof of Theorem 4.11.1. Let x be a maximum flow in $\mathcal{N}_B$ and let $(S, \overline{S})$ be the minimum cut defined with respect to x, as in the proof of Theorem 4.5.3 (see Figure 4.17). Recall that S is precisely those vertices of $V \cup \{s, t\}$ which can be reached from s in $\mathcal{N}_B(x)$. Since the capacity of each arc from X to Y is ∞, there is no edge from $S \cap X$ to $\overline{S} \cap Y$ in G. Thus $U = (X \cap \overline{S}) \cup (Y \cap S)$ is a vertex cover in B. Furthermore, it follows from the definition of S that we must have $x_{si} = 1$ for all $i \in X \cap \overline{S}$ and $x_{jt} = 1$ for all $j \in Y \cap S$. This shows that $|x| = |X \cap \overline{S}| + |Y \cap S|$. We showed in the proof of Theorem 4.11.1 that $|M^*| = |x| = |X \cap \overline{S}| + |Y \cap S|$, where M^* is a maximum matching in B. Hence $|M^*| = |U|$, implying that U is a minimum vertex cover and the proof is complete. □

We say that a matching M **covers** a set of vertices Z if every vertex in Z is incident with an edge from M. Recall that a matching is perfect if it covers

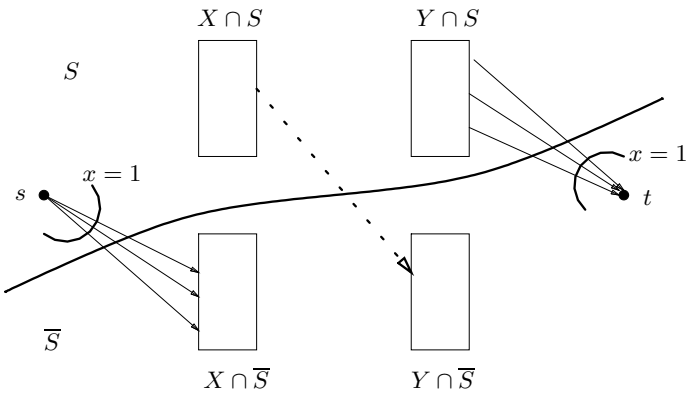


Figure 4.17 The situation when a maximum flow has been found. The thick dotted arc indicates that there is no arc between the two sets $X \cap S$ and $Y \cap \overline{S}$.

all vertices. We saw above that the simple proof of Theorem 4.11.1 was easily modified to a proof of König's theorem. We can also derive the following classical result due to Hall [493]. For an undirected graph $G = (V, E)$ and a subset $U \subset V$, we denote by $N(U)$ the set of vertices in $V - U$ which have at least one edge to a vertex in U.

Theorem 4.11.3 (Hall's theorem) *[493] A bipartite graph $B = (X, Y; E)$ has a matching covering X if and only if the following holds:*

$$|N(U)| \geq |U| \qquad \text{for every } U \subseteq X. \tag{4.25}$$

Proof: The necessity of (4.25) is clear since every vertex in U has a private neighbour in Y if B has a matching covering X.

Suppose now that (4.25) holds. Then $|Y| \geq |X|$ and by adding a set X' of $|Y| - |X|$ vertices to X and joining each of these completely to Y we can obtain a bipartite supergraph B' of B for which (4.25) also holds. Clearly B' has a perfect matching if and only if B has a matching covering X. So it suffices to prove that $|X| = |Y|$ and (4.25) implies that B has a perfect matching.

Let x be an integer maximum flow in the network $\mathcal{N}_B$ which is defined as in the proof of Theorem 4.11.1. If we can prove that $|x| = |X|$, then it follows from the proof of Theorem 4.11.1 that B has a perfect matching. So suppose $|x| < |X|$. By the proof of Theorem 4.11.2, we have $|x| = |X \cap \overline{S}| + |Y \cap S|$, where S is the set of vertices that are reachable from s in $\mathcal{N}_B(x)$. Since (4.25) holds and we argued in the proof of Theorem 4.11.2 that all neighbours of $X \cap S$ are in $Y \cap S$, we also have

$$|X| = |X \cap S| + |X \cap \overline{S}| \leq |Y \cap S| + |X \cap \overline{S}| = |x| < |X|,$$

a contradiction. Hence we must have $|x| = |X|$ and the proof is complete. □

The case $|X| = |Y|$ in Hall's theorem, the so-called marriage theorem, was proved much earlier (in 1917) by Frobenius.

Corollary 4.11.4 (The Marriage theorem) *[363] A bipartite graph $B = (X, Y; E)$ has a matching covering X if and only if $|X| = |Y|$ and (4.25) holds.* □

4.11.2 The Directed Chinese Postman Problem

Suppose a postman has to deliver mail along all the streets in a small[14] town. Assume furthermore that on one-way streets the mail boxes are all on one side of the street, whereas for two-way streets, there are mail boxes on both sides of the street. For obvious reasons the postman wishes to minimize the distance he has to travel in order to deliver all the mail and return home to his starting point. We show below how to solve this problem in polynomial time using minimum cost flows.

We can model the problem by a directed graph $D = (V, A)$ and a cost function $c : A \rightarrow \mathbb{R}_+$ where V contains a vertex for each intersection of streets in the town and the arcs model the streets. A 2-cycle corresponds to a two-way street and an arc which is not in a 2-cycle corresponds to a one-way street in the obvious way. The cost of an arc corresponds to the length of the corresponding street. Now it is easy to see that an optimal route for the postman corresponds to a minimum cost closed walk in D which traverses each arc at least once.

We have seen in Theorem 1.7.2 that if a digraph is eulerian, then it contains a closed trail which covers all arcs precisely once. Thus if D is eulerian, the optimal walk is simply a eulerian trail in D (using each arc exactly once). Below we show how to solve the general case by reducing the problem to a minimum cost circulation problem. First observe that there is no solution to the problem if D is not strongly connected, since any closed walk is strongly connected as a digraph. Hence we assume below that the digraph in question is strong, a realistic assumption when we think of the postman problem.

Let $D = (V, A)$ be a strong digraph and let c be a cost function on A. The cost $c(W)$ of a walk W is $\sum_{ij \in A} c_{ij} w_{ij}$ where w_{ij} denotes the number of times the arc ij occurs on W. Define $\mathcal{N}$ as the network $\mathcal{N} = (V, A, l \equiv 1, u \equiv \infty, c)$, that is, all arcs have lower bounds one, capacity infinity and cost equal to the cost on each arc.

Theorem 4.11.5 *The cost of a minimum cost circulation in $\mathcal{N}$ equals the minimum cost of a Chinese postman walk in D.*

Proof: Suppose W is a closed walk in D which uses each arc $ij \in A$ exactly $w_{ij} \geq 1$ times. Then it is easy to see that we can obtain a feasible circulation of cost $c(W)$ in $\mathcal{N}$ just by sending w_{ij} units of flow along each arc $ij \in A$.

[14] This assumption is to make sure that the postman can carry all the mail in his backpack, say. Without this assumption the problem becomes much harder.

Conversely, suppose x is an integer feasible circulation in $\mathcal{N}$. Form a directed multigraph $D' = (V, A')$ by letting A' contain x_{ij} copies of the arc ij for each $ij \in A$. It follows from the fact that x is an integer circulation that D' is an eulerian directed multigraph (see Figure 4.18). Hence, by Theorem 1.7.2, D' has an eulerian tour T. The tour T corresponds to a closed walk W in D which uses each arc at least once and clearly we have $c(W) = c^T x$. □

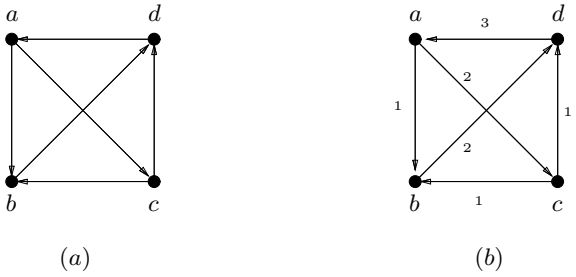


Figure 4.18 An instance of the directed Chinese postman problem. Part (a) shows a digraph with cost 1 (not shown) on every arc. Part (b) shows the values of a minimum cost circulation in the corresponding network. This circulation corresponds to the postman tour *abdacdacbda*.

The corresponding problem can be considered for a connected edge-weighted undirected graph $G = (V, E)$. Here the goal is to find a tour which traverses all edges in E at least once and minimizes the total weight of the tour. It is not hard to see that this problem is equivalent to finding a minimum cost subset of E so that by duplicating these edges we obtain a supergraph G^* of G in which all vertices have even degree. This is equivalent to G^* having an orientation as a strongly connected eulerian directed multigraph. Edmonds and Johnson showed how to solve this problem via matching techniques [287] . Finally there is also the Mixed Chinese Postman problem (MCPP) in which the input is a mixed graph $M = (V, A, E)$ with weights on the arcs and edges. Again the goal is to find minimum cost subsets $E' \subseteq E$ and $A' \subseteq A$ so that duplicating these edges and arcs in M results in a mixed supergraph M^* of M which can be oriented as a strongly connected eulerian directed multigraph. A necessary and sufficient condition for a mixed graph to have an orientation as an eulerian directed multigraph is given in Corollary 11.7.4. The MCCP is NP-hard [741]. From the point of view of practical relevance the MCPP is probably the most important of the three variants as it directly models situations such as garbage collection, snow removal, street sweeping, etc. For an exact algorithm for the MCPP see, e.g., Nobert and Picard [730].

4.11.3 Finding Subdigraphs with Prescribed Degrees

In some algorithms on directed multigraphs, an important step is to decide whether a directed multigraph D contains a subdigraph with prescribed degrees on the vertices. One such example is when we are interested in checking whether D contains a cycle factor (see Chapter 6). Below we show that such problems and more general versions of these problems can be answered using flows. See Exercise 4.60 for another application of flows to a similar question involving construction of directed multigraphs with specified in- and out-degrees. Other applications of the techniques illustrated in this subsection can be found in Chapter 12.

Theorem 4.11.6 *There exists a polynomial algorithm for the following problem. Given a directed multigraph $D = (V, A)$ with $V = \{v_1, v_2, \ldots, v_n\}$ and integers $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n$, find a subdigraph $D' = (V, A^*)$ of D which satisfies $d^+_{D'}(v_i) = a_i$ and $d^-_{D'}(v_i) = b_i$ for each $i = 1, 2, \ldots, n$, or show that no such subdigraph exists. Furthermore, if there are costs specified for each arc, then we can also find in polynomial time the cheapest (minimum cost) subdigraph which satisfies the degree conditions.*

Proof: We may assume that $a_i \leq d^+_D(v_i)$, $b_i \leq d^-_D(v_i)$ for each $i = 1, 2, \ldots, n$ and that $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i$. Clearly each of these conditions is necessary for the existence of D' and they can all be checked in time $O(n)$. Let $M = \sum_{i=1}^n a_i$ and define a network $\mathcal{N}$ as follows: $\mathcal{N} = (V' \cup V'' \cup \{s, t\}, A', l \equiv 0, u)$, where $V' = \{v'_1, v'_2, \ldots, v'_n\}$, $V'' = \{v''_1, v''_2, \ldots, v''_n\}$ and $A' = \{sv'_i : i = 1, 2, \ldots, n\} \cup \{v''_j t : j = 1, 2, \ldots, n\} \cup \{v'_i v''_j : v_i v_j \in A\}$. Finally, we let $u_{sv'_i} = a_i$, $u_{v''_i t} = b_i$ for $i = 1, 2, \ldots, n$ and all other arcs have capacity one.

Clearly the maximum possible value of an (s, t)-flow in $\mathcal{N}$ is M. We claim that $\mathcal{N}$ has an (s, t)-flow of value M if and only if D has the desired subdigraph.

Suppose first that $D' = (V, A^*)$ is a subdigraph satisfying $d^+_{D'}(v_i) = a_i$ and $d^-_{D'}(v_i) = b_i$ for each $i = 1, 2, \ldots, n$. Then the following is an (s, t)-flow of value M in $\mathcal{N}$: $x_{sv'_i} = a_i$, $x_{v''_i t} = b_i$, for each $i = 1, 2, \ldots, n$ and $x_{v'_i v''_j}$ equals one if $v_i v_j \in A^*$ and zero otherwise.

Suppose now that x is an integer (s, t)-flow of value M in $\mathcal{N}$ and let $A^* = \{v_i v_j : x_{v'_i v''_j} = 1\}$. Then $D' = (V, A^*)$ is the desired subdigraph.

It follows from our arguments above that we can find the desired subdigraph D' in polynomial time using any polynomial algorithm for finding a maximum flow in a network.

Observe also that if we have a cost function c on the arcs of D and let $\mathcal{N}$ inherit costs in the obvious way (arcs incident to s or t have cost zero), then finding a minimum cost subdigraph D' can be solved using any algorithm for minimum cost flows. □

It follows from Theorem 4.11.6 that we can decide whether a given digraph has a spanning k-regular subdigraph for some specified natural number k in

polynomial time. In fact, using minimum cost flows we can even find the cheapest such subdigraph in the case that there are costs on the arcs. What happens if we do not require the regular subdigraph to be spanning? If $k = 1$, then the existence version of the problem is trivial, since such a subdigraph exists unless D is acyclic. Yannakakis and Alon observed that already when $k \geq 2$ the existence version of the problem becomes $\mathcal{NP}$-complete. For details see [361].

4.11.4 Path-Cycle Factors in Directed Multigraphs

We start with three necessary and sufficient conditions for the existence of a cycle factor in a directed multigraph. The reason for giving all three is that in certain cases one of them provides a better way to deal with the problem under consideration than the other two. The first two parts are given in Ore's book [733]; the last is due to Yeo [919].

Proposition 4.11.7 *Let $D = (V, A)$ be a directed multigraph.*

(a) D has a cycle factor if and only if the bipartite representation $BG(D)$ of D contains a perfect matching.
(b) D has a cycle factor if and only if there is no subset X of V such that either $|\bigcup_{v\in X} N^+(v)| < |X|$ or $|\bigcup_{v\in X} N^-(v)| < |X|$.
(c) D has a cycle factor if and only if V cannot be partitioned into subsets Y, Z, R_1, R_2 such that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z|$ and Y is an independent set.

Proof: (a): Suppose $BG(D)$ has a perfect matching consisting of edges $v'_1v''_{\pi(1)}, ..., v'_nv''_{\pi(n)}$, where π is a permutation of the set $\{1, ..., n\}$. Then the arcs $v_1v_{\pi(1)}, ..., v_nv_{\pi(n)}$ form a cycle factor. Indeed, in the digraph D' induced by these arcs every vertex v_i has out-degree and in-degree equal to one and such a digraph is precisely a disjoint union of cycles.

Conversely, if $C_1 \cup C_2 \cup \ldots \cup C_k$ is a cycle factor in D, then for every $v_i \in V$ let $\pi(i)$ be the index of the successor of v_i on the cycle containing v_i. Then π induces a permutation of V and $\{v_iv_{\pi(i)} : v_i \in V\}$ is a perfect matching in $BG(D)$.

(b): Clearly D has a cycle factor if and only if the converse of D has a cycle factor, so it suffices to show that D has a cycle factor if and only if there is no subset X satisfying $|\bigcup_{v\in X} N^+(v)| < |X|$. Necessity is clear because if $|\bigcup_{v\in X} N^+(v)| < |X|$ holds for some X, then there can be no cycle subdigraph which covers all vertices of X (there are not enough distinct out-neighbours). So suppose $|\bigcup_{v\in X} N^+(v)| \geq |X|$ holds for all $X \subset V$. Then it is easy to see that $|N(X')| \geq |X'|$ holds for every subset $X' \subset V'$ of $BG(D)$ (where $V(BG(D)) = V' \cup V''$, recall Section 1.6). It follows from Theorem 4.11.3 that $BG(D)$ has a perfect matching and now we conclude from (a) that D has a cycle factor.

(c): We first prove the necessity. Suppose D has a cycle factor $\mathcal{F}$ and yet there is a partition Y, R_1, R_2, Z as described in (c). By deleting suitable arcs from the cycles in $\mathcal{F}$ we can find a collection of $|Y|$ vertex-disjoint paths such that all these paths start in Y and end at vertices of $V - Y$ each of which dominates some vertex in Y (here we used that Y is an independent set). However, this contradicts the existence of the partition Y, R_1, R_2, Z as described in (c), since it follows from the fact that $|Z| < |Y|$ that there can be at most $|Z|$ such paths in D (all such paths must pass through Z).

Now suppose that D has no cycle factor. Then we conclude from (b) that there exists a set X such that $|\bigcup_{v\in X} N^+(v)| < |X|$ holds. Let

$$Y = \{v \in X : d^-_{D\langle X\rangle}(v) = 0\}, R_1 = V - X - N^+(X), R_2 = X - Y, Z = N^+(X).$$

Then $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$ and Y is an independent set. Furthermore, since $|\bigcup_{v\in X} N^+(v)| < |X|$, we also have $|Z| + |X - Y| = |\bigcup_{v\in X} N^+(v)| < |X| = |X - Y| + |Y|$, implying that $|Z| < |Y|$. This shows that Y, Z, R_1, R_2 form a partition as in (c). □

It is not difficult to show that Proposition 4.11.7 remains valid for directed pseudographs (where we allow loops) provided that we consider a loop as a cycle (Exercise 4.66). We will use that extension below.

Combining Proposition 4.11.7 with Theorem 4.11.1 we obtain

Corollary 4.11.8 *The existence of a cycle factor in a digraph can be checked and a cycle factor found (if one exists) in time* $O(\sqrt{n}m)$. □

Recall that the path-cycle covering number $\mathrm{pcc}(D)$ of a directed pseudograph is the least positive integer k such that D has a k-path-cycle factor. The next result (whose proof is left as Exercise 4.68) and Theorem 4.11.1 imply that we can calculate $\mathrm{pcc}(D)$ in polynomial time for any directed pseudograph.

Proposition 4.11.9 *Let n be the number of vertices in a directed pseudograph D and let ν be the number of edges in a maximum matching of $BG(D)$. If $\nu = n$, then* $\mathrm{pcc}(D) = 1$, *otherwise* $\mathrm{pcc}(D) = n - \nu$. □

The following result by Gutin and Yeo generalizes Proposition 4.11.7(c).

Corollary 4.11.10 *[475] A digraph D has a k-path-cycle factor ($k \geq 0$) if and only if $V(D)$ cannot be partitioned into subsets Y, Z, R_1, R_2 such that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z| + k$ and Y is an independent set.*

Proof: Assume that $k \geq 1$. Let D' be an auxiliary digraph obtained from D by adding k new vertices $u_1, ..., u_k$ together with the arcs $\{u_i w,\ wu_i :\ w \in V(D),\ i = 1, 2, ..., k\}$. Observe that D has a k-path-cycle factor if and only if D' has a cycle factor. By Proposition 4.11.7(c), D' has a cycle factor if

and only if its vertex set cannot be partitioned into sets Y, Z', R_1, R_2 that satisfy $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z'|$ and Y is an independent set. Note that if Y, Z', R_1, R_2 exist in D', then the vertices $u_1, ..., u_k$ are in Z'. Let $Z = Z' - \{u_1, ..., u_k\}$. Clearly, the subsets Y, Z, R_1, R_2 satisfy $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z| + k$ and Y is an independent set. □

The proof above and Corollary 4.11.8 easily implies the first part of the following proposition.

Proposition 4.11.11 *Let D be a directed pseudograph and let k be a fixed non-negative integer. Then*

(a) In time $O(\sqrt{n}m)$ we can check whether D has a k-path-cycle-factor and construct one (if it exists).
(b) Given a k-path-cycle factor in D, in time $O(m)$, we can check whether D has a $(k-1)$-path-cycle factor and construct one (if it exists).

Proof: Exercise 4.69. □

4.12 Exercises

Unless otherwise stated, all numerical data in the exercises below are integers.

4.1. Find a feasible flow in the network $\mathcal{N}$ of Figure 4.19.

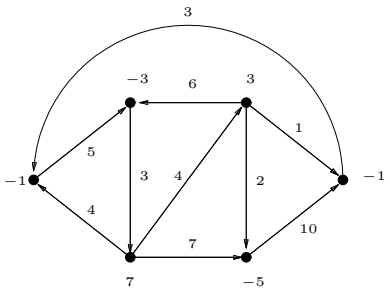


Figure 4.19 A network $\mathcal{N}$ with balance vector b specified at each vertex. All lower bounds and costs are zero and capacities are shown on the arcs.

4.2. Suppose the network $\mathcal{N} = (V, A, l, u, b, c)$ has some 2-cycle iji for which $c_{ij} \neq -c_{ji}$. Show how to transform $\mathcal{N}$ into another network $\mathcal{N}'$ without 2-cycles such that every feasible flow in $\mathcal{N}$ corresponds to a feasible flow in $\mathcal{N}'$ of the same cost. What is the complexity of this transformation?

4.3. Prove Lemma 4.2.1(a).

4.4. Prove Lemma 4.2.2.

4.5. Prove Lemma 4.2.3. In particular, argue why we need to take $l_{ts} = M$ rather than $l_{ts} = 0$.

4.6. Prove Lemma 4.2.4.

4.7. **(+) Fast decomposition of flows.** Prove Lemma 4.3.2.

4.8. **Decomposing an eulerian directed multigraph into arc-disjoint cycles**. Prove that the arc set of every eulerian directed multigraph can be decomposed into arc-disjoint cycles. Hint: form a circulation in an appropriate network and apply Theorem 4.3.1.

4.9. Find the residual network corresponding to the network and flow indicated in Figure 4.20.

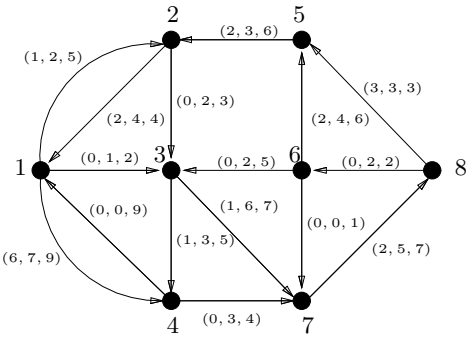


Figure 4.20 A network with a flow x. The notation for the arcs is (l, x, u).

4.10. Find the balance vector b_x for the flow x in Figure 4.20.

4.11. **Eliminating lower bounds on arcs in maximum flow problems.** Show how to reduce the maximum (s,t)-flow problem in a network $\mathcal{N}$ with some non-zero lower bounds on the arcs to the maximum (s',t')-flow problem in a network $\mathcal{N}'$ with source s' and sink t' and all lower bounds equal to zero.

4.12. Let x be a flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let $f(W)$ be a cycle flow of value δ in $\mathcal{N}(x)$. Show that the flow $x^* = x \oplus f(W)$ has the same balance vector as x in $\mathcal{N}$. Show also that the cost of x^* is given by $c^T x + c^T f(W)$.

4.13. Prove that the flow $\bar{x}$ defined in the proof of Theorem 4.4.3 is a feasible flow in $\mathcal{N}(x)$.

4.14. Let x be a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let y be a feasible flow in $\mathcal{N}(x)$. Show that $\mathcal{N}(x \oplus y) = \mathcal{N}(x)(y)$, where $\mathcal{N}(x)(y)$ denotes the residual network of $\mathcal{N}(x)$ with respect to y. That is, show that the two networks contain the same arcs and with the same residual capacities.

4.15. **An alternative decomposition of a flow.** Consider the proof of Theorem 4.3.1 and suppose that, instead of taking $\mu = \min\{b_x(i_0), -b_x(i_k), \delta\}$, we let $\mu = \delta$. What kind of decomposition into path and cycle flows will we get and what is the bound on their number?

4.16. **Structure of minimum (s,t)-cuts.** Decide which of the following is true or false. In each case either give a counterexample or a proof of correctness.
(a) If all arcs have different capacities, then there is a unique minimum (s,t)-cut.
(b) If we multiply the capacity of each arc by a constant k, then the structure (as subset of the vertices) of the minimum (s,t)-cuts is unchanged.
(c) If we add a constant k to the capacity of each arc, then the structure (as subset of the vertices) of the minimum (s,t)-cuts is unchanged.

4.17. **(+) The Ford-Fulkerson algorithm may never terminate if capacities are real numbers**.

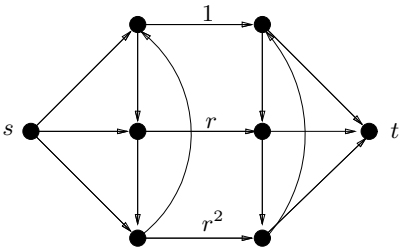


Figure 4.21 A bad network for the generic Ford-Fulkerson algorithm. All arcs except the three in the middle have capacity $r+2$. Those in the middle have capacities $1, r, r^2$, where r is the golden ratio.

Let $\mathcal{N}$ be the network in Figure 4.21. Here r is the golden ratio, i.e., $r^2 = 1-r$. Observe that $r^{n+2} = r^n - r^{n+1}$ for $n = 1, 2, \ldots$.
(a) Show that the value of a maximum flow in $\mathcal{N}$ is $1 + r + r^2 = 2$.
(b) Devise an infinite sequence of augmentations along properly chosen augmenting paths in the current residual network so that the flow value will converge towards $1 + \sum_{i=2}^{\infty} r^i = 2$. This shows that, when the capacities are non-rational numbers, the Ford-Fulkerson algorithm may never terminate. Hint: first augment by one unit and then by r^i units in the ith augmentation step, $i \geq 2$, along an appropriately chosen augmenting path.

4.18. **(+)** Prove that the Ford-Fulkerson algorithm will always terminate if all capacities are rational numbers.

4.19. Let S be a totally unimodular $p \times q$ matrix and I the $p \times p$ identity matrix. Show that the matrix $[S\ I]$ is also totally unimodular.

4.20. **Exact distance labels give a height function for the push-relabel algorithm.** Let $\mathcal{N}$ be a network with source s and sink t and let x be a preflow in $\mathcal{N}$ such that there is no (s,t)-path in $\mathcal{N}(x)$. Prove that if we let $h(i)$ equal the distance from i to t in $\mathcal{N}(x)$ for $i \in V - s$ and $h(s) = n$, then we obtain a height function.

4.21. **Bad performance of the push-relabel algorithm**. Give an example which shows that the push-relabel algorithm may use many applications of **push** and **lift** without sending any extra flow into t or back to s.

4.22. **Eliminating some useless work in the push-relabel algorithm**. Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. Suppose that we execute the generic push-relabel algorithm on $\mathcal{N}$. Let h be a height function with respect to $\mathcal{N}$ and x. We say that h has a **hole** at position $i+1$, for some $i < n$ at some point in the execution of the algorithm if at that time the following holds:
$|\{v : h(v) = j\}| > 0$ for every $j \leq i$ and
$|\{v : h(v) = i+1\}| = 0$.
Let h' be defined as follows:
$h'(v) = h(v)$ if $h(v) \in \{1, 2, \ldots, i\} \cup \{n, n+1, \ldots, 2n-1\}$
$h'(v) = n+1$ if $i < h(v) < n$.
(a) Prove that h' is a height function, that is, (4.14) is satisfied.
(b) Describe how to implement this modification of the height function efficiently so that it may be used as a subroutine in the push-relabel algorithm.
(c) Explain why changing the height function as above, when a hole is detected, may help speed up the push-relabel algorithm.

4.23. **Using the height function to detect a minimum cut after termination of the push-relabel algorithm.** Suppose x is a maximum (s,t)-flow that has been found by executing the push-relabel algorithm on a network $\mathcal{N} = (V, A, l \equiv 0, u)$. Describe a method to detect a minimum (s,t)-cut in $O(n)$ steps using the values of the height function upon termination of the algorithm.

4.24. **(+) Re-optimizing a maximum (s,t)-flow**. Suppose x is a maximum flow in a network $\mathcal{N} = (V, A, l \equiv 0, u)$. Show how to re-optimize x (that is, to change it to a feasible flow of maximum value) in each of the following cases:
(a) Increase the capacity of one arc by k units. Show that the new optimal solution can be found in time $O(km)$.
(b) Decrease the capacity of one arc by k units. Show that new optimal solution can be found in time $O(km)$. Hint: use Theorem 4.3.1.

4.25. **(+) Pulling and pushing flow, the MKM algorithm.** The purpose of this exercise is to introduce another, very efficient, method for finding a maximal (s,t)-flow in a layered network due to Malhotra, Kumar and Maheshwari [679]. Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$ be a layered network with $V_0 = \{s\}$ and $V_k = \{t\}$. Let y be a feasible (s,t)-flow which is not maximal in $\mathcal{L}$. For each vertex $i \in V - \{s,t\}$ let $\alpha_i, \beta_i, \rho_i$ be defined as follows:

$$\alpha_i = \sum_{ji \in A} u_{ji} - y_{ji}, \tag{4.26}$$

$$\beta_i = \sum_{ij \in A} u_{ij} - y_{ij}, \tag{4.27}$$

$$\rho_i = \min\{\alpha_i, \beta_i\}. \tag{4.28}$$

Let

$$\rho_s = \sum_{sj \in A} u_{sj} - y_{sj}, \rho_t = \sum_{jt \in A} u_{jt} - y_{jt}. \tag{4.29}$$

Finally let $\rho = \min_{i \in V}\{\rho_i\}$.
Suppose that $\rho > 0$ and let $i \in V$ be chosen such that $\rho = \rho_i$.

(a) Prove that it is possible to send an additional amount of ρ units from i to t (called **pushing from i to t**) and ρ units of flow from s to i in $\mathcal{L}$ (called **pulling from s to i**). Hint: use that the network is layered.

The observation above leads to the following algorithm $\mathcal{A}$ for finding a maximal flow in a layered network. Below the ρ-values always refer to the current flow.

The MKM algorithm

1. Start with the zero flow $y \equiv 0$ and calculate ρ_i for all $i \in V$. If some $i \in V$ has $\rho_i = 0$, then go to Step 6;
2. Choose i such that $\rho_i = \rho$;
3. Push ρ units of flow from i to t and pull ρ units from s to i;
4. Delete all arcs which are saturated with respect to the new flow. If this results in some vertex of in- or out-degree zero, then also delete that vertex and all incident arcs. Continue this until no more arcs can be deleted;
5. Calculate ρ_i for all vertices in the current layered network. If $\rho_i > 0$ for all vertices, then go to Step 2. Otherwise go to Step 6.
6. If $\rho_s = 0$ or $\rho_t = 0$, then halt;
7. If there is a vertex i with $\rho_i = 0$, then delete all such vertices and their incident arcs;
8. Go to Step 5.

(b) Prove that the algorithm above correctly determines a maximal flow in the input layered network $\mathcal{L}$.
The complexity of $\mathcal{A}$ depends on how we perform the different steps, especially Step 3. Suppose we apply the following rule for performing Step 3. We always push/pull ρ units one layer at a time. If j is the current vertex from (to) which we wish to send flow to (from) the next (previous) layer, then we always fill an arc with tail (head) j completely if there is still enough flow left and then continue to fill the next arc as much as possible.

(c) Argue that, using the rule above, we can implement the algorithm to run in $O(n^2)$ time. Hint: at least one vertex will be deleted between two consecutive applications of Step 3. Furthermore, one can keep the ρ-values effectively updated (explain how).

(d) Illustrate the algorithm on the layered network in Figure 4.10.

4.26. **Finding maximum (s,t)-flows by scaling.** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let U denote the maximum capacity of an arc in $\mathcal{N}$.

(a) **(−)** Prove that the capacity of a minimum (s,t)-cut is at most $U|A|$.

(b) Let C be a constant and let x be a feasible (s,t)-flow in $\mathcal{N}$. Show that in time $O(|A|)$ one can find an augmenting path of capacity at least C, or detect that no such path exists in $\mathcal{N}(x)$. Hint: consider the subnetwork of $\mathcal{N}(x)$ containing only arcs whose capacity is at least C.

(c) Consider the following algorithm:

Max-flow by scaling

1. $U := \max\{u_{ij} : ij \in A\}$;
2. $x_{ij} := 0$ for every $ij \in A$;
3. $C := 2^{\lfloor \log_2 U \rfloor}$;
4. while $C \geq 1$ do
5. while $\mathcal{N}(x)$ contains an augmenting path of capacity at least C

do augment x along P;
6. $C := C/2$
7. return x

Prove that the algorithm correctly determines a maximum flow in the input network $\mathcal{N}$.

(d) Argue that every time Step 4 is performed the residual capacity of every minimum (s,t)-cut is at most $2C|A|$.
(e) Argue that the number of augmentations performed in Step 5 is at most $O(|A|)$ before Step 6 is executed again.
(f) Conclude that **Max-flow by scaling** can be implemented so that its complexity becomes $O(|A|^2 \log U)$. Compare this complexity to that of other flow algorithms in this chapter.

4.27. Show how to find a maximum (s,t)-flow in the network of Figure 4.22 using
(a) The Ford-Fulkerson method;
(b) Dinic's algorithm;
(c) The push-relabel algorithm;
(d) The MKM algorithm described in Exercise 4.25;
(e) The scaling algorithm described in Exercise 4.26.

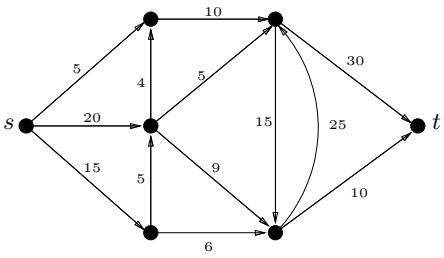


Figure 4.22 A network with lower bounds and cost equal to zero on all arcs and capacities as indicated on the arcs.

4.28. **(+) Rounding a real-valued flow**. Let $\mathcal{N} = (V, A, l, u)$ be a network with source s and sink t and all data on the arcs non-negative integers (note that some of the lower bounds may be non-zero). Suppose x is a real-valued feasible flow in $\mathcal{N}$ such that x_{ij} is a non-integer for at least one arc.
(a) Prove that there exists a feasible integer flow x' in $\mathcal{N}$ with the property that $|x_{ij} - x'_{ij}| < 1$ for every arc $ij \in A$.
(b) Suppose now that $|x|$ is an integer. Prove that there exists an integer feasible flow x'' in $\mathcal{N}$ such that $|x''| = |x|$.
(c) Describe algorithms to find the flows x', x'' above. What is the best complexity you can achieve?

4.29. **Finding a feasible circulation.** Turn the proof of Theorem 4.8.2 into a polynomial algorithm which either finds a feasible circulation, or a proof that none exists. What is the complexity of the algorithm?

4.30. **Residual networks of networks with non-zero lower bounds**. Show how to modify the definition of $x \oplus \tilde{x}$ in order to obtain an analogue of

Theorem 4.4.2 for the case of networks where some lower bounds are non-zero.

4.31. Show that a feasible circulation (if one exists) can always be found by just one max flow calculation in a suitable network. Hint: transform the network into an (s,t)-flow network with all lower bounds equal to zero.

4.32. **(+) Flows with balance vectors within prescribed intervals.** Let $\mathcal{N} = (V, A, l, u)$ be a network where $V = \{1, 2, \ldots, n\}$ and let $a_i \leq b_i$, $i = 1, 2 \ldots, n$, be integers. Prove that there exists a flow x in $\mathcal{N}$ which satisfies

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A, \tag{4.30}$$

$$a_i \leq b_x(i) \leq b_i \quad \forall i \in V, \tag{4.31}$$

if and only if the following three conditions are satisfied:

$$\sum_{i \in V} a_i \leq 0, \tag{4.32}$$

$$\sum_{i \in V} b_i \geq 0, \tag{4.33}$$

$$u(X, \overline{X}) \geq l(\overline{X}, X) + \max\{a(X), -b(\overline{X})\} \quad \forall X \subset V, \tag{4.34}$$

where $a(X) = \sum_{i \in X} a_i$.
Hint: construct a network which has a feasible circulation if and only if (4.30) and (4.31) hold. Then apply Theorem 4.8.2.

4.33. **Submodularity of the capacity function for cuts**. Let $\mathcal{N} = (V, A, l, u)$ be a network with source s and sink t. Prove that if $(S, \overline{S})$ and $(T, \overline{T})$ are (s,t)-cuts, then

$$u(S, \overline{S}) + u(T, \overline{T}) \geq u(S \cap T, \overline{S \cap T}) + u(S \cup T, \overline{S \cup T}).$$

Hint: consider the contribution of each arc in the network to the four cuts.

4.34. Show that if $(S, \overline{S})$ and $(T, \overline{T})$ are minimum (s,t)-cuts, then so are $(S \cap T, \overline{S \cap T})$ and $(S \cup T, \overline{S \cup T})$. Hint: use Exercise 4.33.

4.35. **(+) Finding special minimum cuts**. Suppose that x is a maximum (s,t)-flow in a network $\mathcal{N} = (V, A, l, u)$. Let

$$U = \{i : \text{ there exists an } (s,i)\text{-path in } \mathcal{N}(x)\},$$

$$W = \{j : \text{ there exists a } (j,t)\text{-path in } \mathcal{N}(x)\}.$$

Prove that $(U, \overline{U})$ and $(\overline{W}, W)$ are minimum (s,t)-cuts. Then prove that for every minimum (s,t)-cut (S, T) we have $U \subseteq S$ and $W \subseteq T$.

4.36. **(+)** Let x be an (s,t)-flow in a network $\mathcal{N} = (V, A, l, u)$. Explain how to find an augmenting path of maximum capacity in polynomial time. Hint: use a variation of Dijkstra's algorithm.

4.37. **(+) Augmenting along maximum capacity augmenting paths.** Show that if we always augment along an augmenting path with the maximum residual capacity, then the Ford-Fulkerson algorithm becomes a polynomial algorithm (Edmonds and Karp [288]). Hint: show that the number of augmentations is $O(m \log U)$, where U is the maximum capacity of an arc.

4.38. **Converting a maximum preflow to a maximum (s,t)-flow.** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. A preflow x in $\mathcal{N}$ is **maximum** if $|b_x(t)|$ equals the value of a maximum (s,t)-flow in $\mathcal{N}$.
(a) Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let y be a maximum preflow in $\mathcal{N}$. Prove that there exists a maximum (s,t)-flow x in $\mathcal{N}$ with the property that $x_{ij} \leq y_{ij}$ for every arc $ij \in A$. Hint: use flow decomposition.
(b) How fast can you convert a maximum preflow to a maximum (s,t)-flow?

4.39. **(−)** Prove Lemma 4.7.1.

4.40. **(−)** Prove Lemma 4.7.6.

4.41. Show that the complexity of Dinic's algorithm for unit capacity networks remains $O(n^{\frac{2}{3}}m)$ even if we allow the network to have 2-cycles. Hint: prove a modified version of Lemma 4.7.3 and apply that as we applied Lemma 4.7.3 in the proof of Theorem 4.7.4.

4.42. **Elimination of 2-cycles from simple networks.** Suppose that $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ is a simple unit capacity network with source s, sink t and that uvu is a 2-cycle in $\mathcal{N}$. Show that we may always delete one of the arcs uv or vu without affecting the value of a maximum (s,t)-flow in $\mathcal{N}$.

4.43. Prove Theorem 4.7.7. Hint: see the proof of Theorem 4.7.4.

4.44. Show how to derive Theorem 4.8.4 from Lemma 4.2.2 and Theorem 4.8.2.

4.45. **Scheduling jobs on identical machines.** Let J be a set of jobs which are to be processed on a set of identical machines (such as processors, airplanes, trucks, etc.). Each job is processed by one machine. There is a fixed schedule for the jobs, specifying that job $j \in J$ must start at time s_j and finish at time f_j. Furthermore, there is a transition time t_{ij} required to set up a machine which has just performed job i to perform job j (e.g., jobs could be different loads for trucks and t_{ij} could be time to drive a truck from the position of load i to that of load j). The goal is to find a feasible schedule for the jobs which requires as few machines as possible. Show how to formulate this problem as a minimum value (s,t)-flow problem.

4.46. **(+) Scheduling supervision of projects**. This exercise deals with a practical problem concerning the assignment of students to various projects in a course. All projects which are chosen by at least one student are to be supervised by one or more qualified teachers. Each student is supervised by one teacher only. There are n students, m different projects and t possible supervisors for the projects.
Let b_i, $i = 1, 2, \ldots, m$, denote the maximum number of students who may choose the same project (they work alone and hence need individual supervision). For each project i, $i = 1, 2, \ldots, m$, there is a subset $A_i \subseteq \{1, \ldots, t\}$ of the teachers who are capable of supervising the ith project. Finally each teacher j, $j = 1, 2, \ldots, t$, has an upper limit of k_j on the number of students (s)he can supervise.

Every student must be assigned exactly one project. We also assume that each student has ranked the projects from 1 to m according to the order of preference. Namely, the project the student would like best is ranked one. Denote the rank of project j by student i by r_{ij}.
The goal is to find an assignment $p(1), p(2), \ldots, p(n)$ of students to projects (that is, student i is assigned project $p(i)$) which respects the demands above and at the same time minimizes the sum $\sum_{i=1}^{n} r_{ip(i)}$.

(a) Show how to formulate the problem as a minimum cost flow problem.
(b) If we only wish to find a feasible assignment (i.e., one that does not violate the demands above), then which is the fastest algorithm you can devise?
(c) Which minimum cost flow algorithm among those in Section 4.10 will give the fastest algorithm for the problem when formulated as in question (a)?
(d) Let $p(1), p(2), \ldots, p(n)$ be an optimal assignment of students to projects. Suppose that *before* the actual supervision of the projects starts, some supervisor $j \in \{1, 2, \ldots, t\}$ lowers his/her capacity for supervision from k_j to $k'_j < k_j$.
Describe a fast algorithm which either proves that no feasible assignment exists or changes the assignment $p(1), p(2), \ldots, p(n)$ to a new optimal assignment $p'(1), p'(2), \ldots, p'(n)$ with respect to the new restrictions.
(e) Suppose now that the change in capacity only happens *after* the students have started working on the projects. The goal now is to find a new optimal and feasible solution or show that no feasible solution exists, while at the same time rescheduling as few students as possible to new projects (we assume that rescheduled students must start all over again). Explain briefly how to solve this variant of the problem. Hint: devise some measure of cost for rescheduling a student in a minimum cost flow model.

4.47. **(−)** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let $\mathcal{N}' = (V, A', l' \equiv 0, u', c')$ be obtained from $\mathcal{N}$ by adding a new arc ts with $u_{ts} = \infty$ and $c_{ts} = -1$ taking $u'_{ij} = u_{ij}$ for all $ij \in A$ and $c'_{ij} = 0$ for all $ij \in A$. Prove that there is a 1-1 correspondence between the minimum cost circulations in $\mathcal{N}'$ and the maximum (s, t)-flows in $\mathcal{N}$.

4.48. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with some arcs of infinite capacity and some arcs of negative cost.
(i) Show that there exists a finite optimal solution to the minimum cost flow problem (finding a feasible flow in $\mathcal{N}$ of minimum cost) if and only if $\mathcal{N}$ has no cycle C of negative cost such that all arcs of C have infinite capacity. Hint: study the difference between an arbitrary feasible solution and some fixed solution of finite cost.
(ii) Let K be the sum of all finite capacities and those b-values that are positive. Show that if there exists a finite optimal solution to the minimum cost flow problem for $\mathcal{N}$, then there exists one for which no arc has flow value more than K. Hint: use flow decomposition.

4.49. **Eliminating negative cost arcs from minimum cost flow problems.** Suppose $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ contains an arc uv of negative cost, but no cycle of infinite capacity and negative cost (see Exercise 4.48). Derive a result similar to Lemma 4.2.1 which can be used to transform $\mathcal{N}$ into a new network $\mathcal{N}^+$ in which all costs are non-negative and such that given any feasible flow x^+ in $\mathcal{N}^+$ we can obtain a feasible flow x in $\mathcal{N}$ and find the

cost of x efficiently, given the cost of x^+. Hint: reverse arcs of negative costs, negate the costs of such arcs and update balance vectors.

4.50. Prove Theorem 4.10.6.

4.51. Try to generalize the statement of Proposition 4.10.7 to the case when the paths P, P' do not necessarily have the same end-vertices. Hint: consider the network $\mathcal{N}_{st}$ obtained as in Lemma 4.2.2.

4.52. Show by an example that the cycle canceling algorithm may use $\Omega(mUC)$ augmentations before arriving at an optimal flow.

4.53. Show that the buildup algorithm of Section 4.10 can be applied to solve the assignment problem in time $O(n^3)$.

4.54. Show how to reduce the case when $\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{n} b_j$ to the case when the equality holds for the transportation problem. Hint: introduce new plants or retailers.

4.55. (−) Show how to reduce the problem of finding a matching in a bipartite graph $B = (X, Y, E)$ which maximizes the number of edges incident to vertices in X to the problem of finding a maximum matching in a bipartite graph.

4.56. (+) Prove that if D is a k-regular semicomplete digraph on n vertices, then D contains a spanning tournament T which is regular or almost regular ($|\delta^+(T) - \delta^-(T)| \leq 1$) depending on whether n is odd or even. Observe that every regular tournament has an odd number of vertices (Bang-Jensen [69]).

4.57. (+) **Generalized matchings in undirected graphs.** Let $G = (V, E)$ be an undirected graph. Recall that for any subset $S \subset V$ we denote by $N_G(S)$ the set of vertices in $V - S$ which have at least one edge to S. Prove that every graph G either has a vertex disjoint collection of edges $e_1, \ldots, e_k$ and odd cycles $C_1, \ldots, C_r$ covering V, or a set $S \subset V$ with $|N_G(S)| < |S|$ and S is independent. Derive an algorithm from your proof which either finds the desired generalized matching, or an independent subset S such that $|N(S)| < |S|$. Hint: use Theorem 4.8.2 on an appropriate network.

4.58. Prove the following theorem due to König [620].

Theorem 4.12.1 *[620] Every regular bipartite graph has a perfect matching.*

4.59. Find a minimum cost Chinese postman walk in the digraph of Figure 4.23.

4.60. Show how to formulate the following problem as a flow problem. Given two sequences of non-negative integers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ decide whether or not there exists a directed multigraph $D = (\{v_1, v_2, \ldots, v_n\}, A)$ such that $d_D^+(v_i) = a_i$ and $d_D^-(v_i) = b_i$ for each $i = 1, 2, \ldots, n$. Hint: use Theorem 4.11.3 or the proof idea of this theorem.

4.61. **Tree solution to a flow problem**. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with n vertices for which there exists a feasible flow and let $D = (V, A)$ be the underlying digraph of $\mathcal{N}$. Prove that there exists a feasible flow x in $\mathcal{N}$ such that the number of arcs on which $0 < x_{ij} < u_{ij}$ is at most $n-1$. We call such a feasible flow a **tree solution**. Hint: show that if C is a cycle

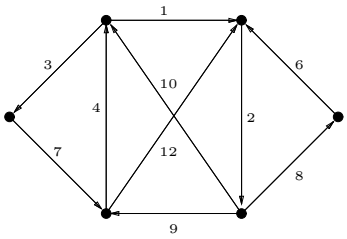


Figure 4.23 A digraph with costs on the arcs.

in $UG(D)$ where $0 < x_{ij} < u_{ij}$ for every arc on the cycle, then we can change the current flow such that the resulting flow x' is either 0 or u_{ij} for at least one arc ij of C and no new arc pq with $0 < x'_{pq} < u_{pq}$ is created.

4.62. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with n vertices for which there exists a feasible flow. Prove that there exists an optimal feasible flow which is a tree solution.

4.63. **Vertex potentials and flows.** Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network and x a feasible flow in $\mathcal{N}$. Prove that x is an optimal flow if and only if there exists a function $\pi : V \rightarrow \mathcal{R}$ such that $c^{\pi}_{ij} \geq 0$ for every arc ij in $\mathcal{N}(x)$. Here $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$ is the **reduced cost function** and the costs in $\mathcal{N}(x)$ are with respect to c^{π} instead of c. Hint: see Exercises 3.16-3.18.

4.64. **Complementary slackness conditions for optimality of a flow**. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network and x a feasible flow in $\mathcal{N}$. Prove that x is an optimal flow if and only if there exists a function $\pi : V \rightarrow \mathcal{R}$ such that the following holds where $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$ as above. Hint: use Exercise 4.63.

$$c^{\pi}_{ij} > 0 \quad \Rightarrow \quad x_{ij} = 0, \tag{4.35}$$

$$c^{\pi}_{ij} < 0 \quad \Rightarrow \quad x_{ij} = u_{ij}, \tag{4.36}$$

$$0 < x_{ij} < u_{ij} \quad \Rightarrow \quad c^{\pi}_{ij} = 0. \tag{4.37}$$

4.65. **(+) A primal-dual algorithm for minimum cost flows.** Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be a network with source s and sink t for which the value of a maximum (s, t)-flow is $K > 0$. Let x be an optimal (feasible) (s, t)-flow of value $k < K$ and let $\pi : V \rightarrow \mathcal{R}$ be chosen such that $c^{\pi}_{ij} \geq 0$ for every arc ij in $\mathcal{N}(x)$ (see Exercise 4.63). Define A_0 as those arcs ij of $\mathcal{N}(x)$ for which we have $c^{\pi}_{ij} = 0$ and let $\mathcal{N}_0$ be the subnetwork of $\mathcal{N}(x)$ induced by the arcs of A_0.

(a) Show that if y is a feasible (s, t)-flow in $\mathcal{N}_0$ of value p, then $x' = x \oplus y$ is an optimal (s, t)-flow of value $k + p$ in $\mathcal{N}$. Hint: verify that $c^{\pi}_{ij} \geq 0$ holds for every arc ij in $\mathcal{N}(x')$.

(b) Suppose y is a maximum (s, t)-flow in $\mathcal{N}_0$, but $x' = x \oplus y$ has value less than K. Let S denote the set of vertices which are reachable from s in $\mathcal{N}_0(y)$. Let $\epsilon, \epsilon_1, \epsilon_2$ be defined as follows. Here we let $\epsilon_i = \infty$ if there are no arcs in the corresponding set, $i = 1, 2$:

$$\epsilon_1 = \min \{c^{\pi}_{ij} | i \in S, j \in \overline{S}, c^{\pi}_{ij} > 0 \text{ and } x_{ij} < u_{ij}\},$$

$$\epsilon_2 = \min\{-c_{ij}^{\pi} | i \in \overline{S}, j \in S, c_{ij}^{\pi} < 0 \text{ and } x_{ij} > 0\}.$$

Let $\epsilon = \min\{\epsilon_1, \epsilon_2\}$. Prove that $\epsilon < \infty$.

(c) Now define π' as follows: $\pi'(v) := \pi(v) + \epsilon$ if $v \in S$ and $\pi'(v) := \pi(v)$ if $v \in \overline{S}$. Let $\mathcal{N}_0'$ contain those arcs of $\mathcal{N}(x')$ for which we have $c_{ij}^{\pi'} = 0$ and let S' denote the set of vertices which are reachable from s in $\mathcal{N}_0'$. Show that S is a proper subset of S' and that $c_{ij}^{\pi'} \geq 0$ holds for all arcs in $\mathcal{N}(x')$. Hint: use Exercise 4.14.

(d) If $t \notin S'$, then we can change π' as above (based on the set S' rather than S). Conclude that after at most $n-1$ such updates of the vector π', the current network $\mathcal{N}_0'$ contains an (s,t)-path.

(e) Use the observations above to design an algorithm that finds a minimum cost (s,t)-flow of value K in $\mathcal{N}$ by solving a sequence of maximum flow problems. What is the complexity of this algorithm?

4.66. **Cycle factors of directed pseudographs**. Prove that Proposition 4.11.7 also holds for directed pseudographs provided we consider a loop as a cycle.

4.67. **(+) Calculating the path-cycle covering number of a digraph.** Show how to find in time $O(\sqrt{n}m)$ the minimum integer k such that a given digraph D has a path-cycle factor with k paths. Hint: use minimum value flows in an appropriately constructed simple network.

4.68. Prove Proposition 4.11.9.

4.69. Prove Proposition 4.11.11. Hint: use the same network as in Exercise 4.67.

4.70. **(+) Path-cycle covering numbers of extensions of digraphs.** Let R be a digraph on r vertices, and let $l_1 \leq u_1, l_2 \leq u_2, ..., l_r \leq u_r$ be $2r$ non-negative integers. Let I_p denote an independent set on p vertices. Show how to find $\min\{\text{pcc}(R[I_{p_1}, ..., I_{p_r}]) : \ l_i \leq p_i \leq u_i, \ i = 1, ..., r\}$ in time $O(r^3)$. Hint: generalize the network you used in Exercise 4.67 (Bang-Jensen and Gutin [89, 454]).

4.71. Let $k \in \mathbb{Z}_+$. Show that a directed graph $D = (V, A)$ has a k-path-cycle factor if and only if $|\bigcup_{v \in X} N^+(v)| \geq |X| - k$ and $|\bigcup_{v \in X} N^-(v)| \geq |X| - k$.

5. Connectivity of Digraphs

The concept of connectivity is one of the most fundamental concepts in (directed) graph theory. There are numerous practical problems which can be formulated as (local) connectivity problems for digraphs and hence a significant part of this theory is also important from a practical point of view. Results on connectivity are often quite difficult and a deep insight may be required before one can obtain results in the area. Because of the very large number of important results on connectivity, we will devote this chapter as well as Chapters 10, 11, 12 and 14 to this area. Several connectivity problems, such as the connectivity augmentation problems in Sections 14.2 and 14.3, are of significant practical interest. These chapters illustrate several important topics as well as techniques that have been successful in solving local or global connectivity problems.

We will often consider directed multigraphs rather than directed graphs, since several results on arc-strong connectivity hold for this larger class and also it becomes easier to prove many results. However, when we consider vertex-strong connectivity, multiple arcs play no role and then we may assume that we are considering digraphs. Note that, unless we explicitly say otherwise, we will assume that we are working with a directed graph (i.e., there are no multiple arcs).

After introducing some new terminology, an efficient way of representing a directed multigraph as a network and a fast algorithm for finding the strong components of a digraph, we proceed to ear decompositions of strong directed multigraphs. We show how to use ear decompositions to obtain short proofs of several basic connectivity results. Then we state and prove Menger's theorem which is one of the most fundamental results in graph theory. Based on Menger's theorem, we describe various algorithms to determine the arc-strong and vertex-strong connectivity of a directed multigraph. In Section 5.6 we study the structure of directed multigraphs which are k-(arc)-strong but removing any arc destroys that property. We prove deep results by Mader on the structure of such directed multigraphs. Section 5.7 deals with digraphs which are k-strong but no vertex can be deleted without decreasing the vertex-strong connectivity. In Section 5.8 we consider connectivity properties of special classes of digraphs.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 191
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_5,

5.1 Additional Notation and Preliminaries

Let $D = (V, A)$ be a directed multigraph and let $X, Y \subseteq V$ be subsets of V. We denote by $d^+(X, Y)$ the number of arcs with tail in $X - Y$ and head in $Y - X$, i.e., $d^+(X, Y) = |(X - Y, Y - X)_D|$. Furthermore we let $d(X, Y) = d^+(X, Y) + d^+(Y, X)$. Hence we have $d^+(X) = d^+(X, V - X)$ and $d^-(X) = d^+(V - X, X)$. An arc xy **leaves** a set X if $x \in X$ and $y \in V - X$. The sets X, Y are **intersecting** if each of the sets $X - Y, X \cap Y, Y - X$ is non-empty. If also $V - (X \cup Y) \neq \emptyset$, then X and Y are **crossing**.

Let $\mathcal{F}$ be a family of subsets of a set S. We call a set $A \in \mathcal{F}$ a **member** of $\mathcal{F}$. The family $\mathcal{F}$ is an **intersecting family** (a **crossing family**) if $A, B \in \mathcal{F}$ implies $A \cup B, A \cap B \in \mathcal{F}$ whenever A, B are intersecting (crossing) members of $\mathcal{F}$. A family $\mathcal{F}$ of subsets of a set S is **laminar** if it contains no two intersecting members. That is, if $A, B \in \mathcal{F}$ and $A \cap B \neq \emptyset$, then either $A \subseteq B$ or $B \subseteq A$ holds. A family of sets is **cross-free** if it contains no two crossing members.

For an arbitrary directed multigraph $D = (V, A)$ and vertices $x, y \in V$, $\lambda(x, y)$ ($\kappa(x, y)$) denote the maximum number of arc-disjoint (internally disjoint) (x, y)-paths in D. The numbers $\lambda(x, y), \kappa(x, y)$ are called the **local arc-strong connectivity**, respectively, the **local vertex-strong connectivity** from x to y. Furthermore we let

$$\begin{aligned} \lambda'(D) &= \min_{x,y \in V} \lambda(x, y), \\ \kappa'(D) &= \min_{x,y \in V} \kappa(x, y). \end{aligned} \tag{5.1}$$

Analogously to the way we defined a cut with respect to an (s, t)-flow in Chapter 4 we define an **(s, t)-cut** to be a set of arcs of the form $(U, \bar{U})$, where $\bar{U} = V - U$ and $s \in U, t \in \bar{U}$. Recall that an (s, t)-separator is a subset $X \subseteq V(D) - \{s, t\}$ with the property that $D - X$ has no (s, t)-path. We also say that X **separates** s from t. Thus a separator of D is a set of vertices S such that S is an (s, t)-separator for some pair $s, t \in V(D)$ (recall the definition of a separator from Section 1.5). A **minimum separator** of D is a minimum cardinality separator X of D.

The following simple observation plays a central role in many proofs of connectivity results.

Proposition 5.1.1 *Let $D = (V, A)$ be a directed multigraph and let X, Y be subsets of V. Then the following holds:*

$$\begin{aligned} d^+(X) + d^+(Y) &= d^+(X \cup Y) + d^+(X \cap Y) + d(X, Y), \\ d^-(X) + d^-(Y) &= d^-(X \cup Y) + d^-(X \cap Y) + d(X, Y). \end{aligned} \tag{5.2}$$

Furthermore, if $d^-(X \cap Y) = d^+(X \cap Y)$, then we also have

$$d^+(X) + d^+(Y) = d^+(X - Y) + d^+(Y - X) + \epsilon,$$
$$d^-(X) + d^-(Y) = d^-(X - Y) + d^-(Y - X) + \epsilon, \tag{5.3}$$

where $\epsilon = d(X \cap Y, V - (X \cup Y))$.

Proof: Each of these equalities can easily be proved by considering the contribution of the different kinds of arcs that are counted on at least one side of the equality. For example, Figure 5.1 shows the possible edges contributing to at least one side of the first equality. □

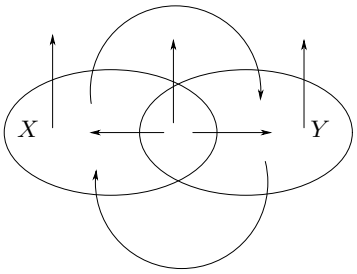


Figure 5.1 The various types of arcs contributing to the out-degrees of the sets $X, Y, X \cap Y$ and $X \cup Y$.

A set function f on a ground-set S is **submodular** if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ for all $X, Y \subseteq S$. The following easy corollary of Proposition 5.1.1 is very useful, as we shall see many times in this chapter.

Corollary 5.1.2 *For an arbitrary directed multigraph* D, d_D^+, d_D^- *are submodular functions on* $V(D)$. □

Recall that for a proper subset X of $V(D)$ we denote by $N^+(X)$ the set of out-neighbours of X. The next result shows that the functions $|N^-|, |N^+|$ are also submodular.

Proposition 5.1.3 *Let* $D = (V, A)$ *be a digraph and let* X, Y *be subsets of* V. *Then the following holds:*

$$|N^+(X)| + |N^+(Y)| \geq |N^+(X \cap Y)| + |N^+(X \cup Y)|,$$
$$|N^-(X)| + |N^-(Y)| \geq |N^-(X \cap Y)| + |N^-(X \cup Y)|.$$

Proof: These inequalities can easily be checked by considering the contributions of the different kind of neighbours of the sets $X, Y, X \cap Y$ and $X \cup Y$ (Exercise 5.1). □

5.1.1 The Network Representation of a Directed Multigraph

In many proofs and algorithms concerning directed multigraphs, it is convenient to think of a directed multigraph as a (flow) network. Here we will formalize this and prove an elementary result which will be applied in later sections.

Definition 5.1.4 *Let $D = (V, A)$ be a directed multigraph. The* **network representation** *of D, denoted $\mathcal{N}(D)$, is the following network: $\mathcal{N}(D) = (V, A', \ell \equiv 0, u)$ where A' contains the arc ij precisely when D contains at least one arc from i to j. For every arc $ij \in A'$ u_{ij} is equal to the number of arcs from i to j in D. See Figure 5.2.*

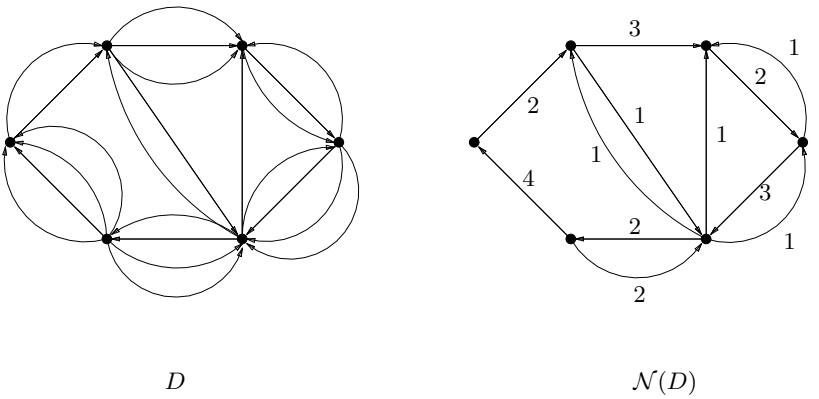


Figure 5.2 A directed multigraph D and its network representation $\mathcal{N}(D)$. Numbers on arcs indicate capacity in $\mathcal{N}(D)$.

The next lemma shows a useful connection between arc-disjoint paths in D and flows in $\mathcal{N}(D)$.

Lemma 5.1.5 *Let $D = (V, A)$ be a directed multigraph and let s, t be distinct vertices of V. Then $\lambda(s, t)$ equals the value of a maximum (s, t)-flow in $\mathcal{N}(D)$.*

Proof: Let $P_1, \ldots, P_r$ be a collection of pairwise arc-disjoint (s, t)-paths in D. These paths may use different copies of an arc between the same two vertices i and j, but, since the paths are arc-disjoint, in total they use no more than u_{ij} copies of the arc ij. Hence we can construct a feasible (s, t)-flow of value r in $\mathcal{N}(D)$ just by sending one unit of flow along each of the paths $P_1, \ldots, P_r$. Conversely, if x is any integral (s, t)-flow of value k in $\mathcal{N}(D)$ (recall Theorem 4.5.5), then by Theorem 4.3.1, x can be decomposed into k (s, t)-path-flows $f(P_1), \ldots, f(P_k)$ of value 1 (those that have a higher value $r > 1$ can be replaced by r (s, t)-path-flows of value 1 along the same path)

and some cycle flows. By the capacity constraint on the arcs, at most u_{ij} of these path flows use the arc ij. Hence we can replace the arcs used by each $f(P_i)$ by arcs in D in such a way that we obtain k arc-disjoint (s,t)-paths in D. This completes the proof of the lemma. □

5.2 Finding the Strong Components of a Digraph

In many problems on digraphs it suffices to consider the case of strong digraphs. For example, if we wish to find a cycle through a given vertex x in a digraph D, we need only consider the strong component of D containing x. Furthermore, certain properties, such as being hamiltonian, imply that the digraph in question must be strong. The aim of this section is to develop a fast algorithm for finding strong components in a digraph and in particular to recognize strong digraphs.

Tarjan [843] was the first to obtain an $O(n+m)$ algorithm to compute the strong components of a digraph. We start this section by presenting a simpler algorithm due to S. R. Kosaraju and M. Sharir, then we discuss its complexity and prove its correctness. Our presentation is adapted from the book [232] by Cormen, Leiserson, Rivest and Stein. The reader may wish to recall the definition of the DFS and DFSA algorithms from Sections 1.9 and 2.1, respectively.

SCA(D)
Input: A digraph D.
Output: The vertex sets of strong components of D.

1. Call DFSA(D) to compute the 'acyclic' ordering $v_1, v_2, \ldots, v_n$.
2. Compute the converse D' of D.
3. Call DFS(D'), but in the main loop of DFS consider the vertices according to the ordering $v_1, v_2, \ldots, v_n$. In the process of DFS(D') output the vertices of each DFS tree as the vertices of a strong component of D.

Figure 5.3 illustrates the strong component algorithm (SCA). Clearly, the complexity of SCA is $O(n+m)$. It is more difficult to establish the correctness of SCA. Several lemmas are needed.

The proof of our first lemma is simple and left as an exercise, Exercise 5.2.

Lemma 5.2.1 *If a pair x, y of vertices belongs to the same strong component S of a digraph D, then the vertices of every path between x and y are in S.* □

Lemma 5.2.2 *In any execution of DFS on a digraph, all vertices of the same strong component are placed in the same DFS tree.*

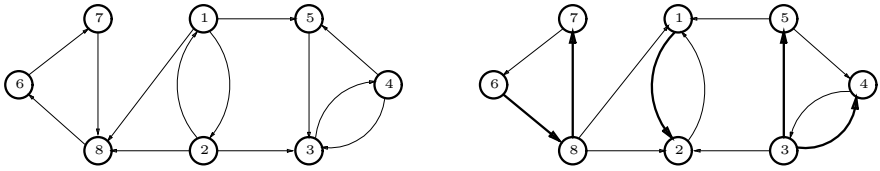


Figure 5.3 (a) A digraph D; the order of vertices found by DFSA is shown. (b) The converse D' of D; the bold arcs are the arcs of a DFS forest for D'.

Proof: Let S be a strong component of a digraph D, let r be the first vertex of S visited by DFS and let x be another vertex of S. Consider the time tvisit(r) of DFS. By Lemma 5.2.1, all vertices on an (r, x)-path belong to S and apart from r are unvisited. Thus, by Proposition 1.9.3, x belongs to the same DFS tree as r. □

In the rest of this section tvisit(u) and texpl(u) are the time-stamps calculated during the first step of SCA (recall that these depend on the order in which the DFS routine visits the vertices). The **forefather** $\phi(u)$ of a vertex u is the vertex w reachable from u such that texpl(w) is maximum.

Since u is reachable from itself, we have

$$\text{texpl}(u) \leq \text{texpl}(\phi(u)). \tag{5.4}$$

Clearly, by the definition of forefather

$$\text{if } v \text{ is reachable from } u, \text{ then } \text{texpl}(\phi(v)) \leq \text{texpl}(\phi(u)). \tag{5.5}$$

The next lemma gives a justification for the term 'forefather'.

Lemma 5.2.3 *In any execution DFS on a digraph D, every vertex $u \in V(D)$ is a descendant of its forefather $\phi(u)$.*

Proof: If $\phi(u) = u$, this lemma is trivially true. Thus, assume that $\phi(u) \neq u$ and consider the time tvisit(u) of DFS for D. Look at the status of $\phi(u)$. The vertex $\phi(u)$ cannot be already explored as that would mean $\text{texpl}(\phi(u)) < \text{texpl}(u)$, which is impossible. If $\phi(u)$ is already visited but not explored, then, by Corollary 1.9.2, u is a descendant of $\phi(u)$ and the lemma is proved.

It remains to show that $\phi(u)$ has been indeed visited before time tvisit(u). Assume it is not true and consider a $(u, \phi(u))$-path P. If every vertex of P except for u has not been visited yet (at the time tvisit(u)), then by Proposition 1.9.3, $\phi(u)$ is a descendant of u, i.e., $\text{texpl}(\phi(u)) < \text{texpl}(u)$, which is impossible. Suppose now that there is a vertex x in P apart from u which has been visited. Assume that x is the last such vertex in P (going from u towards $\phi(u)$). Clearly, x has not been explored yet (as x dominates an unvisited vertex). By Proposition 1.9.3 applied to $P[x, \phi(u)]$, $\phi(u)$ is a descendant of x. Thus, $\text{texpl}(\phi(u)) < \text{texpl}(x)$, which contradicts the definition of $\phi(u)$.

Thus, $\phi(u)$ has been indeed visited before time tvisit(u), which completes the proof of this lemma. □

Lemma 5.2.4 *For every application of DFS to a digraph D and for every $u \in V(D)$, the vertices u and $\phi(u)$ belong to the same strong component of D.*

Proof: There is a $(u, \phi(u))$-path by the definition of forefather. The existence of a path from $\phi(u)$ to u follows from Lemma 5.2.3. □

Now we show a stronger version of Lemma 5.2.4.

Lemma 5.2.5 *For every application of DFS to a digraph D and for every pair $u, v \in V(D)$, the vertices u and v belong to the same strong component of D if and only if $\phi(u) = \phi(v)$.*

Proof: If u and v belong to the same strong component of D, then every vertex reachable from one of them is reachable from the other. Hence, $\phi(u) = \phi(v)$. By Lemma 5.2.4, u and v belong to the same strong components as their forefathers. Thus, $\phi(u) = \phi(v)$ implies that u and v are in the same strong component of D. □

Theorem 5.2.6 *The algorithm SCA correctly finds the strong components of a digraph D.*

Proof: We prove, by induction on the number of DFS trees found in the execution of DFS on D', that the vertices of each of these trees induce a strong component of D. Each step of the inductive argument proves that the vertices of a DFS tree formed in D' induce a strong component of D provided the vertices of each of the previously formed DFS trees induce a strong component of D. The basis for induction is trivial, since the first tree obtained has no previous trees, and hence the assumption holds trivially. Recall that by the description of SCA, in the second application of DFS, we always start a new DFS tree from the vertex which currently has the highest value of texpl among vertices not yet in the DFS forest under construction.

Consider a DFS tree T with root r produced in DFS(D'). By the definition of a forefather $\phi(r) = r$. Indeed, r is reachable from itself and has the maximum texpl among the vertices reachable from r. Let $S(r) = \{v \in V(D) : \phi(v) = r\}$. We now prove that

$$V(T) = S(r). \tag{5.6}$$

By Lemmas 5.2.2 and 5.2.5, every vertex in $S(r)$ is in the same DFS tree. Since $r \in S(r)$ and r is the root of T, every vertex in $S(r)$ belongs to T. To complete the proof of (5.6), it remains to show that if $u \in V(T)$, then $u \in S(r)$, namely, if texpl$(\phi(x)) \neq$ texpl(r), then x is not placed in T. Suppose that texpl$(\phi(x)) \neq$ texpl(r) for some vertex x. By induction hypothesis, we

may assume that $\text{texpl}(\phi(x)) < \text{texpl}(r)$, since otherwise x is placed in the tree with root $\phi(x) \neq r$. If x was placed in T, then r would be reachable from x. By (5.5) and $\phi(r) = r$, this would mean $\text{texpl}(x) \geq \text{texpl}(\phi(r)) = \text{texpl}(r)$, a contradiction. □

5.3 Ear Decompositions

In this section we study the structure of strongly connected digraphs by introducing the concept of an ear decomposition (see Figure 5.4) and derive a number of results from this definition. Among other things, we reprove some of the results from Chapter 1.

Definition 5.3.1 *An* **ear decomposition** *of a directed multigraph D is a sequence $\mathcal{E} = \{P_0, P_1, P_2, \ldots, P_t\}$, where P_0 is a cycle or a vertex and each P_i is a path, or a cycle with the following properties:*

(a) P_i and P_j are arc-disjoint when $i \neq j$.

(b) For each $i = 0, 1, \ldots, t$: let D_i denote the digraph with vertices $\bigcup_{j=0}^{i} V(P_j)$ and arcs $\bigcup_{j=0}^{i} A(P_j)$. If P_i is a cycle, then it has precisely one vertex in common with $V(D_{i-1})$. Otherwise the end-vertices of P_i are distinct vertices of $V(D_{i-1})$ and no other vertex of P_i belongs to $V(D_{i-1})$.

(c) $\bigcup_{j=0}^{t} A(P_j) = A(D)$.

Each P_i, $0 \leq i \leq t$, is called an **ear** *of $\mathcal{E}$. The* **size** *of an ear P_i is the number $|A(P_i)|$ of arcs in the ear. The* **number of ears** *in $\mathcal{E}$ is the number $t + 1$. An ear P_i is* **trivial** *if $|A(P_i)| = 1$. All other ears are* **non-trivial**.

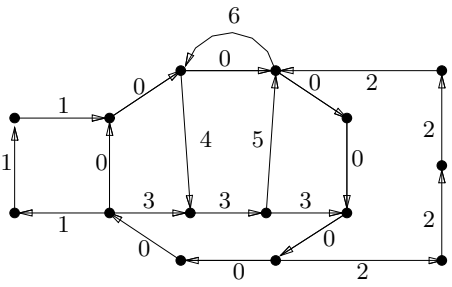


Figure 5.4 An ear decomposition $\mathcal{E} = \{P_0, P_1, \ldots, P_6\}$ of a digraph. The number on each arc indicates the number of the ear to which it belongs. The ears P_0, P_1, P_2, P_3 are non-trivial and the ears P_4, P_5, P_6 are trivial.

It is easy to see from the definition above that parallel arcs play no role in the structure of ear decompositions (if there is more than one copy of an arc,

at most one copy will be part of a non-trivial ear). Hence we assume below that we are dealing with digraphs.

Theorem 5.3.2 *Let D be a digraph on at least two vertices. Then D is strong if and only if it has an ear decomposition. Furthermore, if D is strong, every cycle can be used as starting cycle P_0 for an ear decomposition of D.*

Proof: We may assume that $|V(D)| \geq 3$ since otherwise the claim is trivial. Suppose first that D has an ear decomposition $\mathcal{E} = \{P_0, P_1, P_2, \ldots, P_t\}$. Note that the digraph P_0 is strong. Now it is easy to prove, by induction on the number of ears in $\mathcal{E}$, that D is strong. If D_i is strong, then D_{i+1} is also strong since it is obtained by adding a path with two end-vertices x, y in D_i and all other vertices outside of $V(D_i)$.

Conversely, assume that D is strong and let v be an arbitrary vertex in $V(D)$. Since $|V(D)| \geq 3$ and D is strong, there is some cycle $C = u_1u_2 \ldots u_r$, where $u_1 = u_r = v$. Let $P_0 := C$, $i := 0$ and execute phases 1 and 2 below:

Phase 1:

1. If every vertex of $V(D)$ is in $V(D_i)$, then go to Phase 2.
2. Let $i := i + 1$ and let u be a vertex not in $V(D_{i-1})$ such that there is some arc xu from $V(D_{i-1})$ to u.
3. Let P_i be a shortest path from u to $V(D_{i-1})$.
4. Take xP_i as the next ear and repeat Phase 1.

Phase 2:

1. For each remaining arc zw of D which was not included in $A(D_i)$ (i is the counter above) do the following:
2. Let $i := i + 1$ and let $P_i = zw$ (that is, include all these arcs as trivial ears).

To see that the algorithm above finds an ear decomposition of D, it suffices to check that we can always find an arc xu and a path from u to $V(D_i)$ as claimed in Phase 1. This follows easily from the fact that D is strong. □

There are several interesting consequences of Theorem 5.3.2 and its proof.

Corollary 5.3.3 *Every ear decomposition of a strong digraph on n vertices and m arcs has $m - n + 1$ ears.*

Proof: Exercise 5.3. □

The next lemma follows from Definition 5.3.1.

Lemma 5.3.4 *Let $\mathcal{E} = \{P_0, P_1, P_2, \ldots, P_k, a_1, a_2, \ldots, a_p\}$ be an ear decomposition of a strong digraph D such that P_0 is a cycle, each P_i is a path of length at least 2 and $a_1, a_2, \ldots a_p$ are arcs (the trivial ears). Then the digraph induced by $P_0 \cup P_1 \cup P_2 \cup \ldots \cup P_k$ is a strong spanning subdigraph of D with $\sum_{i=0}^{k} |A(P_i)|$ arcs.* □

Lemma 5.3.5 *Let D be a strong digraph on n vertices, let $P_0, P_1, P_2, \ldots, P_r$ be the non-trivial ears of an ear decomposition $\mathcal{E}$ of D and assume that $|A(P_i)| \geq 3$ for $i = 0, \ldots, s(\leq r)$. Then D has a strong spanning subdigraph D' on at most $2n - (|V(P_0)| + s)$ arcs.*

Proof: By Lemma 5.3.4, we can let D' be the strong spanning subdigraph of D formed by the union of $P_0, P_1, \ldots, P_r$, which implies that D' has n vertices and $\sum_{i=0}^{k} |A(P_i)|$ arcs. Note that P_0 contributes $|V(P_0)|$ vertices and $|V(P_0)|$ arcs to D' and each P_i, $1 \leq i \leq r$, contributes $|A(P_i)| - 1$ vertices and $|A(P_i)|$ arcs to D'. This implies that $|A(D')| - |V(D')| = r$.

As $P_1, P_2, \ldots, P_s$ all contribute at least two vertices to D' and each of the paths $P_{s+1}, P_{s+2}, \ldots, P_r$ contributes one vertex to D' we get that $n = |V(D')| \geq |V(P_0)| + 2s + (r - s)$. This implies the claim as follows.

$$|A(D')| = n + r \leq n + (n - |V(P_0)| - s) \leq 2n - (V(P_0) + s).\square$$

Corollary 5.3.6 *Every strong digraph D on n vertices has a strong spanning subdigraph with at most $2n-2$ arcs and equality holds only if the longest cycle in D has length 2 in which case $UG(D)$ is a tree.*

Proof: The $2n - 2$ bound follows from Lemma 5.3.5 since $|V(P_0)| \geq 2$. It also follows from the same lemma that we only have equality if the longest cycle in D has length two. As D is strong this only happens if $UG(D)$ is a tree. □

Corollary 5.3.7 *There is a linear algorithm to find an ear decomposition of a strong digraph D.*

Proof: This can be seen from the proof of Theorem 5.3.2. The proof itself is algorithmic and it is not too hard to see that if we use breadth-first search (see Section 3.3.1) together with a suitable data structure to find the path from u to $V(D_{i-1})$, then we can obtain a linear algorithm. Details are left to the interested reader as Exercise 5.18. □

Corollary 5.3.8 *It is an $\mathcal{NP}$-complete problem to decide whether a given digraph D has an ear decomposition with at most r non-trivial ears. It is $\mathcal{NP}$-complete to decide whether a given digraph D has an ear decomposition with at most q arcs in the non-trivial ears.*

Proof: Note that in both cases the numbers r (respectively q) are assumed to be part of the input to the problem. A strong digraph D has an ear decomposition with only one non-trivial ear (respectively, precisely n arcs in the non-trivial ears) if and only if D has a Hamilton cycle. Hence both claims follow from Theorem 6.1.1. □

The next two corollaries were proved in Chapter 1, but we reprove them here to illustrate an application of ear decompositions. Recall that a bridge of an undirected graph G is an edge e such that $G - e$ is not connected.

Corollary 5.3.9 *[780] A strong digraph D contains a spanning oriented subgraph which is strong if and only if $UG(D)$ has no bridge.*

Proof: If $UG(D)$ has a bridge, xy, then D contains the 2-cycle xyx, since D is strong. Observe that no matter which of these two arcs we delete we obtain a non-strong digraph. Suppose conversely that $UG(D)$ has no bridge. Consider again the proof of Theorem 5.3.2. If we can always choose the path from u to $V(D_{i-1})$ in such a way that it does not end in x, or contains at least one inner vertex, then it follows from the fact that we use shortest paths that no ear P_i, $i \geq 1$, contains a 2-cycle. In the remaining case, the only path from u to $V(D_{i-1})$ is the arc ux and hence the 2-cycle xux is a bridge in $UG(D)$. It remains to avoid using a 2-cycle as starting point (that is, as the cycle P_0). This can be done, unless all cycles in D are 2-cycles. If this is the case, then $UG(D)$ is a tree and every edge of $UG(D)$ is a bridge, contradicting the assumption. □

Corollary 5.3.10 *[162] A mixed graph M has a strong orientation if and only if M is strongly connected and has no bridge.*

Proof: This follows from Corollary 5.3.9, since we may associate with any mixed graph $M = (V, A, E)$ the directed graph D one obtains by replacing each edge in M by a 2-cycle. Clearly deleting an arc of a 2-cycle in D corresponds to orienting the corresponding edge in M. □

Ear decompositions of undirected graphs can be similarly defined. These play an important role in many proofs on undirected graphs, in particular in Matching Theory; see, e.g., the book by Lovász and Plummer [657].

5.4 Menger's Theorem

The following theorem, due to Menger [696], is one of the most fundamental results in graph theory.

Theorem 5.4.1 (Menger's theorem) *[696] Let D be a directed multigraph and let $u, v \in V(D)$ be a pair of distinct vertices. Then the following holds:*

(a) The maximum number of arc-disjoint (u, v)-paths equals the minimum number of arcs covering all (u, v)-paths and this minimum is attained for some (u, v)-cut $(X, \bar{X})$.
(b) If the arc uv is not in $A(D)$, then the maximum number of internally disjoint (u, v)-paths equals the minimum number of vertices in a (u, v)-separator.

Proof: First let us see that version (b) involving vertex-disjoint paths can be easily derived from the arc-disjoint version (a). Recall that multiple arcs play no role in questions regarding (internally) vertex-disjoint paths and hence we

can assume that the directed multigraph in question is actually a digraph. Given a digraph $D = (V, A)$ and $u, v \in V$ construct the digraph D_{ST} by the vertex splitting procedure (see Section 4.2.4). Now it is easy to check that arc-disjoint (u_s, v_t)-paths in D_{ST} correspond to internally disjoint (u, v)-paths in D (if a (u_s, v_t)-path in D_{ST} contains the vertex x_t (x_s) for some $x \neq u, v$, then it must also contain x_s (x_t)). Furthermore, for any set of ℓ arcs that cover all (u_s, v_t)-paths in D_{ST}, there exists a set of ℓ arcs of the form $w_t^1 w_s^1, \ldots, w_t^\ell w_s^\ell$ with the same property and such a set corresponds to a (u, v)-separator $X = \{w^1, \ldots, w^\ell\}$ in D. Hence it suffices to prove (a).

Because of the similarity between Menger's theorem (in the form (a)) and the max-flow min-cut theorem (Theorem 4.5.3), it is not very surprising that we can prove Menger's theorem in version (a) using Theorem 4.5.3. We did part of the work already in Section 5.1.1 where we showed that $\lambda(u, v)$ equals the value of a maximum (u, v)-flow in $\mathcal{N}(D)$. Similarly it is easy to see that every (u, v)-cut $(X, \bar{X})$ in D corresponds to a (u, v)-cut $(X, \bar{X})$ in $\mathcal{N}(D)$ of capacity $|(X, \bar{X})|$ and conversely. Now (a) follows from Theorem 4.5.3. □

As we shall see in Exercise 5.14, for networks where all capacities are integers, we can also derive the max-flow min-cut theorem from Menger's theorem.

In order to illustrate the use of submodularity in proofs concerning connectivity for digraphs we will give a second proof of Theorem 5.4.1(a) due to Frank [344][1]:

Second proof of Menger's theorem part (a):

Clearly the maximum number of arc-disjoint (s, t)-paths can be no more than the minimum size of an (s, t)-cut.

The proof of the other direction is by induction on the number of arcs in D. Let k denote the size of a minimum (s, t)-cut. The base case is when D has precisely k arcs. Then these all go from s to t and thus D has k arc-disjoint (s, t)-paths. Hence we can proceed to the induction step. Call a vertex set U **tight** if $s \in U, t \notin U$ and $d^+(U) = k$. If some arc xy does not leave any tight set, then we can remove it without creating an (s, t)-cut of size $(k-1)$ and the result follows by induction. Hence we can assume that every arc in D leaves a tight set.

Claim: If X and Y are tight sets, then the sets $X \cap Y$ and $X \cup Y$ are tight.

To see this we use the submodularity of d^+. First note that each of $X \cap Y$ and $X \cup Y$ contains s and none of them contains t. Hence, by our assumption, they both have degree at least k in D. Now using (5.2) we conclude

$$k + k = d^+(X) + d^+(Y) \geq d^+(X \cup Y) + d^+(X \cap Y) \geq k + k, \qquad (5.7)$$

by the remark above. It follows that each of $X \cup Y$ and $X \cap Y$ is tight and the claim is proved.

[1] Note that this proof requires no other prerequisites than Proposition 5.1.1.

If every arc in D is of the form st, then we are done, so we may assume that D has an arc su where $u \neq t$. Let T be the union of all tight sets that do not contain u. Then $T \neq \emptyset$, since the arc su leaves a tight set. By the claim, T is also tight. Now consider the set $T \cup \{u\}$. If there is no arc from u to $V - T$, then $d^+(T \cup \{u\}) \leq k - 1$, a contradiction since $T \cup \{u\}$ contains s but not t. Hence there must be some $v \in V - T - u$ such that $uv \in A(D)$. Now let D' be the directed multigraph we obtain from D by replacing[2] the two arcs su, uv by the arc sv. Suppose D' contains an (s,t)-cut of size less than k. That means that some set X containing s but not t has out-degree at most $k-1$ in D'. Since $d^+_D(X) \geq k$ it is easy to see that we must have $s, v \in X$ and $u \notin X$. Hence $d^+_D(X) = k$ and now we get a contradiction to the definition of T (since we know that $v \notin T$). Thus every (s,t)-cut in D' has size at least k. Since D' has fewer arcs than D, it follows by induction that D' contains k arc-disjoint (s,t)-paths. At most one of these can use the new arc sv (in which case we can replace this arc by the two we deleted). Thus it follows that D also has k arc-disjoint (s,t)-paths. □

Corollary 5.4.2 *Let $D = (V, A)$ be a directed multigraph. Then the following holds:*

(a) D is k-arc-strong if and only if it contains k arc-disjoint (s,t)-paths for every choice of distinct vertices $s, t \in V$.
(b) D is k-strong if and only if $|V(D)| \geq k+1$ and D contains k internally disjoint (s,t)-paths for every choice of distinct vertices $s, t \in V$.

Proof: Recall that, by definition, a directed multigraph $D = (V, A)$ is k-arc-strong if and only if $D - A'$ is strong for every $A' \subset A$ with $|A'| \leq k-1$. Now we see that (a) follows immediately from Theorem 5.4.1(a). To prove (b) we argue as follows: By definition (see Chapter 1) D is k-strong if and only if $|V(D)| \geq k+1$ and $D - X$ is strong for every $X \subset V$ such that $|X| \leq k-1$. Suppose that D has at least $k+1$ vertices but is not k-strong. Then we can find a subset $X \subset V$ of size at most $k-1$ such that $D - X$ is not strong. Let $D_1, \ldots, D_r$, $r \geq 2$, be any acyclic ordering of the strong components in $D - X$. Taking $s \in V(D_r)$ and $t \in V(D_1)$ it follows that there is no arc from s to t and that X is an (s,t)-separator of size less than k. Now it follows from Theorem 5.4.1(b) that D does not contain k internally disjoint paths from s to t.

Suppose conversely that there exists $s, t \in V(D)$ such that there are no k internally disjoint (s,t)-paths in D. If there is no arc from s to t, then it follows from Theorem 5.4.1(b) that D contains an (s,t)-separator X of size less than k. Then $D - X$ is not strong and, by definition, D is not k-strong. Hence we may assume that there is an arc st in D. Let r be the number of arcs from s to t in D (i.e., $\mu(s,t) = r$). If $r \geq k$, then k of these arcs form

[2] We will return to this useful reduction technique, called **splitting off** the arcs su, uv, in Chapter 14.

the desired (s,t)-paths, so by our assumption on s,t we have $r < k$. Now consider the digraph D' obtained from D by removing all arcs from s to t. In D' there can be no $k-r$ internally disjoint (s,t)-paths (since otherwise these together with the r arcs from s to t would give a collection of k internally disjoint (s,t)-paths). Thus, by Theorem 5.4.1(b), there exists a set $X' \subset V$ of size less than $k-r$ which forms an (s,t)-separator in D'.

Let A, B denote a partition of $V - X'$ in such a way that $s \in B$, $t \in A$ and there is no arc from B to A in D'. Since $|V| \geq k+1$, at least one of the sets A, B contains more than one vertex. Without loss of generality, we may assume that A contains a vertex v distinct from t. Now we see that $X' \cup \{t\}$ is an (s,v)-separator of size less than $k-r+1 \leq k$ in D and there is no arc from s to v in D. Applying Theorem 5.4.1(b) to this pair we conclude as above that D is not k-strong. □

Recall the numbers $\lambda'(D), \kappa'(D)$ which were defined in (5.1).

Corollary 5.4.3 *Let D be a directed multigraph. The number $\lambda'(D)$ equals the maximum number k for which D is k-arc-strong. The number $\kappa'(D)$ equals the maximum number k for which $k \leq |V|-1$ and D is k-strong. Hence we have $\lambda'(D) = \lambda(D)$ and $\kappa'(D) = \kappa(D)$.* □

5.5 Determining Arc- and Vertex-Strong Connectivity

In applications it is often important to be able to calculate the degree of arc-strong or vertex-strong connectivity of a directed multigraph. We can reduce the problem of finding $\kappa_D(x,y)$ to that of finding the local arc-strong connectivity from x_s to y_t in the digraph D_{ST} which we obtain by applying the vertex splitting procedure to D (see the proof of Corollary 5.4.2). Thus it is sufficient to consider arc-strong connectivity. It follows from Menger's theorem and Lemma 5.1.5 that $\lambda(D)$ can be found using $O(n^2)$ flow calculations. Namely, determine $\lambda(x,y)$ for all choices of $x, y \in V(D)$. However, as we shall see below we can actually find $\lambda(D)$ with just $O(n)$ flow calculations. For a similar result see Exercise 5.7.

Proposition 5.5.1 *[796] For any directed multigraph $D = (V,A)$ with $V = \{v_1, v_2, \ldots, v_n\}$ the arc-strong connectivity of D satisfies*

$$\lambda(D) = \min\{\lambda(v_1,v_2), \ldots, \lambda(v_{n-1},v_n), \lambda(v_n,v_1)\}.$$

Proof: Let $k = \lambda(D)$. By (5.1) and Corollary 5.4.3, $\lambda(D)$ is no more than the minimum of the numbers $\lambda(v_1,v_2), \ldots, \lambda(v_{n-1},v_n), \lambda(v_n,v_1)$. Hence it suffices to prove that $k = \lambda(v_i,v_{i+1})$ for some $i = 1,2,\ldots,n$ (where $v_{n+1} = v_1$). By Corollary 5.4.3 and Theorem 5.4.1, some $X \subset V$ has out-degree k. If there is an index $i \leq n-1$ such that $v_i \in X$ and $v_{i+1} \in V - X$, then, by Menger's

theorem, $\lambda(v_i, v_{i+1}) \leq k$ and the claim follows. If no such index exists, then we must have $X = \{v_r, v_{r+1}, \ldots, v_n\}$ for some $1 < r \leq n$. Now we get by Menger's theorem that $\lambda(v_n, v_1) \leq k$ and the proof is complete. □

Combining this with Lemma 5.1.5, we get the following result due to Schnorr:

Corollary 5.5.2 *[796] We can calculate the arc-strong connectivity of a directed multigraph by $O(n)$ maximum flow calculations in $\mathcal{N}(D)$.* □

If D has no multiple arcs, then its network representation $\mathcal{N}(D)$ has all capacities equal to 1 and it follows from Theorem 4.7.4 that we can find a maximum flow in $\mathcal{N}(D)$ in time $O(n^{\frac{2}{3}}m)$ and hence we can calculate $\lambda(D)$ in time $O(n^{\frac{5}{3}}m)$. Esfahanian and Hakimi [302] showed that the bound, n, on the number of max-flow calculations that is needed can be improved by a factor of at least 2.

Note that if we are only interested in deciding whether $\lambda(D) \geq k$, for some value of k which is not too big compared to m, then it may be better to use the simple labelling algorithm of Ford and Fulkerson (see Chapter 4). In that case it is sufficient to check for flows of value at least k, which can be done with k flow-augmenting paths and hence in time $O(km)$ per choice of source and terminal. Thus the overall complexity of finding $\lambda(D)$ is $O(knm)$ (see also the book by Even [306]). This can be improved slightly; see the paper [384] by Galil. For other connectivity algorithms based on flows, see, e.g., [305, 308].

One may ask if there is a way of deciding whether a given directed multigraph D is k-(arc)-strong without using flows. Extending work by Linial, Lovász and Wigderson [645] (see also [656]), Cheriyan and Reif [208] gave Monte Carlo and Las Vegas[3] type algorithms for k-strong connectivity in digraphs. Both algorithms in [208] are based on a characterization of k-strong digraphs via certain embeddings in the Euclidean space $\mathbb{R}^{k-1}$. The algorithms are faster than the algorithms described above, but the price is the chance of an error (for the Monte Carlo algorithm), respectively only the expected running time can be given (for the Las Vegas algorithm). We refer the reader to [208] for details.

The currently fastest algorithm to determine the arc-strong connectivity uses matroid intersection[4] and is due to Gabow [374]. This algorithm finds the arc-strong connectivity of a digraph D in time $O(\lambda(D)m \log(n^2/m))$. It is based on Edmonds' branching theorem (Theorem 9.3.1). In Chapter 9 we discuss the relation between arc-strong connectivity and arc-disjoint branchings, which is used in Gabow's algorithm. Gabow's approach also works very

[3] A Monte Carlo algorithm always terminates, but may make an error with some small probability, whereas a Las Vegas algorithm may (with some small probability) never terminate, but if it does, then the answer it provides is correct; see e.g., the book [180] by Brassard and Bratley.

[4] See Section 18.8 for the definition of the matroid intersection problem.

efficiently for the case when we want to decide whether $\lambda(D) \geq k$ for some number k.

The currently fastest algorithm to determine $\kappa(D)$ is due to Gabow [377]. The complexity of the algorithm is $O((n + \min\{\kappa(D)^{\frac{5}{2}}, \kappa(D)n^{\frac{3}{4}}\})m)$.

Nagamochi and Ibaraki [712] found a very elegant and effective way to calculate the edge-connectivity of an undirected graph without using flow algorithms. We describe their method briefly below (see also [352, 713]).

A **maximum adjacency ordering** of an undirected graph $G = (V, E)$ is an ordering $v_1, v_2, \ldots, v_n$ of its vertices, satisfying the following property:

$$d(v_{i+1}, V_i) \geq d(v_j, V_i) \text{ for } i \in [n], i < j \leq n, \tag{5.8}$$

where $V_i = \{v_1, v_2, \ldots, v_i\}$ and $d(X, Y)$ denotes the number of edges with one end in $X - Y$ and the other in $Y - X$.

Theorem 5.5.3 *[712]*

(a) Given any undirected graph G on n vertices, one can find a maximum adjacency ordering of G starting at a prescribed vertex v_1 in time $O(n + m)$.

(b) For every maximum adjacency ordering $v_1, v_2, \ldots, v_n$ of G we have $\lambda(v_{n-1}, v_n) = d_G(v_n)$. □

Corollary 5.5.4 *[712] There is an $O(nm + n^2)$ algorithm to determine the edge-connectivity of a graph with n vertices and m edges.*

Proof: This is an easy consequence of (b) and the fact that for every choice of $x, y \in V(G)$:

$$\lambda(G) = \min\{\lambda(x, y), \lambda(G/\{x, y\})\}, \tag{5.9}$$

where $G/\{x, y\}$ is the graph we obtain from G by contracting the set $\{x, y\}$. The equality (5.9) follows from the fact that $\lambda(G)$ equals the size of a minimum cut $(X, V - X)$ in G. If this cut separates x, y, then $\lambda(G) = \lambda(x, y)$ by Menger's theorem, and otherwise X is still a cut in $G/\{x, y\}$, implying that $\lambda(G) = \lambda(G/\{x, y\})$ (contractions do not decrease edge-connectivity). Hence we can start from an arbitrary maximum adjacency ordering $v_1, v_2, \ldots, v_n$. This gives us $\lambda(v_{n-1}, v_n)$. Save this number, contract $\{v_{n-1}, v_n\}$ and continue with a maximum adjacency ordering of $G/\{v_{n-1}, v_n\}$. The edge-connectivity of G is the minimum of the numbers saved. We leave the remaining details to the interested reader (see also the paper [714] by Nagamochi and Ibaraki). □

It is an interesting open problem whether some similar kind of ordering can be used to find the arc-strong connectivity of a directed multigraph. Note that (5.9) does not hold for arbitrary directed multigraphs. To see this consider Figure 5.5.

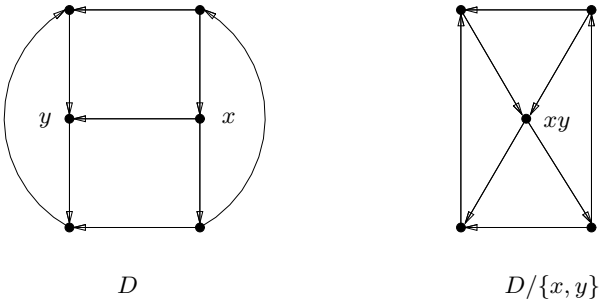


Figure 5.5 A digraph D with $\lambda(D) = 0$, $\lambda(x, y) = 2$ and $\lambda(D/\{x, y\}) = 1$.

5.6 Minimally k-(Arc)-Strong Directed Multigraphs

A directed multigraph $D = (V, A)$ is **minimally k-(arc)-strong** if D is k-(arc)-strong, but for every arc $e \in A$, $D - e$ is not k-(arc)-strong. From an application point of view it is very important to be able to identify a small subgraph of a k-(arc)-strong directed multigraph which is spanning and still k-(arc)-strong. The reason for this could be as follows. If many arcs of D are redundant, then it may make sense to discard these. If one is writing an algorithm for finding a certain structure that is based on k-(arc)-strong connectivity, then working with the smaller subgraph could speed up the algorithm, especially if k is relatively small compared to n.

Note, however, that if we are given a k-(arc)-strong directed multigraph $D = (V, A)$ and ask for the smallest number of arcs in a spanning k-(arc)-strong subgraph of D, then this is an $\mathcal{NP}$-hard problem. Indeed, a strong digraph D on n vertices has a strong spanning subgraph on n arcs if and only if D has a hamiltonian cycle. Hence, we must settle for finding spanning subgraphs with relatively few arcs. Since every k-arc-strong directed multigraph on n vertices has at least kn arcs, the proof of Theorem 5.6.1 together with Exercise 9.7 implies that there is a polynomial algorithm to find a spanning k-arc-strong subgraph with no more than twice the optimum number of arcs. We discuss this topic in more detail in Section 12.4.

5.6.1 Minimally k-Arc-Strong Directed Multigraphs

We start with a result by Dalmazzo which gives an upper bound on the number of arcs in any minimally k-arc-strong directed multigraph of order n.

Theorem 5.6.1 *[239] A minimally k-arc-strong directed multigraph has at most $2k(n-1)$ arcs and this is the best possible.*

Proof: Let $D = (V, A)$ be k-arc-strong and let s be a fixed vertex of V. By Corollary 5.4.2, $d^+(U), d^-(U) \geq k$ for every $\emptyset \neq U \subset V$. Hence, by Edmonds'

branching theorem (Theorem 9.3.1), D contains k arc-disjoint in-branchings $B^-_{s,1}, \ldots, B^-_{s,k}$ rooted at s and k arc-disjoint out-branchings $B^+_{s,1}, \ldots, B^+_{s,k}$ rooted at s. Let $A' = A(B^-_{s,1}) \cup \ldots \cup A(B^-_{s,k}) \cup A(B^+_{s,1}) \cup \ldots \cup A(B^+_{s,k})$ and let $D' = (V, A')$. Then D' is k-arc-strong and has at most $2k(n-1)$ arcs. Thus if D is minimally k-arc-strong, then $A = A'$. To see that this bound cannot be sharpened it suffices to consider the directed multigraph obtained from a tree T (as an undirected graph) and replacing each edge uv of T by k arcs from u to v and k arcs from v to u. □

Berg and Jordán [141] showed that for digraphs with sufficiently many vertices the bound in Theorem 5.6.1 can be improved. The complete bipartite digraph $\overleftrightarrow{K}_{k,n-k}$ shows that the bound on the number of arcs in Theorem 5.6.2 cannot be improved.

Theorem 5.6.2 *[141] There exists a function $f(k)$ such that every minimally k-arc-strong digraph on $n \geq f(k)$ vertices has at most $2k(n-k)$ arcs.* □

We now present some important results by Mader [667]. Combining these results with Theorem 14.1.2 we obtain a construction method (also due to Mader) to generate all k-arc-strong directed multigraphs.

A set $\emptyset \neq X \subset V$ is ***k*-in-critical** (***k*-out-critical**) if $d^-(X) = k$ ($d^+(X) = k$). When we do not want to specify whether X is k-in-critical or k-out-critical, we say that X is ***k*-critical**. It it easy to see that if $D = (V, A)$ is minimally k-arc-strong, then every arc uv leaves a k-out-critical set and enters a k-in-critical set. Applying (5.2) we obtain Lemma 5.6.3 below, which implies that every arc uv leaves precisely one minimal k-out-critical set X_u and enters precisely one minimal k-in-critical set Y_u. Here minimal means with respect to inclusion.

Lemma 5.6.3 *If X, Y are crossing k-in-critical sets in D, then $X \cap Y$ and $X \cup Y$ are also k-in-critical sets and $d(X, Y) = 0$.*

Proof: Suppose X, Y are crossing and k-in-critical. Using (5.2) we get

$$\begin{aligned} k + k &= d^-(X) + d^-(Y) \\ &= d^-(X \cup Y) + d^-(X \cap Y) + d(X, Y) \\ &\geq k + k, \end{aligned}$$

implying that $X \cap Y$ and $X \cap Y$ are both k-in-critical and $d(X, Y) = 0$. □

Intuitively, Lemma 5.6.3 implies that minimally k-arc-strong directed multigraphs have vertices of small in-degree and vertices small out-degree. The next result by Mader shows that this is indeed the case. In fact, a much stronger statement holds.

Theorem 5.6.4 *[667] Every minimally k-arc-strong directed multigraph has at least two vertices x, y with $d^+(x) = d^-(x) = d^+(y) = d^-(y) = k$.*

Proof: We give a proof due to Frank [344]. Let $\mathcal{R}$ be a family of k-in-critical sets with the property that

$$\text{every arc in } D \text{ enters at least one member of } \mathcal{R}. \tag{5.10}$$

By our remark above such a family exists since D is minimally k-arc-strong.

Our first goal is to make $\mathcal{R}$ cross-free (that is, we want to replace $\mathcal{R}$ by a new family $\mathcal{R}^*$ of k-in-critical sets such that $\mathcal{R}^*$ still satisfies (5.10) and no two members of $\mathcal{R}^*$ are crossing). To do this we apply the so-called **uncrossing technique** which is quite useful in several proofs. If there are crossing members X, Y in $\mathcal{R}$, then by Lemma 5.6.3, $X \cap Y$ and $X \cup Y$ are k-in-critical and $d(X, Y) = 0$. Hence every arc entering X or Y also enters $X \cup Y$, or $X \cap Y$. Thus we can replace the sets X, Y by $X \cap Y, X \cup Y$ in $\mathcal{R}$ (we only add sets if they are not already there). Since $|X \cap Y|^2 + |X \cup Y|^2 > |X|^2 + |Y|^2$ and the number of sets in R does not increase, we will end up with a family $\mathcal{R}$ which is cross-free. Note that we could have obtained such a family directly by choosing the members in $\mathcal{R}$ as the unique minimal k-in-critical sets entered by the arcs of A. However, this choice would make the proof more complicated, since we lose the freedom of just working with a cross-free family satisfying (5.10). We shall use this freedom in Case 2 below. Assume below that

$$\mathcal{R} \text{ is cross-free.} \tag{5.11}$$

Now the trick is to consider an arbitrary fixed vertex s and show that $V - s$ contains a vertex with in-degree and out-degree k. This will imply the theorem.

Let s be fixed and define the families $\mathcal{S}, \mathcal{U}$ and $\mathcal{L}$ as follows:

$$\mathcal{S} = \{X \in \mathcal{R} : s \notin X\}, \tag{5.12}$$

$$\mathcal{U} = \{V - X : s \in X \in \mathcal{R}\}, \tag{5.13}$$

$$\mathcal{L} = \mathcal{L}(\mathcal{R}) = \mathcal{S} \cup \mathcal{U}. \tag{5.14}$$

Claim A: The family $\mathcal{L}$ is laminar.

Proof of Claim A: We must show that no two members of $\mathcal{L}$ are intersecting. Suppose $X, Y \in \mathcal{L}$ are intersecting. Then X and Y cannot both be from $\mathcal{S}$ since then they are crossing and this contradicts (5.11). Similarly X and Y cannot both be from $\mathcal{U}$, since then $V - X, V - Y$ are crossing members of $\mathcal{R}$, a contradiction again. Finally, if $X \in \mathcal{S}$ and $Y \in \mathcal{U}$, then X and $V - Y$ are crossing members of $\mathcal{R}$, contradicting (5.11). This proves that $\mathcal{L}$ is laminar. □

By the choice of $\mathcal{S}$ and $\mathcal{U}$ we have the following property:

Every arc either enters a member of $\mathcal{S}$ or leaves a member of $\mathcal{U}$ (or both). (5.15)

Suppose $\mathcal{R}$ is chosen such that (5.10) and (5.11) hold and furthermore

$$\sum_{X\in\mathcal{L}} |X| \text{ is minimal.} \tag{5.16}$$

To complete the proof of the theorem we consider two cases.

Case 1: Every member of $\mathcal{L}$ has size one:

Let $X = \{x \in V - s : \{x\} \in \mathcal{S}\}$ and $Y = \{y \in V - s : \{y\} \in \mathcal{U}\}$. Then X cannot be empty, since every arc leaving s enters X. Similarly Y is non-empty. Now if $X \cap Y = \emptyset$, then there can be no arc leaving X, by the definition of X and (5.15). However, $d^+(X) \geq k$, since D is k-arc-strong and hence we have shown that $X \cap Y \neq \emptyset$. Let t be any element in $X \cap Y$, then we have $d^+(t) = d^-(t) = k$.

Case 2: Some member Z of $\mathcal{L}$ has size at least two:

Choose Z such that $|Z|$ is minimal among all members of $\mathcal{L}$ of size at least two.

Note that if we consider the converse D^* of D and let $\mathcal{R}^* = \{V - X : X \in \mathcal{R}\}$ and then define $\mathcal{S}^*$,$\mathcal{U}^*$ as we defined $\mathcal{S}$ and $\mathcal{U}$ from $\mathcal{R}$, then $\mathcal{S}^* = \mathcal{U}$ and $\mathcal{U}^* = \mathcal{S}$. Furthermore, the corresponding family $\mathcal{L}^*$ satisfies (5.15) and (5.16). This shows that we may assume without loss of generality that $Z \in \mathcal{S}$. We claim that

$$\text{the directed multigraph } D\langle Z\rangle \text{ is strongly connected.} \tag{5.17}$$

Suppose this is not the case and let Z_1, Z_2 be a partition of Z with the property that there are no arcs from Z_2 to Z_1. Then we have $k \leq d^-(Z_1) \leq d^-(Z) = k$, implying that Z_1 is k-in-critical and that every arc that enters Z also enters Z_1. Let $\mathcal{R}' = \mathcal{R} - \{Z\} + \{Z_1\}$, $\mathcal{S}' = \mathcal{S} - \{Z\} + \{Z_1\}$ and let $\mathcal{L}' = \mathcal{S}' \cup \mathcal{U}$. Then $\mathcal{L}'$ still satisfies (5.15) and

$$\sum_{X\in\mathcal{L}'} |X| < \sum_{X\in\mathcal{L}} |X|.$$

However, this contradicts the choice of $\mathcal{R}$. Thus we have shown that $D\langle Z\rangle$ is strongly connected. This establishes (5.17).

We return to the proof of the theorem. Let

$$A = \{z \in Z : \{z\} \in \mathcal{S}\}, B = \{z \in Z : \{z\} \in \mathcal{U}\}.$$

If $A \cap B \neq \emptyset$, then any vertex $t \in A \cap B$ has $d^+(t) = d^-(t)$ and we are done. Suppose $A \cap B = \emptyset$. Then we claim that

$$A = \emptyset. \tag{5.18}$$

Suppose $A \neq \emptyset$. By the choice of $\mathcal{R}$ so that $\mathcal{L}$ satisfies (5.16), we cannot leave out any set without violating (5.15). Hence we cannot have $A = Z$, because then we could leave out Z without violating (5.15). Now (5.17) implies that there is an arc uv from A to $Z - A$. Since $\mathcal{L}$ satisfies (5.15), the arc uv either enters some member of $\mathcal{S}$ or leaves a member of $\mathcal{U}$. If it enters a member M of $\mathcal{S}$, then by the definition of A, M cannot be of size one. On the other hand, by the fact that $\mathcal{L}$ is laminar and the minimality of Z, M also cannot have size at least two. Hence uv must leave a member W of $\mathcal{U}$. Since we have assumed $A \cap B = \emptyset$, this must be a set of size more than one. Using that $\mathcal{L}$ is laminar it follows that $W \subset Z$, contradicting the choice of Z. Hence we must have $A = \emptyset$ and (5.18) is established. Next we claim that

$$B = Z. \tag{5.19}$$

Since $A = \emptyset$ and Z is minimal among all members of $\mathcal{L}$ of size at least 2, every arc with both ends in Z must leave a member of B (using the same arguments as above). Hence $B \neq \emptyset$ and we must have $B = Z$, since otherwise (5.17) would imply the existence of an arc from $Z - B$ to B, contradicting what we just concluded.

Now we are ready to complete the proof of the theorem. Since $B = Z$, every vertex in Z has out-degree k. Thus we have

$$\begin{aligned} k|Z| &= \sum_{v \in Z} d^+(v) \\ &= d^+(Z) + |A(D\langle Z\rangle)| \\ &\geq k + |A(D\langle Z\rangle)| \\ &= k + (\sum_{v \in Z} d^-(v)) - d^-(Z) \\ &= \sum_{v \in Z} d^-(v) \\ &\geq k|Z|. \end{aligned}$$

Hence equality holds everywhere, in particular, every vertex in Z has in- and out-degree k. □

When D is a k-arc-strong directed multigraph containing parallel arcs, the number of vertices with in- and out-degree equal to k may be exactly two, as shown by taking an undirected path $v_1v_2 \dots v_r$, $r \geq 3$, and replacing each edge v_iv_{i+1} by k arcs from v_i to v_{i+1} and k arcs from v_{i+1} to v_i. Here only v_1 and v_r have the desired semi-degrees [925].

Now we turn to digraphs. Let $u^+(D)$ (respectively, $u^=(D)$ and $u^-(D)$) denote the number of vertices v in D with $d^+(v) = k < d^-(v)$ (respectively, $d^+(v) = d^-(v) = k$ and $d^+(v) > k = d^-(v)$).

Mader [675] gave examples showing that for $k = 2, 3$ there are even infinite families of directed graphs (no multiple arcs) which are k-arc-strong and still have $u^-(D) + u^=(D) + u^-(D)$ bounded by a constant. However, for $k \geq 4$ such examples do not seem to exist.

Conjecture 5.6.5 *[675] For every integer $k \geq 4$, there is a $c_k > 0$ such that every minimally k-arc-strong digraph D on n vertices has $u^-(D) + 2u^=(D) + u^-(D) \geq c_k n$.*

Verifying another conjecture of Mader [675], Yuan and Cai [925] proved the following result.

Theorem 5.6.6 *Let D be a minimally k-arc-strong digraph, then*

$$u^+(D) + 2u^=(D) + u^-(D) \geq 2k + 2.\square$$

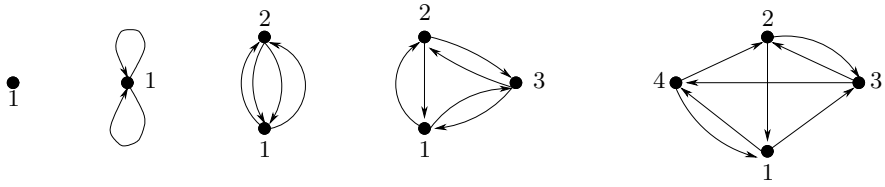


Figure 5.6 A construction of a 2-arc-strong directed multigraph starting from a single vertex.

Using Corollary 14.1.3 and Theorem 5.6.4 one can obtain the following complete characterization of k-arc-strong directed multigraphs, due to Mader [669].

Theorem 5.6.7 *[669] A directed multigraph D is k-arc-strong if and only if it can be obtained by starting from a single vertex and applying the following two operations (in any order):*

Operation A: Add a new arc connecting existing vertices.
Operation B: Choose k distinct arcs $u_1v_1, \ldots u_kv_k$ and replace these by $2k$ new arcs $u_1s, \ldots, u_ks, sv_1, \ldots, sv_k$, where s is a new vertex.

Proof: Clearly operation A preserves the property of being k-arc-strong. To see that this also holds for operation B we apply Menger's theorem. Suppose D is k-arc-strong and D' is obtained from D by one application of operation B but D' is not k-arc-strong. Let $U \subset V(D')$ be some subset such that $d^+_{D'}(U) \leq k - 1$. Then we must have $U \neq \{s\}$ and $U \neq V(D)$, since clearly s has in- and out-degree k in D'. Now it is easy to see that the corresponding set $U - s$ has out-degree less than k in D, a contradiction. From these observations it is easy to prove by induction on the number of vertices that every directed

multigraph that can be constructed via operations A and B is k-arc-strong. Here we assume by definition that every directed pseudograph having just one vertex is k-arc-strong.

The other direction can be proved using induction on the number of arcs. If D is k-arc-strong and not minimally k-arc-strong, then we can remove an arc and apply induction. Otherwise it follows from Theorem 5.6.4 that D contains a vertex s such that $d^+(s) = d^-(s) = k$. According to Theorem 14.1.3, this vertex and the $2k$ arcs incident with it can be replaced by k new arcs in such a way that the resulting directed multigraph D' is k-arc-strong. By induction D' can be constructed via operations A and B. Since we can go from D' back to D by using operation B once, D can be constructed using operations A and B. □

See Figure 5.6 for an illustration of the theorem.

5.6.2 Minimally k-Strong Digraphs

In this section $D = (V, A)$ is always a digraph (i.e., no multiple arcs) and hence we know that $d^+(v) = |N^+(v)|$ for each $v \in V$. Several results from this section will be used in Section 12.4.

Mader [671] proved that when we consider vertex connectivity instead of arc connectivity, the bound of Theorem 5.6.1 can be improved as follows. The complete bipartite digraph $\overleftrightarrow{K}_{k,n-k}$ shows that this bound on the number of arcs is best possible.

Theorem 5.6.8 *[671] Every minimally k-strong digraph on $n \geq 4k+3$ vertices has at most $2k(n-k)$ arcs.* □

We saw in the last section that every minimally k-arc-strong directed multigraph has at least two vertices with in- and out-degree equal to k. Mader conjectures that this is also the case for vertex-strong connectivity in digraphs.

Conjecture 5.6.9 *[671] Every minimally k-strong digraph contains at least two vertices such that both have in- and out-degree k.*

This longstanding conjecture is still open and seems very difficult. For $k = 1$ the truth of Conjecture 5.6.9 follows from Theorem 5.6.4. Mader [676] has proved the conjecture for $k = 2$. For all other values of k the conjecture is open. Examples by Mader [667] show that one cannot replace two by three in the conjecture.

Before reading the next couple of pages, the reader is advised to consult Subsection 14.3.1 to understand the definition of a one-way pair. An arc e of a k-strong digraph is **k-critical** if $D - e$ is not k-strong. By Lemma 14.3.2, for each k-critical arc uv we can associate sets T_{uv}, H_{uv} such that (T_{uv}, H_{uv}) is a one-way pair in $D - uv$ and $h(T_{uv}, H_{uv}) = k - 1$. This one-way pair

may not be unique, but below we always assume that we have chosen a fixed one-way pair for each k-critical arc in D.

Lemma 5.6.10 *Let $D = (V, A)$ be a k-strong digraph. Then the following is true:*

(a) If D has two k-critical arcs ux, uy, such that $d^+(u) \geq k+1$, then $|T_{uy}| > |H_{ux}|$.

(b) If D has two k-critical arcs xu, yu, such that $d^-(u) \geq k+1$, then $|H_{xu}| > |T_{yu}|$.

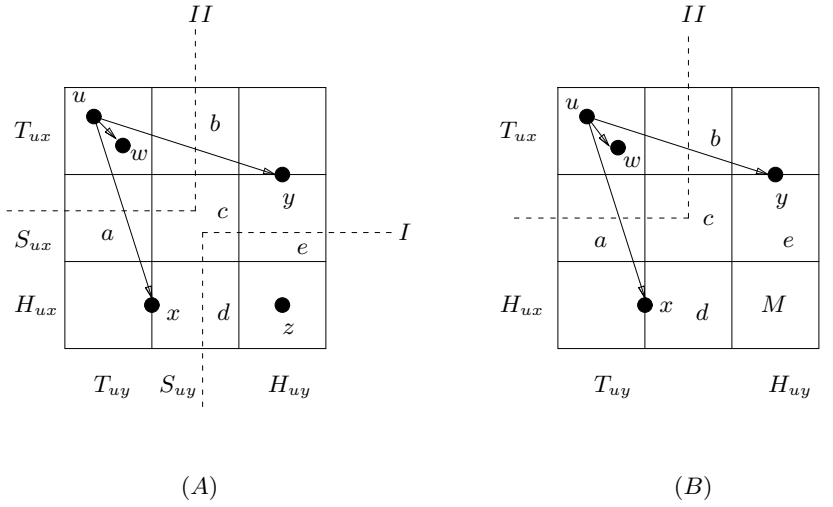


Figure 5.7 Illustration of the proof of Lemma 5.6.10. Part (A) illustrates the case when $H_{ux} \cap H_{uy} \neq \emptyset$. Part (B) illustrates the case when $H_{ux} \cap H_{uy} = \emptyset$. The first row of each 3×3 diagram corresponds to the set T_{ux}. The first column corresponds to T_{uy} and so on. The positions of x, y indicate that they can be in either of the two neighbouring cells. The numbers a, b, c, d, e denote the cardinality of the sets corresponding to their cell.

Proof: Since (b) follows from (a) by considering the converse of D, it suffices to prove (a). Hence we assume that ux, uy are k-critical arcs of D and that $d^+(u) \geq k+1$. Let $(T_{ux}, H_{ux}), (T_{uy}, H_{uy})$ be the pairs associated with ux, uy above. Note that these are not one-way pairs in D, since there is a (unique) arc, namely, ux (uy) which goes from T_{ux} (T_{uy}) to H_{ux} (H_{uy}). Let also $S_{ux} = V - (T_{ux} \cup H_{ux})$ and $S_{uy} = V - (T_{uy} \cup H_{uy})$. Then $|S_{ux}| = |S_{uy}| = k-1$ and $x \in H_{ux} - H_{uy}, y \in H_{uy} - H_{ux}$. It will be useful to study Figure 5.7 while reading the proof.

Let a, b, c, d, e be defined as in Figure 5.7. Since each of the sets S_{ux}, S_{uy} has size $k-1$ we see that

$$a + b + 2c + d + e = 2k - 2. \tag{5.20}$$

We claim that $H_{ux} \cap H_{uy} = \emptyset$. Suppose this is not the case and let $z \in H_{ux} \cap H_{uy}$ be arbitrarily chosen. Now it follows from the fact that (T_{ux}, H_{ux}) is a one-way pair in $D - ux$ and (T_{uy}, H_{uy}) is a one-way pair in $D - uy$, that the set C_I, indicated by the line I in Figure 5.7, separates u from z in D. Hence $c + d + e \geq k$, since D is k-strong. Now (5.20) implies that the set C_{II}, indicated by the line II, has size at most $k - 2$. Since $d^+(u) \geq k + 1$ and u has precisely two arcs, namely, ux, uy out of $T_{ux} \cap T_{uy}$ in $D - C_{II}$, we see that there is some out-neighbour w of u inside $T_{ux} \cap T_{uy}$. But now it is easy to see that $C_{II} \cup \{u\}$ separates w from z, contradicting that D is k-strong. Hence we have shown that $H_{ux} \cap H_{uy} = \emptyset$.

To complete the proof, we only need to show that $a \geq d$. Suppose this is not the case. Then in particular $d \geq 1$ and the size of the set C_{II} is at most $|S_{uy}| + a - d \leq k - 2$. Thus as above we can argue that u has an out-neighbour w inside $T_{ux} \cap T_{uy}$. Now $C_{II} \cup \{u\}$ separates w from x in D, a contradiction. □

An **anti-directed trail** is the digraph $\bar{T}$ one obtains from a closed undirected trail T of even length by fixing a traversal of T and orienting the edges so that every second vertex v has in-degree zero when we consider just the two arcs between v and its successor and predecessor on T. We denote the anti-directed trail $\bar{T}$ by $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2 \dots v_r\bar{v}_rv_1$, where $\bar{v}_i$ indicates that the vertex $\bar{v}_i$ is dominated by both its successor and its predecessor on the trail T. A vertex which dominates (is dominated by) both its successor and its predecessor on $\bar{T}$ is a **source** (**sink**) of $\bar{T}$. Note that if a vertex v is repeated on $\bar{T}$, then v may be both a source and a sink. An **anti-directed cycle** is an anti-directed trail in which no vertex occurs twice (that is, the underlying graph is just a cycle). See Figure 5.8 for an illustration of the definitions.

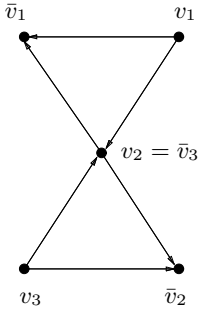


Figure 5.8 An anti-directed trail $v_1\bar{v}_1v_2\bar{v}_2v_3\bar{v}_3v_1$ on 6 vertices. The vertex $v_2 = \bar{v}_3$ is both a source and a sink of $\bar{T}$. Note that $\bar{T}$ contains no anti-directed cycle.

Now we can prove the following important result due to Mader:

Theorem 5.6.11 *[671] Let D be a k-strong digraph containing an anti-directed trail $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2 \dots v_r\bar{v}_rv_1$. Then at least one of the following holds:*

(a) Some arc $e \in A(\bar{T})$ is not k-critical in D.
(b) Some source v_i of $\bar{T}$ has out-degree k in D.
(c) Some sink $\bar{v}_j$ of $\bar{T}$ has in-degree k in D.

Proof: If (b) or (c) holds there is nothing to prove so suppose that $d^+(v_i) \geq k+1$ for each source and $d^-(\bar{v}_j) \geq k+1$ for each sink of $\bar{T}$. We shall prove that (a) holds.

Suppose to the contrary that every arc e on $\bar{T}$ is k-critical. Applying Lemma 5.6.10(a) to the arcs $v_1\bar{v}_r, v_1\bar{v}_1$, we obtain $|T_{v_1\bar{v}_r}| > |H_{v_1\bar{v}_1}|$. Similarly, we get from Lemma 5.6.10(b) that $|H_{v_1\bar{v}_1}| > |T_{v_2\bar{v}_1}|$. Repeating this argument around the trail we reach the following contradiction:

$$|T_{v_1\bar{v}_r}| > |H_{v_1\bar{v}_1}| > |T_{v_2\bar{v}_1}| > |H_{v_2\bar{v}_2}| > \dots > |H_{v_r\bar{v}_r}| > |T_{v_1\bar{v}_r}|.$$

Hence we have shown that (a) holds. □

The following is an easy consequence.

Corollary 5.6.12 *[671] Every minimally k-strong digraph contains a vertex x of in-degree k, or a vertex y of out-degree k.* □

Using Theorem 5.6.11, Mader proved the following much stronger statement.

Theorem 5.6.13 *[676] Every minimally k-strong digraph contains at least $k+1$ vertices of out-degree k and at least $k+1$ vertices of in-degree k.* □

This is best possible for digraphs on at most $2k+1$ vertices as shown by $\overleftrightarrow{K}_{k,k+1}$. When the number of vertices becomes larger, the following result gives a better bound and shows that the number of vertices of out-degree k (and hence also the number of vertices of in-degree k) grows as n grows.

Theorem 5.6.14 *[676] Every minimally k-strong digraph on n vertices contains at least $\frac{k-1}{2k-1}\sqrt{\frac{2n}{k+2}}$ vertices of out-degree k.* □

The following conjecture by Mader would imply that, in fact, the right lower bound grows linearly with n.

Conjecture 5.6.15 *[675] Every minimally k-strong digraph on n vertices contains at least $\frac{n-k}{k+1} + k$ vertices with out-degree equal to k.*

Proposition 5.6.16 *[676] For $k \geq 2$, every minimally k-strong digraph contains a vertex v with $\min\{d^+(v), d^-(v)\} = k$ and $\max\{d^+(v), d^-(v)\} \leq 2k-2$.* □

For more on the topic see the surveys [675] and [677] by Mader. Theorem 5.6.11 has many other nice consequences. Here is one for undirected graphs.

Corollary 5.6.17 *[665] Let C be a cycle of a k-connected undirected graph G. Then either C contains an edge e which can be removed without decreasing the connectivity of G, or some vertex $v \in V(C)$ has degree k in G.*

Proof: To see this, it suffices to consider the complete biorientation D of G and notice that $D - xy$ is k-strong if and only if $D - \{xy, yx\}$ is k-strong (Exercise 5.22) which happens if and only if $G - e$ is k-connected, where $e = xy$. Next, observe that in D, the cycle C either corresponds to one anti-directed trail C', obtained by alternating the orientation on the arcs taken twice around the cycle C, when $|C|$ is odd, or to two anti-directed cycles C', C'' when $|C|$ is even. Now the claim follows from Theorem 5.6.11. □

One reason why Corollary 5.6.17 is important is the following easy consequence concerning augmentations of undirected graphs, which was pointed out by Jordán.

Corollary 5.6.18 *[577] Let $G = (V, E)$ be an undirected graph which is k-connected, but not $(k+1)$-connected. Then every minimal set of edges F which augments the connectivity of G to $(k+1)$ induces a forest.* □

For directed graphs one obtains the following result, due to Jordán, on augmentations from k-strong to $(k+1)$-strong connectivity. Compare this with Theorem 14.3.8.

Corollary 5.6.19 *[575] Let $D = (V, A)$ be a directed graph which is k-strong, but not $(k+1)$-strong and let F be a minimal set of new arcs, whose addition to D gives a $(k+1)$-strong digraph. Then the digraph induced by the arcs in F contains no anti-directed trail.* □

One can also apply Theorem 5.6.11 to questions like: how many arcs can be deleted from a k-strong digraph, so that it still remains $(k-1)$-strong [675] (for undirected graphs see [164])? One easy consequence is the following.

Corollary 5.6.20 *[675] If $D = (V, A)$ is minimally k-strong and $D' = (V, A')$ is a spanning $(k-1)$-strong subgraph of D, then the difference $D_0 = (V, A - A')$ contains no anti-directed trail.*

Proof: Suppose $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2 \dots v_r\bar{v}_rv_1$ is an anti-directed trail in D_0. Since D is minimally k-strong, (a) cannot hold in Theorem 5.6.11. Suppose without loss of generality that (b) holds, then some source v_i has $d^+_D(v_i) = k$. However, since $d^+_{\bar{T}}(v_i) = 2$, this implies that $d^+_{D'}(v_i) = k - 2$, contradicting the fact that D' is $(k-1)$-strong. □

Theorem 5.6.11 has many other important applications. We illustrate one such application in Section 12.4.1.

5.7 Critically k-Strong Digraphs

In this section we always consider directed graphs (no multiple arcs). A vertex v of a digraph D is **critical** if $\kappa(D-v) < \kappa(D)$. The goal of this section is to illustrate some conditions under which we can always find a non-critical vertex in a digraph D. First observe that there can be no function $f(k)$ with the property that every k-strong digraph D with at least $f(k)$ vertices has a vertex v such that $D-v$ is still k-strong. This is not even the case for tournaments. To see this consider the example due to Thomassen (private communication, 1985) in Figure 5.9.

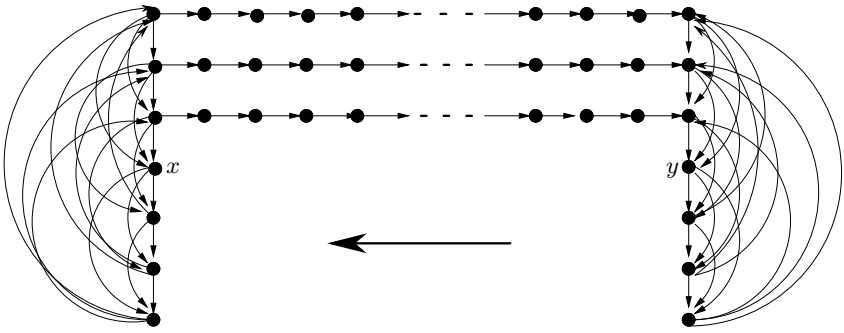


Figure 5.9 A family $\mathcal{T}$ of 3-strong tournaments (the three paths from left to right can be arbitrary long). The big arc indicates that all arcs not explicitly shown go from right to left. It can be verified (Exercise 5.26) that each tournament in $\mathcal{T}$ is 3-strong and has the property that every vertex other than x, y is critical. Thus after removing at most two vertices we obtain a 3-strong tournament in which every vertex is critical.

The example in Figure 5.9 can easily be generalized to arbitrary degrees of vertex-strong connectivity, by replacing each of the tournaments on seven vertices (right and left side of the figure) by the kth power of a $(2k+1)$-cycle and replacing the three long paths by k long paths starting at the top k vertices in the left copy and ending at the top k vertices in the right copy.

Below we discuss some results by Mader on sufficient conditions for a k-strong digraph to contain a non-critical vertex.

Definition 5.7.1 *Let D have $\kappa(D) = k$. A* **fragment** *in D is a subset $X \subset V$ with the property that either $|N^+(X)| = k$ and $X \cup N^+(X) \neq V$, or $|N^-(X)| = k$ and $X \cup N^-(X) \neq V$.*

Thus a fragment X corresponds to a one-way pair (X, Y) with $h(X, Y) = k$. Mader proved the following important result:

Theorem 5.7.2 *[672] Every critically k-strong digraph contains a fragment of size at most k.* □

This was conjectured by Hamidoune [494, Conjecture 4.8.3] who also conjectured the next two results, both of which are easy consequences of Theorem 5.7.2.

Corollary 5.7.3 *[672] Every critically k-strong digraph contains a vertex x with in-degree, or out-degree less than $2k$.*

Proof: Let $D = (V, A)$ be a critically k-strong digraph. By Theorem 5.7.2, D contains a fragment X with $|X| \leq k$. By considering the converse of D if necessary, we may assume that $|N^+(X)| = k$. We prove that every vertex of X has out-degree at most $2k-1$. Let $x \in X$ be arbitrary. Note that every out-neighbour of x outside X contributes to $|N^+(X)|$, implying that there are at most k of these. Now the claim follows from the fact that $d^+_{D\langle X\rangle}(x) \leq k-1$. □

We leave the proof of the next easy consequence as Exercise 5.24.

Corollary 5.7.4 *[672] Every critically k-strong oriented graph contains a vertex x with in-degree, or out-degree less than $\lfloor \frac{3k-1}{2} \rfloor$.* □

In [677] Mader surveys a number of results on k-strong digraphs D with the property that whenever we delete a set of vertices X of size at most $k' \leq k$ the resulting digraph $D - X$ is only $(k - |X|)$-strong. Such digraphs are also called (k, k')**-critical** and our discussion above on critically k-strong digraphs correspond to the case $k' = 1$.

Among other things, Mader shows that for every prime power k there exists a $(k+1, 2)$-critical digraph without 2-cycles.

Conjecture 5.7.5 *[677] For every $k \geq 2$, there is only a finite number of $(k, 2)$-critical oriented graphs.*

Mader also conjectured the following and showed by an example that this does not hold for infinite digraphs (implying that also the conjecture above is false for infinite oriented graphs).

Conjecture 5.7.6 *[677] If D is a (k, r)-critical digraph with $r > (2k+1)/3$, then D is the complete digraph $\overleftrightarrow{K}_{k+1}$.*

Conjecture 5.7.7 *[677] For all $k, r \in \mathbb{N}$, there is an integer $h(k, r)$ so that every finite, k-strong digraph $D = (V, A)$ on at least $h(k, r)$ vertices contains a set $W \subset V$ with $|W| = r$ such that $\kappa(D - W) \geq k - 2$ holds.*

The case $r = 3$ follows from the following result by Mader.

Theorem 5.7.8 *[677] For every $k \in \mathbb{N}$ only a finite number of $(k, 3)$-critical finite digraphs exist.* □

5.8 Connectivity Properties of Special Classes of Digraphs

In this section we describe a few results on the connectivity of various classes of digraphs introduced in Chapter 2. Some of these results will be used in other sections and chapters in this book.

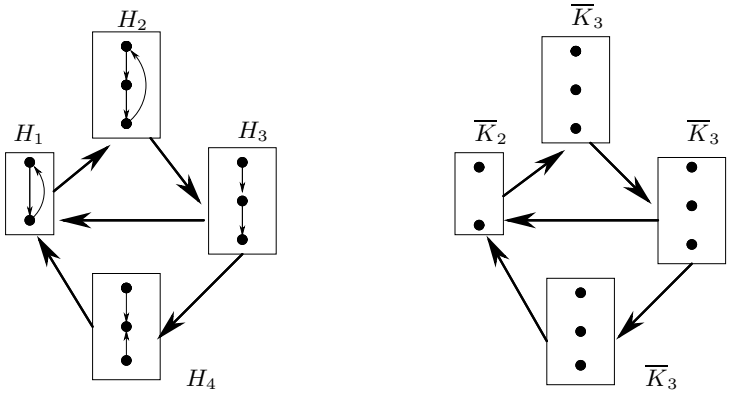


Figure 5.10 A 2-strong digraph D with decomposition $D = Q[H_1, H_2, H_3, H_4]$. Bold arcs indicate that all possible arcs are present and have the direction shown. The right figure shows the 2-strong digraph $D_0 = Q[\overline{K}_2, \overline{K}_3, \overline{K}_3, \overline{K}_3]$ obtained from D by deleting all arcs inside each H_i.

The first lemma, due to Bang-Jensen, implies that almost all minimally k-strong decomposable digraphs are subdigraphs of extensions of digraphs.

Lemma 5.8.1 *[74] Let $D = F[S_1, S_2, \ldots, S_f]$ where F is a strong digraph on $f \geq 2$ vertices and each S_i is a digraph with n_i vertices and let $D_0 = F[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_f}]$ be the digraph obtained from D by deleting every arc which lies inside some S_i[5]. Let S be a minimal (with respect to inclusion) separating set of D_0. Then S is also a separating set of D, unless each of the following holds:*

(a) $S = V(S_1) \cup V(S_2) \ldots \cup V(S_f) \setminus V(S_i)$ for some $i \in [f]$,
(b) $D\langle S_i \rangle$ is a strong digraph, and
(c) $D = C_2[S, S_i]$.

In particular, if F has at least three vertices, then D is k-strong if and only if D_0 is k-strong.

Proof: Let S be a minimal separating set of D_0 and assume S is not separating in D. It is easy to see that if x and y with $x, y \notin S$ belong to

[5] Recall that $\overline{K}_{n_i}$ is the digraph on n_i vertices and no arcs.

different sets S_i, then $D - S$ has an (x, y)-path if and only if $D_0 - S$ has such a path. Thus, since S is separating in D_0 but not in D, we must have $S = V(S_1) \cup V(S_2) \ldots \cup V(S_f) \setminus V(S_i)$ for some $i \in [f]$. Note that here we used the minimality of S to get that $S \cap S_j = \emptyset$ for some j. Now it follows trivially that $D\langle S_i \rangle$ must be a strong digraph, since $D - S$ is strong and the minimality of S implies that $D = C_2[S, S_i]$ (if some $S_j \subset S$ does not have arcs in both directions to S_i, then $S - S_j$ is also separating, contradicting the choice of S). □

See Figure 5.10 for an example illustrating the lemma.

Combining Lemma 5.8.1 with Theorem 2.7.5 we obtain:

Corollary 5.8.2 *If D is a k-strong quasi-transitive digraph with decomposition $D = Q[W_1, \ldots, W_{|Q|}]$, then the digraph $D_0 = Q[\overline{K}_{|W_1|}, \ldots, \overline{K}_{|W_{|Q|}|}]$ (that is, the digraph obtained by deleting all arcs inside each W_i) is also k-strong.* □

Another easy consequence of Lemma 5.8.1 is the following result by Bang-Jensen, Gutin and Yeo:

Theorem 5.8.3 *[95] Suppose that D is a digraph which can be decomposed as $D = F[S_1, S_2, ..., S_f]$, where $f = |V(F)| \geq 2$, and define the digraph D_0 by $D_0 = D - \cup_{i=1}^{f} \{uv : u, v \in V(S_i)\}$. Then D is strong if and only if D_0 is strong.*

Here is a useful observation on locally semicomplete digraphs due to Bang-Jensen. The proof is left as Exercise 5.25.

Lemma 5.8.4 *[66] Let D be a strong locally semicomplete digraph and let S be a minimal (not necessarily minimum) separating set of D. Then $D - S$ is connected.* □

Proposition 5.8.5 *Let $D = (V, A)$ be a k-strong digraph and let D' be obtained from D by adding a new set of vertices X and joining each vertex of X to V in such a way that $|N^+_{D'}(v)|, |N^-_{D'}(v)| \geq k + 1$ for each $v \in X$. Then D' is k-strong. If D' is not $(k + 1)$-strong, then every minimum separating set of D' is also a minimum separating set of D.*

Proof: Suppose D' is not $(k+1)$-strong and let S' be a minimum separating set of D'. Then $|S'| \leq k$. Let $S = S' \cap V(D)$. Since every vertex of $X - S'$ has an in-neighbour and an out-neighbour in $V - S$, we get that $D - S$ is not strong and hence $S = S'$ must hold and S' is also separating in D. This implies that $|S'| = k$, D' is k-strong and every minimum separating set of D' is a minimum separating set of D. □

Now we turn to a characterization of eulerian digraphs in terms of local connectivities. We need the following result due to Lovász.

Theorem 5.8.6 *[652] Let v be a vertex in a digraph D which satisfies $\lambda(v,x) \leq \lambda(x,v)$ for every $x \in V(D)$. Then $d^+(v) \leq d^-(v)$.*

Proof: Call a set $X \subseteq V - v$ **full** with respect to v if there exists an $x \in X$ so that $d^-(X) = \lambda(v,x)$ and call x a **core** of X. The following three properties of full sets are easy consequences of the submodularity of the in-degree function for sets (Corollary 5.1.2):

(a) If X and Y are full sets with cores x and y and we also have $x \in Y$, then $X \cap Y$, $X \cup Y$ are full sets with cores x, y, respectively.
(b) For every $x \neq v$ there exists a full set T_x with core x so that for every full set X containing x we have $T_x \subseteq X$ (hence T_x is the unique minimal full set containing x).
(c) Observation (b) implies that every $x \in V - v$ is contained in a unique maximal full set. Let $X_1, X_2, \ldots, X_k$ be the maximal full sets with cores $x_1, x_2, \ldots, x_k$, respectively. Then $x_i \notin X_j$ whenever $i \neq j$.

Now we are ready to prove the theorem. Let $X_1, X_2, \ldots, X_k$ be the maximal full sets and let x_i be a core of X_i. Furthermore, let $V_i = X_i - \bigcup_{j \neq i} X_j$. Observation (c) implies that $x_i \in V_i$ for every $i \in [k]$. Thus $d^+(V_i) \geq \lambda(x_i, v) \geq \lambda(v, x_i) = d^-(X_i)$ by the assumption of the theorem. Hence,

$$\sum_{i=1}^{k} d^+(V_i) \geq \sum_{i=1}^{k} d^-(X_i). \tag{5.21}$$

By the remark in the beginning of (b), $\bigcup_{i=1}^{k} X_i = V - v$. Hence every arc contributing to the left-hand side of (5.21) either enters v or some X_j, $j \neq i$. Each such edge is counted precisely once on the left-hand side of (5.21). On the other hand, every arc leaving v or entering a set X_j from somewhere other than v is counted at least once on the right-hand side of (5.21). Thus,

$$0 \leq \sum_{i=1}^{k} d^+(V_i) - \sum_{i=1}^{k} d^-(X_i) \leq d^-(v) - d^+(v),$$

which proves the theorem. □

Corollary 5.8.7 *[652] A digraph $D = (V, A)$ is eulerian if and only if $\lambda(x,y) = \lambda(y,x)$ for every pair of vertices $x, y \in V$.*

Proof: If $D = (V, A)$ is eulerian, then, by Corollary 1.7.3, for every nonempty proper subset $X \subset V$ we have $d^+(X) = d^-(X)$. By Menger's theorem (Theorem 5.4.1) this implies that $\lambda(x,y) = \lambda(y,x)$ for every pair of vertices $x, y \in V$. The converse direction follows by applying Theorem 5.8.6 to D and the converse of D. □

5.9 Disjoint X-Paths in Digraphs

A result of Gallai [385] states that if X is a subset of the vertices of an undirected graph $G = (V, E)$, then there exist k disjoint paths each of which connects two distinct vertices in X if and only if $|S| + \sum \frac{1}{2}\lfloor |C \cap X| \rfloor \geq k$ holds for every $S \subset V$, where the sum is taken over connected components C of $G - S$.

Gallai's result does not generalize to digraphs, no matter whether we replace connected component by strong component or by connected component of the underlying graph: in the first case the condition will not be necessary and in the later it will not be sufficient (take, for example, the digraph consisting of three vertices x_1, x_2, s and arcs x_1s, x_2s) and $X = \{x_1, x_2\}$).

Kriesell [627] found a way to refine the definition of vertex separation so that a necessary and sufficient condition can indeed be found. To state the result we need some definitions. Let $D = (V, A)$ be a digraph and let $X \subset V$ a subset of its vertices. An **X-path** is a path P whose initial and terminal vertex are distinct vertices of X and all other vertices of P belong to $V(D) - X$. We denote by $A^-(X)$ ($A^+(X)$) the set of all arcs in D whose tail (head) belongs to X (note that arcs inside X contribute to both sets).

Theorem 5.9.1 *[627] Let k be an integer, $D = (V, A)$ a digraph and $X \subseteq V$. There exist k pairwise disjoint X-paths if and only if*

$$|S \cap T| + \sum_{C \in \mathcal{C}(G)} \lfloor |C \cap (X \cup S \cup T)|/2 \rfloor \geq k \tag{5.22}$$

for all $S, T \in V$ with $S \cap X = T \cap X$. Here $\mathcal{C}(G)$ denotes the set of connected components in the graph $G = UG(D - (A^-(S) \cup A^+(T)))$. □

We will not give the proof here. Kriesell first shows that disjoint X-paths in D correspond, in a natural way, to matchings[6] in the digraph $H_D(X)$ which we obtain by performing the vertex splitting procedure from Section 4.2.4 on D and then contracting each of the arcs x_tx_s, $x \in X$ in D_{ST}. Now one can apply Tutte's characterization for the existence of a matching of a given cardinality in a graph and translate the characterization back to the desired characterization for the existence of k disjoint X-paths in D.

5.10 Exercises

5.1. **Submodularity of $|N^-|$ and $|N^+|$.** Prove Proposition 5.1.3.

5.2. Prove Lemma 5.2.1.

[6] A **matching** in a digraph D is a collection of arcs which have no vertices in common, i.e., the corresponding edges form a matching in $UG(D)$.

5.3. (−) Prove Corollary 5.3.3.

5.4. **Complexity of converting between a directed multigraph and its network representation.** Show that given a directed multigraph D one can construct its network representation $\mathcal{N}(D)$ in polynomial time. Show that converting in the other direction cannot always be done in a time which is polynomial in the size of the network representation. Hint: recall that we assume that capacities are represented as binary numbers.

5.5. Show that every k-regular tournament is k-arc-strong.

5.6. (−) Prove that every eulerian directed multigraph is strong.

5.7. Let D be a digraph, let s be a vertex of D and let k be a natural number. Suppose that $\min\{\lambda(s,v), \lambda(v,s)\} \geq k$ for every vertex $v \in V(D) - s$. Prove that $\lambda(D) \geq k$.

5.8. (−) **Vertex-strong connectivity of planar digraphs**. In a planar undirected graph G on n vertices and m edges we always have $m \leq 3n - 6$ by Euler's formula (see Corollary 2.12.3). Conclude that no planar digraph is 6-strong.

5.9. (−) Let D be a k-strong digraph and let a be an arbitrary arc of D. Let D' be obtained from D by reversing a. Prove that D' is $(k-1)$-strong.

5.10. **Connectivity of powers of cycles**. Recall that the kth power of a cycle $C = v_1 \dots v_n v_1$ is the digraph with vertex set $\{v_1, \dots, v_n\}$ and arc set $\{v_i v_j : i+1 \leq j \leq i+k, i \in [n]\}$. Prove that the kth power of a cycle on $n \geq k+1$ vertices is k-strong.

5.11. (−) For every natural number k describe a k-strong digraph D for which reversing any arc of D results in a digraph with vertex-strong connectivity less than k.

5.12. (+) **Finding k arc-disjoint (x,y)-paths of minimum total weight.** Let $D = (V, A, w)$ be a directed multigraph with weights on the arcs, let $x, y \in V$ be distinct vertices and let k be a natural number. Describe a polynomial algorithm which either finds a minimum weight collection of k arc-disjoint (x,y)-paths, or demonstrates that D does not have k arc-disjoint (x,y)-paths. Hint: use flows. Argue that you can find k internally disjoint (x,y)-paths of minimum total weight using a similar approach.

5.13. (+) **Minimum augmentations to ensure k arc-disjoint (s,t)-paths.** Let $D = (V, A, w)$ be a directed multigraph, let s, t be special vertices of D and let k be a natural number such that D does not have k arc-disjoint (s,t)-paths. Prove that it is possible to augment D optimally by adding new arcs so that the new directed multigraph has k arc-disjoint (s,t)-paths and all new arcs go from s to t. Now consider the same problem when there are weights on the arcs. Devise an algorithm to find the cheapest set of new arcs whose addition to D gives a directed multigraph with k arc-disjoint (s,t)-paths. Hint: use min cost flows.

5.14. **Relation between Menger's theorem and the Max-Flow Min-Cut theorem**. Prove that Menger's theorem implies the max-flow min-cut theorem for network in which all capacities are integer valued.

5.15. **Refining Menger's theorem**. Let D be a k-strong directed multigraph. Let $x_1, x_2, \ldots, x_r, y_1, y_2, \ldots, y_s$ be distinct vertices of D and let $a_1, a_2, \ldots, a_r$, $b_1, b_2, \ldots, b_s$ be natural numbers such that

$$\sum_{i=1}^{r} a_i = \sum_{j=1}^{s} b_j = k.$$

Prove that D contains k internally disjoint paths $P_1, P_2, \ldots, P_k$ with the property that precisely a_i (b_j) of these start at x_i (end at y_j). Argue that the analogous statement concerning arc-disjoint paths is true if we replace vertex-strong connectivity by arc-strong connectivity.

5.16. **Refining Menger's theorem for undirected graphs**. Prove the analogous statement of Exercise 5.15 for undirected graphs.

5.17. **Menger's theorem for sets of vertices**. Let D be k-strong and let X, Y be distinct subsets of $V(D)$. Prove that D contains k internally disjoint paths which start in X and end in Y and have only their starting (ending) vertex in X (Y).

5.18. **(+) Ear decomposition in linear time**. Supply the algorithmic details missing in the proof of Corollary 5.3.7. In particular, describe how to store the arcs in such a way that the ear decomposition can be found in linear time.

5.19. **(+) Strong orientations of mixed multigraphs in linear time**. Give an $O(n+m)$ algorithm for finding a strong orientation of a mixed multigraph or a proof that no such orientation exists (Chung, Garey and Tarjan [218]).

5.20. **(+) Cycle subdigraphs containing specified arcs**. Prove the following. Suppose D is k-strong (respectively, k-arc-strong) and $e_1, e_2, \ldots, e_k$ are arcs of D such that no two arcs have a common head or tail. Then D has a cycle subgraph (respectively, a collection of arc-disjoint cycles) $\mathcal{F} = \{C_1, \ldots, C_r\}$, $1 \leq r \leq k$, such that each arc e_i is an arc of precisely one of the cycles in $\mathcal{F}$. Hint: add two new vertices s, t, connect these appropriately to D and then apply Menger's theorem to s and t.

5.21. Prove the following: Every s-regular round digraph has strong vertex- and arc-connectivity equal to s (Ayoub and Frisch [54]).

5.22. **Connectivity of complete biorientations of undirected graphs.** Let G be a k-connected undirected graph for some $k \geq 1$ and let D be the complete biorientation of G. Prove that for every arc xy of D the digraph $D - xy$ is k-strong if and only if $D - \{xy, yx\}$ is k-strong.

5.23. **Minimal k-out-critical sets are strongly connected**. Prove that if D is a directed multigraph and X is a minimal k-out-critical set, then the directed multigraph $D\langle X\rangle$ is strongly connected.

5.24. Derive Corollary 5.7.4 from Theorem 5.7.2.

5.25. **Removing a minimal separating set from a locally semicomplete digraph**. Prove Lemma 5.8.4.

5.26. **Large 3-strong tournaments with every vertex critical**. Prove that every tournament in the class $\mathcal{T}$ from Figure 5.9 is 3-strong and that every vertex different from x, y is critical.

5.27. (−) Let D be a k-arc-strong semicomplete digraph on at least $2k+2$ vertices. Prove that there exists an arc a of D such that $D-a$ is k-arc-strong. Hint: prove that D cannot be minimally k-arc-strong.

5.28. (−) Describe a polynomial algorithm which, given a directed multigraph D, decides whether $\lambda(D) = \delta^0(D)$.

6. Hamiltonian, Longest and Vertex-Cheapest Paths and Cycles

In this chapter we will consider the hamiltonian path and cycle problems for digraphs as well as some related problems such as the longest or vertex-heaviest path and cycle problems. We describe and prove a number of results in the area as well as formulate several open questions.

In this edition of the book, we do not discuss some powerful necessary conditions, due to Gutin and Yeo, for a digraph to be hamiltonian (see [91, 476]). These conditions are generalizations of the simple conditions that every hamiltonian digraph is strong and contains a cycle factor.

Section 6.1 deals with the time complexity issues. In Section 6.2, we obtain necessary and sufficient conditions by Bang-Jensen for a path-mergeable digraph to be hamiltonian. Since locally in-semicomplete and out-semicomplete digraphs are proper subclasses (see Proposition 2.9.1) of path-mergeable digraphs, we may use these conditions, in Section 6.3, to derive a characterization of hamiltonian locally in-semicomplete and out-semicomplete digraphs. As corollaries, we obtain the corresponding results for locally semicomplete digraphs. Digraphs with restricted degrees are considered in Section 6.4 where a number of degree-related sufficient conditions for a digraph to be hamiltonian are described. In that section, we also consider a powerful proof technique, called multi-insertion, that can be applied to prove many theorems on hamiltonian digraphs. Oriented graphs with restricted semi-degrees are considered in Section 6.5, where six conjectures and some supporting results are presented.

A number of papers were devoted to studying the structure of longest cycles and paths of semicomplete multipartite digraphs. In Section 6.6, we consider the most important results obtained in this area so far including some important results by Yeo. The proofs in that section provide further illustrations of the multi-insertion technique.

Sections 6.7 and 6.8 are devoted to quasi-transitive digraphs. We present two interesting methods to tackle the hamiltonian path and cycle problems, and the vertex-heaviest path and cycle problems, respectively, in this class of digraphs. The second method by Bang-Jensen, Gutin and Yeo allows one to find vertex-heaviest paths and cycles in vertex-weighted quasi-transitive digraphs in polynomial time (where the weights are on the vertices). Since

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_6,

the weights can be positive and negative, the results can be formulated (and are formulated) for vertex-cheapest paths and cycles.

The last section is devoted to results and open problems on hamiltonian paths and cycles in some classes of digraphs not considered in the previous sections. For additional information on hamiltonian and traceable digraphs, see, e.g., the surveys [85, 90] by Bang-Jensen and Gutin, [169] by Bondy, [457] by Gutin and [890, 888] by Volkmann.

6.1 Complexity

For arbitrary digraphs the hamiltonian path and hamiltonian cycle problems are very difficult and both are $\mathcal{NP}$-complete (see, e.g., the book [393] by Garey and Johnson). For convenience of later referencing we state these results as theorems.

Theorem 6.1.1 *The problem to decide whether a given digraph has a hamiltonian cycle is $\mathcal{NP}$-complete.* □

Theorem 6.1.2 *The problem to decide whether a given digraph has a hamiltonian path is $\mathcal{NP}$-complete.* □

It is worthwhile mentioning that the hamiltonian cycle and path problems are $\mathcal{NP}$-complete even for some special classes of digraphs. Garey, Johnson and Tarjan showed [395] that the problem remains $\mathcal{NP}$-complete even for planar 3-regular digraphs. We prove the following result due to A. Yeo (private communication, 2003).

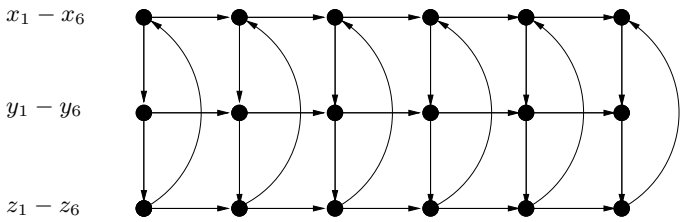


Figure 6.1 The gadget $H(x, y, z)$. The vertices are ordered from the left to the right and labelled as indicated in the left part of the figure.

Theorem 6.1.3 *It is $\mathcal{NP}$-complete to decide whether a 2-regular digraph D has a hamiltonian cycle.*

Proof: We will reduce 3-SAT (see Section 18.3) to the Hamilton cycle problem for 2-regular digraphs. Consider the following digraph $H(x, y, z)$: $V(H(x, y, z)) = \{x_i, y_i, z_i : \ i = 1, 2, 3, 4, 5, 6\}$,

$$A(H(x,y,z)) = \{x_iy_i, y_iz_i, z_ix_i : \ i = 1,2,3,4,5,6\} \cup \\ \{x_jx_{j+1}, y_jy_{j+1}, z_jz_{j+1} : \ j = 1,2,3,4,5\}$$

(see Figure 6.1). It is easy to verify that the digraph $H(x,y,z)$ has the following properties:

(i) Every Hamilton path of $H(x,y,z)$ starting at x_1 (y_1, z_1, respectively) terminates at x_6 (y_6, z_6, respectively).

(ii) Let $P \cup Q$ be a 2-path factor of $H(x,y,z)$ such that the path P starts at x_1 and the path Q starts at y_1. Then P terminates at x_6 and Q at y_6. Similarly for the pairs x_1, z_1 and y_1, z_1.

(iii) Let $P \cup Q \cup R$ be a 3-path factor of $H(x,y,z)$ such that the paths P, Q and R start at x_1, y_1 and z_1, respectively. Then P, Q and R terminate at x_6, y_6 and z_6, respectively.

Consider an instance $\mathcal{I}$ of 3-SAT with variables $v^1, \ldots, v^k$ and clauses $C_1, \ldots, C_p$. We may assume that every variable and its negation appear in $\mathcal{I}$ as literals (if needed, add to $\mathcal{I}$ clauses containing variables together with their negations). Construct a digraph D as follows: start from a disjoint union $U = H_1 \cup H_2 \cup \ldots \cup H_p$, where $H_i = H(\alpha, \beta, \gamma)$, α, β and γ are literals in C_i, $i \in [p]$. Since the same literals can occur in different clauses, we denote the vertices of $H_i = H(\alpha, \beta, \gamma)$ by $\alpha_j(H_i)$, $\beta_j(H_i)$ and $\gamma_j(H_i)$, $j = 1,2,3,4,5,6$.

For each $i \in [k]$, let $S_i = \{j_1^i, j_2^i, \ldots, j_{a(i)}^i\}$ $(j_1^i < j_2^i < \ldots < j_{a(i)}^i)$ be a set defined as follows: $l \in S_i$ if and only if C_l contains v^i as a literal. Similarly, for each $i = 1,2,\ldots,k$, let $S_i' = \{q_1^i, q_2^i, \ldots, q_{b(i)}^i\}$ $(q_1^i < q_2^i < \ldots < q_{b(i)}^i)$ be a set such that $l \in S_i'$ if and only if C_l contains $\overline{v^i}$ as a literal.

Now we finish constructing D. Add to U vertices $u_1, \underline{w_1}, u_2, w_2, \ldots, u_k, w_k$. Each u_i dominates the vertex $v_1^i(H_{j_1^i})$ and the vertex $\overline{v_1^i}(H_{q_1^i})$, $i \in [k]$. Each vertex $v_6^i(H_{j_l^i})$ dominates $v_1^i(H_{j_{l+1}^i})$, $l \in [a(i)-1]$, and each vertex $\overline{v_6^i}(H_{q_l^i})$ dominates $\overline{v_1^i}(H_{q_{l+1}^i})$, $l \in [b(i)-1]$. The vertex w_i is dominated by two vertices, $v_6^i(H_{j_{a(i)}^i})$ and $\overline{v_6^i}(H_{q_{b(i)}^i})$, for every $i \in [k]$. Finally, $w_i \to \{u_{i-1}, u_{i+1}\}$ for every $i \in [k]$, where $u_0 = u_k$, $u_{k+1} = u_1$.

It is easy to verify that D is 2-regular. Consider a truth assignment t in which $v^i = 1$. We show how to construct the fragment[1] of a Hamilton cycle Z in D corresponding to v^i. The fragment contains the arc from u_i to v_1^i in $H_{j_1^i}$, the arcs $v_6^i(H_{j_l^i})v_1^i(H_{j_{l+1}^i})$, $l \in [a(i)-1]$, the arc from u_6^i in $H_{j_{a(1)}^i}$ to w_i and the arc w_iu_{i+1}. If $\overline{v^i} = \underline{1}$ in t, then we can similarly construct the fragment of Z corresponding to $\overline{v^i}$ (we use $H_{q_1^i}, H_{q_2^i}, \ldots, H_{q_{b(1)}^i}$ instead of $H_{j_1^i}, H_{j_2^i}, \ldots, H_{j_{a(1)}^i}$).

Since every clause is satisfied by t, the cycle Z uses vertices from each digraph in the disjoint union $H_1 \cup H_2 \cup \ldots \cup H_p$. By the properties (i), (ii) and (iii) of $H(x,y,z)$ above, if s $(1 \le s \le 3)$ literals are satisfied in a clause C_j by

[1] Each fragment is not a path, but a collection of disjoint arcs.

t, all vertices of the corresponding digraph H_j can be used in Z due to the existence of an appropriate s-path factor in H_j. Thus, Z is indeed Hamilton.

Similarly, from a Hamilton cycle Z of D one can construct a truth assignment t satisfying $\mathcal{I}$ by finding which literals in t are to be assigned 1. □

It follows easily from Theorems 6.1.1 and 6.1.2 that the longest path and cycle problems are $\mathcal{NP}$-hard as optimization problems for arbitrary digraphs. This is also true for several special classes of digraphs. However, for some important special classes of digraphs these problems are polynomial time solvable. We will discuss many such classes in the chapter.

One such important example is as follows. Johnson, Robertson, Seymour and Thomas [573] proved the following theorem for directed tree-width. By Lemma 2.13.9, this theorem holds also for directed path-width and DAG-width (see Section 2.13 for definitions of directed width parameters).

Theorem 6.1.4 *The Hamilton cycle and path problems are polynomial-time solvable for digraphs of bounded directed tree-width (DAG-width, directed path-width, respectively).* □

Many sufficient conditions for the existence of a Hamilton cycle can be transformed into sufficient conditions for the existence of a Hamilton path using the following simple observation.

Proposition 6.1.5 *A digraph D has a Hamilton path if and only if the digraph D^*, obtained from D by adding a new vertex x^* such that x^* dominates every vertex of D and is dominated by every vertex of D, is hamiltonian.* □

6.2 Hamilton Paths and Cycles in Path-Mergeable Digraphs

The class of path-mergeable digraphs was introduced in Section 2.8, where some of its properties were studied. In this section, we prove a characterization of hamiltonian path-mergeable digraphs due to Bang-Jensen [72].

We begin with a simple lemma which forms the basis for the proof of Theorem 6.2.2. For a cycle C, a ***C*-bypass** is a path of length at least two with both end-vertices on C and no other vertices on C.

Lemma 6.2.1 *[72] Let D be a path-mergeable digraph and let C be a cycle in D. If D has a C-bypass P, then there exists a cycle in D containing precisely the vertices $V(C) \cup V(P)$.*

Proof: Let P be an (x, y)-path. Then the paths P and $C[x, y]$ can be merged into one (x, y)-path R, which together with $C[y, x]$ forms the desired cycle. □

Theorem 6.2.2 *[72] A path-mergeable digraph D of order $n \geq 2$ is hamiltonian if and only if D is strong and $UG(D)$ is 2-connected.*

Proof: 'Only if' is obvious; we prove 'if'. Suppose that D is strong, $UG(D)$ is 2-connected and D is not hamiltonian. Let $C = u_1u_2 \ldots u_pu_1$ be a longest cycle in D. Observe that, by Lemma 6.2.1, there is no C-bypass. For each $i \in [p]$ let X_i (respectively, Y_i) be the set of vertices of $D - V(C)$ that can be reached from u_i (respectively, from which u_i can be reached) by a path in $D - (V(C) - u_i)$. Since D is strong,

$$X_1 \cup \ldots \cup X_p = Y_1 \cup \ldots \cup Y_p = V(D) - V(C).$$

Since there is no C-bypass, every path starting at a vertex in X_i and ending at a vertex in C must end at u_i. Thus, $X_i \subseteq Y_i$. Similarly, $Y_i \subseteq X_i$ and, hence, $X_i = Y_i$. Since there is no C-bypass, the sets X_i are disjoint. Since we assumed that D is not hamiltonian, at least one of these sets, say X_1, is non-empty. Since $UG(D)$ is 2-connected, there is an arc with one end-vertex in X_1 and the other in $V(D) - (X_1 \cup u_1)$, and no matter what its orientation is, this implies that there is a C-bypass, a contradiction. □

Using the proof of this theorem, Lemma 6.2.1 and Proposition 2.8.3, it is not difficult to show the following (Exercise 6.1):

Corollary 6.2.3 *[72] There is an $O(nm)$ algorithm to decide whether a given strong path-mergeable digraph has a hamiltonian cycle and find one if it exists.* □

Clearly, Theorem 6.2.2 and Corollary 6.2.3 imply an obvious characterization of longest cycles in path-mergeable digraphs and a polynomial algorithm to find a longest cycle. Neither a characterization nor the complexity of the hamiltonian path problem for path-mergeable digraphs is currently known. The following problem was posed by Bang-Jensen and Gutin:

Problem 6.2.4 *[89] Characterize traceable path-mergeable digraphs. Is there a polynomial algorithm to decide whether a path-mergeable digraph is traceable?*

For a related result, see Proposition 7.3.3. This result may be considered as a characterization of traceable path-mergeable digraphs. However, this characterization seems of not much value from the complexity point of view.

6.3 Hamilton Paths and Cycles in Locally In-Semicomplete Digraphs

According to Proposition 2.9.1, every locally in-semicomplete digraph is path-mergeable. By Exercise 6.2, every strong locally in-semicomplete digraph has

a 2-connected underlying graph. Thus, Theorem 6.2.2 implies the following characterization of hamiltonian locally in-semicomplete digraphs[2].

Theorem 6.3.1 *[105] A locally in-semicomplete digraph D of order $n \geq 2$ is hamiltonian if and only if D is strong.* □

This theorem generalizes Camion's theorem on strong tournaments (Corollary 1.5.2). Bang-Jensen and Hell [101] showed that for the class of locally in-semicomplete digraphs Corollary 6.2.3 can be improved to the following result.

Theorem 6.3.2 *[101] There is an $O(m + n \log n)$ algorithm for finding a hamiltonian cycle in a strong locally in-semicomplete digraph.* □

In Section 6.2, we remarked that the Hamilton path problem for path-mergeable digraphs is unsolved so far. For a subclass of this class, locally in-semicomplete digraphs, an elegant characterization, due to Bang-Jensen, Huang and Prisner, exists.

Theorem 6.3.3 *[105] A locally in-semicomplete digraph is traceable if and only if it contains an in-branching.*

Proof: Since a Hamilton path is an in-branching, it suffices to show that every locally in-semicomplete digraph D with an in-branching T is traceable. We prove this claim by induction on the number b of vertices of T of in-degree zero.

For $b = 1$, the claim is trivial. Let $b \geq 2$. Consider a pair of vertices x, y of in-degree zero in T. By the definition of an in-branching there is a vertex z in T such that T contains both (x, z)-path P and (y, z)-path Q. Assume that the only common vertex of P and Q is z.

By Proposition 2.9.2, there is a path R in D that starts at x or y and terminates at z and $V(R) = V(P) \cup V(Q)$. Using this path, we may replace T with an in-branching with $b - 1$ vertices of in-degree zero and apply the induction hypothesis of the claim. □

Clearly, Theorem 6.3.3 implies that a locally out-semicomplete digraph is traceable if and only if it contains an out-branching. By Proposition 1.7.1, we have the following:

Corollary 6.3.4 *A locally in-semicomplete digraph is traceable if and only if it contains only one terminal strong component.* □

Using Corollary 6.3.4, Bang-Jensen and Hell [101] proved the following:

[2] Actually, this characterization, as well as the other results of this section, were originally proved only for oriented graphs. However, as can be seen from Exercises 2.29 and 2.30, the results for oriented graphs immediately imply the results of this section.

Theorem 6.3.5 *A longest path in a locally in-semicomplete digraph D can be found in time $O(m + n \log n)$.* □

Corollary 6.3.4 and Lemma 2.9.3 imply the following:

Corollary 6.3.6 (Bang-Jensen) *[66] A locally semicomplete digraph has a hamiltonian path if and only if it is connected.* □

Notice that there is a nice direct proof of this corollary (using Proposition 2.9.2), which is analogous to the classical proof of Rédei's theorem displayed in procedure HamPathTour in Section 18.1. See Exercise 6.4.

6.4 Hamilton Cycles and Paths in Degree-Constrained Digraphs

In Subsection 6.4.1 we formulate certain sufficient degree-constrained conditions for hamiltonicity of digraphs. Several of these conditions do not follow from the others, i.e., there are certain digraphs that can be proved to be hamiltonian using some condition but none of the others. (The reader will be asked to show this in the exercises.)

In Subsection 6.4.3 we provide proofs to some of these conditions to illustrate the power of the **multi-insertion technique**. (This technique can be traced back to Ainouche [16] for undirected graphs and to Bang-Jensen [70] for digraphs, see also the paper [93] by Bang-Jensen, Gutin and Huang). The technique itself is introduced in Subsection 6.4.2. The strength of the multi-insertion technique lies in the fact that we can prove the existence of a hamiltonian cycle without actually exhibiting it. Moreover, our hamiltonian cycles may have quite a complicated structure. For example, compare the hamiltonian cycles in the proof of Theorem 6.4.1 to the hamiltonian paths constructed in the inductive proof of Theorem 1.4.2. The multi-insertion technique is used in some other parts of this book, see e.g., Section 6.6.

Let x, y be a pair of distinct vertices in a digraph D. The pair $\{x, y\}$ is **dominated by a vertex** z if $z \to x$ and $z \to y$; in this case we say that the pair $\{x, y\}$ is **dominated**. Likewise, $\{x, y\}$ **dominates a vertex** z if $x \to z$ and $y \to z$; we call the pair $\{x, y\}$ **dominating**.

6.4.1 Sufficient Conditions

Considering the converse digraph and using Theorem 6.3.1, we see that a locally out-semicomplete digraph is hamiltonian if and only if it is strong. This can be generalized as follows. We prove Theorem 6.4.1 in Subsection 6.4.3.

Theorem 6.4.1 *[94] Let D be a strong digraph of order $n \geq 2$. Suppose that, for every dominated pair of non-adjacent vertices $\{x, y\}$, either $d(x) \geq n$ and $d(y) \geq n-1$ or $d(x) \geq n-1$ and $d(y) \geq n$. Then D is hamiltonian.* □

The following example shows the sharpness of the conditions of Theorem 6.4.1 (and Theorem 6.4.5), see Figure 6.2. Let G and H be two disjoint transitive tournaments such that $|V(G)| \geq 2, |V(H)| \geq 2$. Let w be the vertex of out-degree 0 in G and w' the vertex of in-degree 0 in H. Form a new digraph by identifying w and w' to one vertex z. Add four new vertices x, y, u, v and the arcs $\{xv, yv, ux, uy\} \cup \{xz, zx, yz, zy\} \cup \{rg : r \in \{x, y, v\}, g \in V(G) - w\} \cup \{hs : h \in V(H) - w', s \in \{u, x, y\}\}$. Denote the resulting digraph by Q_n, where n is the order of Q_n. It is easy to check that Q_n is strong and non-hamiltonian (Exercise 6.7). Also, x, y is the only pair of non-adjacent vertices which is dominating (dominated, respectively). An easy computation shows that

$$d(x) = d(y) = n - 1 = d^+(x) + d^-(y) = d^-(x) + d^+(y).$$

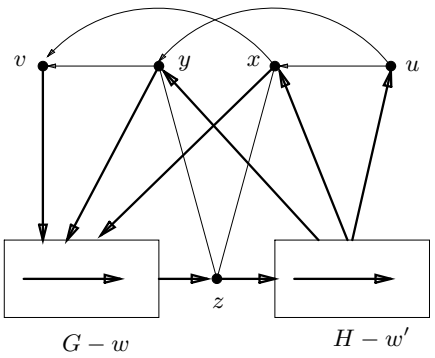


Figure 6.2 The digraph Q_n. The two unoriented edges denote 2-cycles.

Combining Theorem 6.4.1 with Proposition 6.1.5 one can obtain sufficient conditions for a digraph to be traceable (see also Exercise 6.6). Theorem 6.4.1 also has the following immediate corollaries.

Corollary 6.4.2 (Ghouila-Houri) *[403] If the degree of every vertex in a strong digraph D of order n is at least n, then D is hamiltonian.* □

Corollary 6.4.3 *Let D be a digraph of order n. If the minimum semi-degree of D, $\delta^0(D) \geq n/2$, then D is hamiltonian.* □

It turns out that even a slight relaxation of Corollary 6.4.3 brings in non-hamiltonian digraphs. In particular, Darbinyan [245] proved the following:

Proposition 6.4.4 *Let D be a digraph of even order $n \geq 4$ such that the degree of every vertex of D is at least $n-1$ and $\delta^0(D) \geq n/2 - 1$. Then either D is hamiltonian or D belongs to a non-empty finite family of non-hamiltonian digraphs.* □

By Theorem 6.3.1, a locally semicomplete digraph is hamiltonian if and only if it is strong [66]. This result was generalized by Bang-Jensen, Gutin and Li [94] as follows.

Theorem 6.4.5 *Let D be a strong digraph of order n. Suppose that D satisfies $\min\{d^+(x) + d^-(y),\ d^-(x) + d^+(y)\} \geq n$ for every pair of dominating non-adjacent and every pair of dominated non-adjacent vertices $\{x, y\}$. Then D is hamiltonian.*

We prove this theorem in Subsection 6.4.3. Theorem 6.4.5 implies Corollary 6.4.3 as well as the following theorem by Woodall [911]:

Corollary 6.4.6 *Let D be a digraph of order $n \geq 2$. If $d^+(x) + d^-(y) \geq n$ for all pairs of vertices x and y such that there is no arc from x to y, then D is hamiltonian.* □

The following theorem generalizes Corollaries 6.4.2, 6.4.3 and 6.4.6. The inequality of Theorem 6.4.7 is best possible: Consider $\overleftrightarrow{K}_{n-2}$ $(n \geq 5)$ and fix a vertex u in this digraph. Construct the digraph H_n by adding to $\overleftrightarrow{K}_{n-2}$ a pair v, w of vertices such that both v and w dominate every vertex in $\overleftrightarrow{K}_{n-2}$ and are dominated by only u, see Figure 6.3. It is easy to see that H_n is strong and non-hamiltonian ($H_n - u$ is not traceable). However, v, w is the only pair of non-adjacent vertices in H_n and $d(v) + d(w) = 2n - 2$.

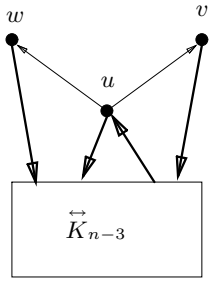


Figure 6.3 The digraph H_n.

Theorem 6.4.7 (Meyniel's theorem) *[698] Let D be a strong digraph of order $n \geq 2$. If $d(x) + d(y) \geq 2n - 1$ for all pairs of non-adjacent vertices in D, then D is hamiltonian.* □

Short proofs of Meyniel's theorem were given by Overbeck-Larisch [735] and Bondy and Thomassen [171]. The second proof is slightly simpler than the first one and can also be found in the book [902] by West (see Theorem 8.4.38). Using Proposition 6.1.5 one can easily see that replacing $2n-1$ by $2n-3$ in Meyniel's theorem we obtain sufficient conditions for traceability. (Note that for traceability we do not require strong connectivity.) Darbinyan [248] proved that by weakening the degree condition in Meyniel's theorem only by one, we obtain a stronger result:

Theorem 6.4.8 *[248] Let D be a digraph of order $n \geq 3$. If $d(x)+d(y) \geq 2n-2$ for all pairs of non-adjacent vertices in D, then D contains a hamiltonian path in which the initial vertex dominates the terminal vertex.* □

Berman and Liu [147] extended Theorem 6.4.7 as formulated below. For a digraph D of order n, a set $M \subseteq V(D)$ is **Meyniel** if $d(x)+d(y) \geq 2n-1$ for every pair x, y of non-adjacent vertices in M. The proof of Theorem 6.4.9 in [147] is based on the multi-insertion technique.

Theorem 6.4.9 *[147] Let M be a Meyniel set of vertices of a strong digraph D of order $n \geq 2$. Then D has a cycle containing all vertices of M.* □

Another extension of Meyniel's theorem was given by Heydemann [527].

Theorem 6.4.10 *[527] Let h be a non-negative integer and let D be a strong digraph of order $n \geq 2$ such that, for every pair of non-adjacent vertices x and y, we have $d(x)+d(y) \geq 2n-2h+1$. Then D contains a cycle of length greater than or equal to $\lceil \frac{n-1}{h+1} \rceil + 1$.* □

Manoussakis [682] proved the following sufficient condition that involves triples rather than pairs of vertices. Notice that Theorem 6.4.11 does not imply either of Theorems 6.4.1, 6.4.5 and 6.4.7 [94].

Theorem 6.4.11 *[682] Suppose that a strong digraph D of order $n \geq 2$ satisfies the following conditions for every triple $x, y, z \in V(D)$ such that x and y are non-adjacent:*

(a) If there is no arc from x to z, then $d(x)+d(y)+d^+(x)+d^-(z) \geq 3n-2$.
(b) If there is no arc from z to x, then $d(x)+d(y)+d^-(x)+d^+(z) \geq 3n-2$.

Then D is hamiltonian. □

The next theorem, due to Zhao and Meng, resembles Theorems 6.4.5 and 6.4.7. However, Theorem 6.4.12 does not imply any of these theorems. The sharpness of the inequality of Theorem 6.4.12 can be seen from the digraph H_n introduced before Theorem 6.4.7.

Theorem 6.4.12 *[933] Let D be a strong digraph of order $n \geq 2$. If*

$$d^+(x) + d^+(y) + d^-(u) + d^-(v) \geq 2n - 1$$

for every pair x, y of dominating vertices and every pair u, v of dominated vertices, then D is hamiltonian. □

Theorems 6.4.5 and 6.4.12 support the following conjecture by Bang-Jensen, Gutin and Li.

Conjecture 6.4.13 *[94] Let D be a strong digraph of order $n \geq 2$. Suppose that $d(x)+d(y) \geq 2n-1$ for every pair of dominating non-adjacent and every pair of dominated non-adjacent vertices $\{x, y\}$. Then D is hamiltonian.*

Bang-Jensen, Guo and Yeo [82] proved that if we replace the degree condition $d(x) + d(y) \geq 2n - 1$ with $d(x) + d(y) \geq \frac{5}{2}n - 4$ in Conjecture 6.4.13, then D is hamiltonian. They also provided additional support for Conjecture 6.4.13 by showing that every digraph satisfying the condition of Conjecture 6.4.13 has a cycle factor.

Perhaps Conjecture 6.4.13 can even be generalized to the following which was conjectured by Bang-Jensen, Gutin and Li:

Conjecture 6.4.14 *[94] Let D be a strong digraph of order $n \geq 2$. Suppose that, for every pair of dominated non-adjacent vertices $\{x, y\}$, $d(x) + d(y) \geq 2n - 1$. Then D is hamiltonian.*

Let F be the digraph obtained from the complete digraph $\overleftrightarrow{K}_{n-3}$ by adding three new vertices $\{x, y, z\}$ and the following arcs $\{xy, yx, yz, zy, zx\} \cup \{xu, ux, yu : u \in V(\overleftrightarrow{K}_{n-3})\}$, see Figure 6.4. Clearly F is strongly connected and the underlying undirected graph of F is 2-connected. However, F is not hamiltonian as all hamiltonian paths in $F - x$ start at z, but x does not dominate z. The only pairs of non-adjacent vertices in D are z and any vertex $u \in V(\overleftrightarrow{K}_{n-3})$ and here we have $d(z) + d(u) = 2n - 2$. Thus both conjectures above would be the best possible.

One of the oldest conjectures in the area of hamiltonian digraphs is the following conjecture by Nash-Williams.

Conjecture 6.4.15 *[719, 720] Let D be a digraph of order $n \geq 3$ satisfying the following conditions:*

1. *For every positive integer k less than $(n-1)/2$, the number of vertices of out-degree less than or equal to k is less than k.*
2. *The number of vertices of out-degree less than or equal to $(n-1)/2$ is less than or equal to $(n-1)/2$.*
3. *For every positive integer k less than $(n-1)/2$, the number of vertices of in-degree less than or equal to k is less than k.*

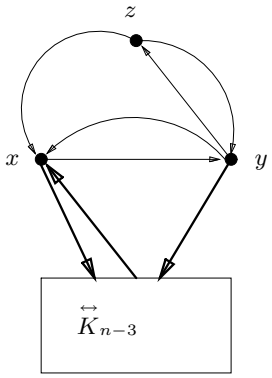


Figure 6.4 The digraph F.

4. *The number of vertices of in-degree less than or equal to* $(n-1)/2$ *is less than or equal to* $(n-1)/2$.

Then D *is hamiltonian.*

Conjecture 6.4.15 seems to be very difficult (see comments by Nash-Williams in [720, 721]). This conjecture was inspired by the corresponding theorem by Pósa [753] on undirected graphs. Pósa's result implies that the assertion of this conjecture is true at least for symmetric digraphs, i.e., digraphs D such that $xy \in A(D)$ implies $yx \in A(D)$.

One may also try to obtain digraph analogues of various other sufficient degree conditions for graphs, such as Chvátal's theorem [220], which asserts that if the degree sequence $d_1 \leq d_2 \leq \ldots \leq d_n$ of an undirected graph satisfies the condition $d_k \leq k < \frac{n}{2} \Rightarrow d_{n-k} \geq n-k$ for each k, then the graph is hamiltonian. Similarly, one may ask whether every strong digraph whose non-decreasing degree sequence $d_1 \leq d_2 \leq \ldots \leq d_n$ satisfies the following condition is hamiltonian:

$$d_k \leq 2k < n \Rightarrow d_{n-k} \geq 2(n-k),\ k = 1, 2, \ldots, n-1. \tag{6.1}$$

For a digraph D we can obtain the non-decreasing out-degree and in-degree sequences: $d_1^+ \leq d_2^+ \leq \ldots \leq d_n^+$ and $d_1^- \leq d_2^- \leq \ldots \leq d_n^-$ (orderings of vertices of D in these two sequences are usually different). Using the two sequences, one may suggest conditions similar to (6.1):

$$\begin{aligned} &d_k^+ \leq k < \frac{n}{2} \Rightarrow d_{n-k}^+ \geq n-k \text{ and} \\ &d_k^- \leq k < \frac{n}{2} \Rightarrow d_{n-k}^- \geq n-k,\ 1 \leq k \leq (n-1)/2. \end{aligned} \tag{6.2}$$

It is not difficult to construct an infinite family of non-hamiltonian strong digraphs that satisfy both (6.1) and (6.2) (Exercise 6.15). However, if we 'mix' the out-degrees with the in-degrees in (6.2), we obtain the following conjecture due to Nash-Williams:

Conjecture 6.4.16 *[721] A strong digraph D is hamiltonian, if the non-decreasing out-degree and in-degree sequences of D satisfy the following:*

$$d_k^+ \leq k < \frac{n}{2} \Rightarrow d_{n-k}^- \geq n-k \text{ and}$$
$$d_k^- \leq k < \frac{n}{2} \Rightarrow d_{n-k}^+ \geq n-k, \ 1 \leq k \leq (n-1)/2.$$

6.4.2 The Multi-Insertion Technique

Let $P = u_1u_2 \dots u_s$ be a path in a digraph D and let $Q = v_1v_2 \dots v_t$ be a path in $D-V(P)$. The path P **can be inserted into** Q if there is a subscript $i \in [t-1]$ such that $v_i \rightarrow u_1$ and $u_s \rightarrow v_{i+1}$. Indeed, in this case the path Q can be extended to a new (v_1, v_t)-path $Q[v_1, v_i]PQ[v_{i+1}, v_t]$. The path P **can be multi-inserted into** Q if there are integers $i_1 = 1 < i_2 < \ldots < i_m = s+1$ such that, for every $k = 2, 3, \dots, m$, the subpath $P[u_{i_{k-1}}, u_{i_k-1}]$ can be inserted into Q. The sequence of subpaths $P[u_{i_{k-1}}, u_{i_k-1}]$, $k = 2, \dots, m$, is a **multi-insertion partition** of P. Similar definitions can be given for the case when Q is a cycle.

The complexity of algorithms in this subsection is measured in terms of the number of queries to the adjacency matrix of a digraph. In this subsection we prove several simple results, which are very useful while applying the multi-insertion technique. Some of these results are used in this section, others will be applied in other parts of this book. The following lemma is a simple extension of a lemma by Bang-Jensen, Gutin and Li [94].

Lemma 6.4.17 *Let P be a path in D and let $Q = v_1v_2 \dots v_t$ be a path (a cycle, respectively) in $D-V(P)$. If P can be multi-inserted into Q, then there is a (v_1, v_t)-path R (a cycle, respectively) in D so that $V(R) = V(P) \cup V(Q)$. Given a multi-insertion partition of P, the path R can be found in time $O(|V(P)||V(Q)|)$.*

Proof: We consider only the case when Q is a path, as the other case (Q is a cycle) can be proved analogously. Let $P = u_1u_2 \dots u_s$. Suppose that integers $i_1 = 1 < i_2 < \ldots < i_m = s+1$ are such that the subpaths $P[u_{i_{k-1}}, u_{i_k-1}]$, $k = 2, 3, \dots, m$, form a multi-insertion partition of P.

We proceed by induction on m. If $m = 2$, then the claim is obvious, hence assume that $m \geq 3$. Let $xy \in A(Q)$ be such that the subpath $P[u_{i_1}, u_{i_2-1}]$ can be inserted between x and y on Q. Choose r as large as possible such that $u_{i_r-1} \rightarrow y$. Clearly, $P[u_{i_1}, u_{i_r-1}]$ can be inserted into Q to give a (v_1, v_t)-path

Q^*. Thus, if $r = m$, we are done. Otherwise apply the induction hypothesis to the paths $P[u_{i_r}, u_s]$ and Q^* (observe that by the choice of r none of the subpaths of the multi-insertion partition of $P[u_{i_r}, u_s]$ can be inserted between x and y in Q, and thus every such subpath can be inserted into Q^*).

If we postpone the actual construction of R till we have found a new multi-insertion partition $\mathcal{M}$ of P and all (distinct) pairs of vertices between which the subpaths of $\mathcal{M}$ can be inserted, then the complexity claim of this lemma follows easily. □

The next two corollaries due to Bang-Jensen, Gutin and Huang, respectively, Yeo can easily be proved using Lemma 6.4.17; their proofs are left as an easy exercise (Exercise 6.11).

Corollary 6.4.18 *[93] Let D be a digraph. Suppose that $P = u_1u_2 \dots u_r$ is a path in D and C is a cycle in $D - P$. Suppose that for each $i \in [r-1]$, either the arc u_iu_{i+1} or the vertex u_i can be inserted into C, and, in addition, assume that u_r can be inserted into C. Then D contains a cycle Z with the vertex set $V(P) \cup V(C)$ and Z can be constructed in time $O(|V(P)||V(C)|)$.* □

Corollary 6.4.19 *[915] Let D be a digraph. Suppose that $P = u_1u_2 \dots u_r$ is a path in D and C is a cycle in $D - P$. Suppose also that for each odd index i the arc u_iu_{i+1} can be inserted into C, and if r is odd, u_r can be inserted into C. Then D contains a cycle Z with the vertex set $V(P) \cup V(C)$ and Z can be constructed in time $O(|V(P)||V(C)|)$.* □

Corollary 6.4.20 *[93] Let D be a digraph. Suppose that C is a cycle of even length in D and Q is a cycle in $D - C$. Suppose also that for each arc uv of C either the arc uv or the vertex u can be inserted into Q. Then D contains a cycle Z with the vertex set $V(Q) \cup V(C)$ and Z can be constructed in time $O(|V(Q)||V(C)|)$.*

Proof: If there is a vertex x on C that can be inserted into Q, then apply Corollary 6.4.18 to $C[x^+, x]$ and Q. Otherwise, all the arcs of C can be inserted into Q and we can apply Corollary 6.4.19 to $C[y^+, y]$ and Q, where y is any vertex of C. □

6.4.3 Proofs of Theorems 6.4.1 and 6.4.5

The following lemma is a slight modification of a lemma by Bondy and Thomassen [171]; its proof is not difficult and is left as an exercise to the reader (Exercise 6.8).

Lemma 6.4.21 *Let $Q = v_1v_2 \dots v_t$ be a path in D and let w, w' be vertices of $V(D) - V(Q)$ (possibly $w = w'$). If there do not exist consecutive vertices v_i, v_{i+1} on Q such that $v_iw, w'v_{i+1}$ are arcs of D, then $d^-_Q(w) + d^+_Q(w') \leq t + \xi$, where $\xi = 1$ if $v_t \to w$ and 0, otherwise.* □

In the special case when $w' = w$ above, we get the following interpretation of the statement of Lemma 6.4.21.

Lemma 6.4.22 *Let $Q = v_1 v_2 \dots v_t$ be a path in D, and let $w \in V(D) - V(Q)$. If w cannot be inserted into Q, then $d_Q(w) \leq t + 1$. If, in addition, v_t does not dominate w, then $d_Q(w) \leq t$.* □

Let C be a cycle in D. Recall that an (x, y)-path P is a C-bypass if $|V(P)| \geq 3$, $x \neq y$ and $V(P) \cap V(C) = \{x, y\}$. The length of the path $C[x, y]$ is the **gap of P with respect to C**.

Proof of Theorem 6.4.1: Assume that D is non-hamiltonian and $C = x_1 x_2 \dots x_m x_1$ is a longest cycle in D. We first show that D contains a C-bypass. Assume D does not have one. Since D is strong, D must contain a cycle Z such that $|V(Z) \cap V(C)| = 1$. Without loss of generality, we may assume that $V(Z) \cap V(C) = \{x_1\}$. Let z be the successor of x_1 on Z. Since D has no C-bypass, z and x_2 are non-adjacent. Since z and x_2 are a dominated pair, $d(z) + d(x_2) \geq 2n - 1$. On the other hand, since D has no C-bypass, we have $d_{C-x_1}(z) = d_{Z-x_1}(x_2) = 0$ and $|(\{z, x_2\}, y) \cup (y, \{z, x_2\})| \leq 2$ for every $y \in V(D) - (V(C) \cup V(Z))$. Thus, $d(z) + d(x_2) \leq 2(n - 1)$; a contradiction.

Let $P = u_1 u_2 \dots u_s$ be a C-bypass ($s \geq 3$). Without loss of generality, let $u_1 = x_1$, $u_s = x_{\gamma+1}$, $0 < \gamma < m$. Suppose also that the gap γ of P is minimum among the gaps of all C-bypasses.

Since C is a longest cycle of D, $\gamma \geq 2$. Let $C' = C[x_2, x_\gamma]$, $C'' = C[x_{\gamma+1}, x_1]$, $R = D - V(C)$, and let x_j be any vertex in C' such that $x_1 \rightarrow x_j$. Let also x_k be an arbitrary vertex in C'. We first prove that

$$d_{C''}(x_j) \geq |V(C'')| + 2. \tag{6.3}$$

Since C is a longest cycle and P has the minimum gap with respect to C, u_2 is not adjacent to any vertex on C', and there is no vertex $y \in V(R) - \{u_2\}$ such that either $u_2 \rightarrow y \rightarrow x_k$ or $x_k \rightarrow y \rightarrow u_2$. Therefore,

$$d_{C'}(x_k) + d_{C'}(u_2) \leq 2(|V(C')| - 1) \tag{6.4}$$

and

$$d_R(x_k) + d_R(u_2) \leq 2(n - m - 1). \tag{6.5}$$

By the maximality of C, u_2 cannot be inserted into C'', so by Lemma 6.4.22,

$$d_{C''}(u_2) \leq |V(C'')| + 1. \tag{6.6}$$

The fact that the pair of non-adjacent vertices $\{x_j, u_2\}$ is dominated by x_1 along with (6.4), (6.5) and (6.6), implies that

$$2n - 1 \leq d(x_j) + d(u_2) \leq d_{C''}(x_j) + 2n - |V(C'')| - 3.$$

This implies (6.3).

By (6.3) and Lemma 6.4.22, x_2 can be inserted into C''. Since C is a longest cycle, it follows from Lemma 6.4.17 that there exists $\beta \in \{3, \ldots, \gamma\}$ so that the subpath $C[x_2, x_{\beta-1}]$ can be multi-inserted into C'', but $C[x_2, x_\beta]$ cannot. In particular, x_β cannot be inserted into C''. Thus, by (6.3) and Lemma 6.4.22, x_1 does not dominate x_β and $d_{C''}(x_\beta) \leq |V(C'')|$. This along with (6.4)-(6.6) gives $d(x_\beta) + d(u_2) \leq 2n - 3$. Since u_2 forms a dominated pair with x_2, we have that $d(u_2) \geq n - 1$. Hence,

$$d(x_\beta) \leq n - 2. \tag{6.7}$$

By the definition of multi-insertion, there are $\alpha \in \{2, 3, \ldots, \beta - 1\}$ and $i \in \{\gamma + 1, \ldots, m\}$ such that $x_i \to x_\alpha$ and $x_{\beta-1} \to x_{i+1}$. Observe that the pair $\{x_\beta, x_{i+1}\}$ is dominated by $x_{\beta-1}$. Thus, by (6.7) and the assumption of the theorem, either $x_\beta \to x_{i+1}$ or $x_{i+1} \to x_\beta$. If $x_\beta \to x_{i+1}$, then the path $P[x_2, x_\beta]$ can be multi-inserted into C'' which contradicts our assumption. Hence, $x_{i+1} \to x_\beta$. Considering the pair x_β, x_{i+2}, we conclude analogously that $x_{i+2} \to x_\beta$. Continuing this process, we finally conclude that $x_1 \to x_\beta$, contradicting the conclusion above that the arc $x_1 x_\beta$ does not exist. □

Proof of Theorem 6.4.5: Assume that D is not hamiltonian and $C = x_1 x_2 \ldots x_m x_1$ is a longest cycle in D. Set $R = D - V(C)$. We first prove that D has a C-bypass with 3 vertices.

Since D is strong, there is a vertex y in R and a vertex x in C such that $y \to x$. If y dominates every vertex on C, then C is not a longest cycle, since a path P from a vertex x_i on C to y such that $V(P) \cap V(C) = \{x_i\}$ together with the arc $y \to x_{i+1}$ and the path $C[x_{i+1}, x_i]$ form a longer cycle in D. Hence, either there exists a vertex $x_r \in V(C)$ such that $x_r \to y \to x_{r+1}$, in which case we have the desired bypass, or there exists a vertex $x_j \in V(C)$ so that y and x_j are non-adjacent, but $y \to x_{j+1}$. Since the pair $\{y, x_j\}$ dominates x_{j+1}, $d^+(x_j) + d^-(y) \geq n$. This implies the existence of a vertex $z \in V(D) - \{x_j, x_{j+1}, y\}$ such that $x_j \to z \to y$. Since C is a longest cycle, $z \in V(C)$. So, $B = z y x_{j+1}$ is the desired bypass.

Without loss of generality, assume that $z = x_1$ and the gap j of B with respect to C is minimum among the gaps of all C-bypasses with three vertices. Clearly, $j \geq 2$.

Let $C' = C[x_2, x_j]$ and $C'' = C[x_{j+1}, x_1]$. Since C is a longest cycle, C' cannot be multi-inserted into C''. It follows from Lemma 6.4.21 that $d^+_{C''}(x_j) + d^-_{C''}(x_2) \leq |V(C'')| + 1$. By Lemma 6.4.22 and the maximality of C, $d_{C''}(y) \leq |V(C'')| + 1$. Analogously to the way we derived (6.4) in the previous proof, we get that $d_R(y) + d^+_R(x_j) + d^-_R(x_2) \leq 2(n - m - 1)$. Clearly, $d^+_{C'}(x_j) + d^-_{C'}(x_2) \leq 2|V(C')| - 2$. Since $d_{C'}(y) = 0$, the last four inequalities imply

$$d(y) + d^+(x_j) + d^-(x_2) \leq 2n - 2. \tag{6.8}$$

Since y is adjacent to neither x_2 nor x_j, the assumption of the theorem implies that $d^+(y) + d^-(x_2) \geq n$ and $d^-(y) + d^+(x_j) \geq n$, which contradicts (6.8). □

6.5 Longest Paths and Cycles in Degree-Constrained Oriented Graphs

One may expect that for oriented graphs (i.e., digraphs with no 2-cycles) a result much stronger than Corollary 6.4.3 holds. Thomassen [854] raised the natural question of determining the minimum semi-degree that ensures a Hamilton cycle in an oriented graph. The first attempt to answer this question was made by Häggkvist [487] who proved that if $\delta^0(D) \geq (\frac{1}{2} - 2^{-18})n$, then D is hamiltonian. Häggkvist [487] also constructed the following infinite family of non-hamiltonian oriented graphs D of order n with $\delta^0(D) = (3n-5)/8$.

Let $n = 4p+3$, where p is an odd positive integer. Define a digraph D as follows: $V(D)$ is the disjoint union of four sets Y, Z, R_1 and R_2 of cardinalities $p+2, p+1, p$ and p, respectively. The arc set $A(D) = A(\vec{C}_4[R_1, Y, R_2, Z]) \cup A_{R_1} \cup A_{R_2} \cup A_{Y,Z}$, where A_{R_1} and A_{R_2} are the arcs sets of regular tournaments on vertex sets R_1 and R_2, respectively, and $A_{Y,Z}$ is the arc set of a bipartite tournament T with partite sets Y and Z in which $|d_T^+(v) - d_T^-(v)| \leq 1$ for each $v \in Y \cup Z$ (such a bipartite tournament exists by Exercise 1.27). Observe that $\delta^+(D) = d^+(r) = \frac{p-1}{2} + p + 1 = \frac{3n-5}{8}$ for each $r \in R_2$ and D has no cycle factor by Proposition 4.11.7(c).

In fact, the above construction was extended by Keevash, Kühn and Osthus [587] to prove the following:

Theorem 6.5.1 *For any integer $n \geq 3$ there is an oriented graph D of order n with minimum semi-degree $\lceil (3n-4)/8 \rceil - 1$ which does not contain a cycle factor.* □

This theorem implies the right lower bound for the minimum semi-degree that ensures hamiltonicity of an oriented graph at least for graphs of large order. Indeed, Keevash, Kühn and Osthus [587] proved the following:

Theorem 6.5.2 *There exists an integer N such that every oriented graph D of order $n \geq N$ with $\delta^0(D) \geq \lceil (3n-4)/8 \rceil$ is hamiltonian.* □

The proof of this theorem is quite involved and uses two powerful tools: Young's version of the Digraph Regularity Lemma [921] and Csaba's version of the so-called Blow-up Lemma [233].

The authors of [587] were unable to settle the following conjecture of Häggkvist [487] even for large values of n.

Conjecture 6.5.3 *Let D be an oriented graph of order n and let $\delta^0(D) + \delta^+(D) + \delta^-(D) > \frac{3n-3}{2}$. Then D is hamiltonian.*

In support of this conjecture Kelly, Kühn and Osthus [588] proved the following result.

Theorem 6.5.4 *For every $\alpha > 0$ there exists an integer $N = N(\alpha)$ such that every oriented graph D of order $n \geq N$ with $\delta^0(D)+\delta^+(D)+\delta^-(D) \geq (\frac{3}{2}+\alpha)n$ is hamiltonian.* □

Jackson conjectured that for regular oriented graphs the following strong assertion holds.

Conjecture 6.5.5 *[555] Every k-regular oriented graph of order at most $4k+1$, where $k \neq 2$, contains a Hamilton cycle.*

Jackson [555] raised two more conjectures:

Conjecture 6.5.6 *Every 2-strong oriented graph D has either a Hamilton cycle or a cycle of length at least $2\delta^-(D)+2$.*

Conjecture 6.5.7 *Every strong oriented graph D has a cycle of length at least $2\delta^0(D)+1$.*

In support of both conjectures, Jackson [555] proved that a strong oriented graph D of minimum in-degree and out-degree k contains either a Hamilton path or a path of length $2k+2$ and that an oriented graph D of minimum in-degree and out-degree k contains a path of length $2k$. The last result provides support to the following conjecture of S. Thomassé (private communication, 2002). Recall that the girth $g(D)$ of a digraph D is the length of a shortest cycle of D.

Conjecture 6.5.8 *Let D be a digraph containing a cycle and let g be the girth of D. Then D has a path of length $(g-1)\delta^+(D)$.*

Even the special case of oriented graphs (i.e., $g \geq 3$ and, thus, the lower bound is $2\delta^+(D)$) remains open.

6.6 Longest Paths and Cycles in Semicomplete Multipartite Digraphs

While both Hamilton path and Hamilton cycle problems are polynomial time solvable for semicomplete multipartite digraphs (the latter was a difficult open problem for a while and was proved by Bang-Jensen, Gutin and Yeo [97] using several deep results on cycles and paths in semicomplete multipartite digraphs, see also [917]), only a characterization of traceable semicomplete multipartite digraphs is known. In Subsection 6.6.1, we give basic results on hamiltonian and longest paths and cycles in semicomplete multipartite

digraphs. Several results of Subsection 6.6.1 are proved in Subsection 6.6.3 using the most important assertion of Subsection 6.6.2. In Subsection 6.6.4, we formulate perhaps the most important known result on cycles in semicomplete multipartite digraphs, Yeo's Irreducible Cycle Subdigraph Theorem, and prove some interesting consequences of this powerful result. Due to space limitations our treatment of hamiltonian semicomplete multipartite digraphs is certainly restricted. The reader can find more information on the topic in the survey papers [89, 90] by Bang-Jensen and Gutin, [457] by Gutin and [890, 894] by Volkmann, the theses [436, 451, 847, 905, 916], by Guo, Gutin, Tewes, Winzen and Yeo respectively and the papers cited there.

6.6.1 Basic Results

We start by considering the longest path problem for semicomplete multipartite digraphs. The following characterization by Gutin is proved in Subsection 6.6.3.

Theorem 6.6.1 *[448, 452] A semicomplete multipartite digraph D is traceable if and only if it contains a 1-path-cycle factor. One can verify whether D is traceable and find a hamiltonian path in D (if any) in time $O(n^{2.5})$.*

This theorem can be reformulated as $\mathrm{pc}(D) = 1$ if and only if $\mathrm{pcc}(D) = 1$ for a semicomplete multipartite digraph D. Using the result of Exercise 4.67, the last statement can be easily extended to the following result by Gutin:

Theorem 6.6.2 *[451] For a semicomplete multipartite digraph D, $\mathrm{pc}(D) = \mathrm{pcc}(D)$. The path covering number of D can be found in time $O(n^{2.5})$.* □

The non-complexity part of the next result by Gutin follows from Theorem 6.6.1. The complexity part is a simple consequence of Theorem 13.8.1.

Theorem 6.6.3 *[452] Let D be a semicomplete multipartite digraph of order n.*

(a) Let $\mathcal{F}$ be a 1-path-cycle subdigraph with maximum number of vertices in D. Then D contains a path P such that $V(P) = V(\mathcal{F})$.
(b) A longest path in D can be constructed in time $O(n^3)$. □

We see from Theorem 6.6.1 that the hamiltonian path problem for semicomplete multipartite digraphs turns out to be relatively simple. The hamiltonian cycle problem for this class of digraphs seems to be much more difficult. One could guess that similarly to Theorem 6.6.1, a semicomplete multipartite digraph is hamiltonian if and only if it is strong and has a cycle factor. Even though these two conditions (strong connectivity and the existence of a cycle factor) are sufficient for semicomplete bipartite digraphs and extended semicomplete digraphs (see Theorems 6.6.4 and 6.6.5), they are not sufficient

for semicomplete k-partite digraphs ($k \geq 3$) (see, e.g., an example later in this subsection). The following characterization was obtained independently by Gutin [444] and Häggkvist and Manoussakis [488].

Theorem 6.6.4 *A semicomplete bipartite digraph D is hamiltonian if and only if D is strong and contains a cycle factor. One can check whether D is hamiltonian and construct a Hamilton cycle of D (if one exists) in time $O(n^{2.5})$.* □

Some sufficient conditions for the existence of a hamiltonian cycle in a bipartite tournament are described in the survey paper [457] by Gutin.

Theorem 6.6.5 *[456] An extended semicomplete digraph D is hamiltonian if and only if D is strong and contains a cycle factor. One can check whether D is hamiltonian and construct a Hamilton cycle of D (if one exists) in time $O(n^{2.5})$.* □

These two theorems were generalized by Gutin as follows.

Theorem 6.6.6 *[447, 451] Let D be a strong semicomplete bipartite digraph. The length of a longest cycle in D is equal to the number of vertices in a cycle subdigraph of D of maximum order. One can find a longest cycle in D in time $O(n^3)$.*

Theorem 6.6.7 *[451] Let D be a strong extended semicomplete digraph and let $\mathcal{F}$ be a cycle subdigraph of D. Then D has a cycle C which contains all vertices of $\mathcal{F}$. The cycle C can be found in time $O(n^3)$. In particular, if $\mathcal{F}$ is maximum, then $V(C) = V(\mathcal{F})$, i.e., C is a longest cycle of D.*

Proofs of the last two theorems are given in Subsection 6.6.3. One can see that the statement of Theorem 6.6.7 is stronger than Theorem 6.6.6. In fact, the analogue of Theorem 6.6.7 for semicomplete bipartite digraphs does not hold [451], see Exercise 6.21. The following strengthening of Theorem 6.6.7 is proved by Bang-Jensen, Huang and Yeo [106].

Theorem 6.6.8 *Let $D = (V, A)$ be a strong extended semicomplete digraph with decomposition given by $D = S[H_1, H_2, \ldots, H_s]$, where $s = |V(S)|$ and every $V(H_i)$ is a maximal independent set in V. Let m_i, $i \in [s]$, denote the maximum number of vertices from H_i which are contained in a cycle subdigraph of D. Then every longest cycle of D contains precisely m_i vertices from each H_i, $i \in [s]$.* □

One may ask whether there is any degree of strong connectivity, which together with a cycle factor is sufficient to guarantee a hamiltonian cycle in a semicomplete multipartite digraph (or a multipartite tournament). The

answer is negative. In fact, there is no s such that every s-strong multipartite tournament with a cycle factor has a Hamilton cycle. Figure 6.5 shows a non-hamiltonian multipartite tournament T which is s-strong (s is the number of vertices in each of the sets A, B, C, D and X, Y, Z), and has a cycle factor. We leave it to the reader to verify that there is no Hamilton cycle in T (Exercise 6.20).

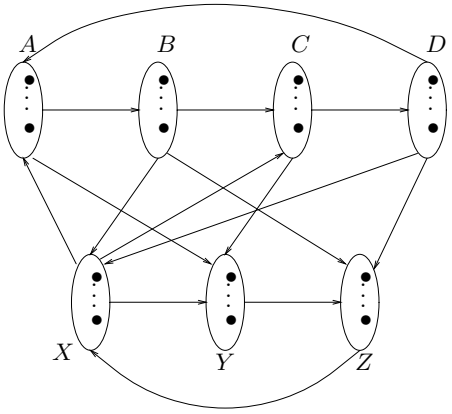


Figure 6.5 An s-strong non-hamiltonian multipartite tournament T with a cycle factor. Each of the sets A, B, C, D and X, Y, Z induces an independent set with exactly s vertices. All arcs between two sets have the direction shown.

We conclude the description of basic results on hamiltonian semicomplete digraphs by the following important result which we mentioned above.

Theorem 6.6.9 (Bang-Jensen, Gutin and Yeo) *[97] One can verify whether a semicomplete multipartite digraph D has a hamiltonian cycle and find one (if it exists) in time $O(n^7)$.* □

Yeo [917] proved that the problem can be solved in time $O(n^5)$.

6.6.2 The Good Cycle Factor Theorem

The purpose of this subsection, based on the paper [93] by Bang-Jensen, Gutin and Huang, is to prove some sufficient conditions for a semicomplete multipartite digraph to be hamiltonian.

Let $\mathcal{F} = C_1 \cup C_2$ be a cycle factor or a 1-path-cycle factor in a digraph D, where C_1 is a cycle or a path in D and C_2 is a cycle. A vertex $v \in V(C_i)$ is called **out-singular** (**in-singular**) with respect to C_{3-i} if $v \Rightarrow C_{3-i}$ ($C_{3-i} \Rightarrow v$); v is **singular** with respect to C_{3-i} if it is either out-singular or in-singular with respect to C_{3-i}.

Lemma 6.6.10 *[93] Let $Q \cup C$ be a cycle factor in a semicomplete multipartite digraph D. Suppose that the cycle Q has no singular vertices (with respect to C) and D has no hamiltonian cycle, then for every arc xy of Q either the arc xy itself can be inserted into C, or both vertices x and y can be inserted into C.*

Proof: Assume without loss of generality that there is some arc xy on Q such that neither x nor xy can be inserted into C. Since D is a semicomplete multipartite digraph and x is non-singular and cannot be inserted into C, there exists a vertex v on C which is not adjacent to x and $v^- \rightarrow x \rightarrow v^+$. Furthermore, v is adjacent to y since x and y are adjacent. Since xy cannot be inserted into C, we have $v \rightarrow y$. Then D contains a Hamilton cycle $Q[y,x]C[v^+,v]y$, which contradicts the assumption. □

Lemma 6.6.11 *[93] Let D be a semicomplete multipartite digraph containing a cycle factor $C_1 \cup C_2$ such that C_i has no singular vertices with respect to C_{3-i}, for both $i = 1, 2$; then D is hamiltonian. Given C_1 and C_2, a hamiltonian cycle in D can be found in time $O(|V(C_1)||V(C_2)|)$.*

Proof: If at least one of the cycles C_1, C_2 is even, then by Corollary 6.4.20 and Lemma 6.6.10 we can find a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Thus, assume that both of C_1, C_2 are odd cycles. If some vertex in C_i can be inserted into C_{3-i} for some $i = 1$ or 2, then by Corollary 6.4.18 and Lemma 6.6.10, we can construct a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Thus, we may also assume that no vertex in C_i can be inserted into C_{3-i} for both $i = 1, 2$. So, by Lemma 6.6.10, every arc of C_i can be inserted into C_{3-i}.

Now we show that either D is hamiltonian or we may assume that every arc of C_i can be inserted between two different pairs of vertices in C_{3-i} $(i = 1, 2)$. Consider an arc x_1x_2 of C_1. Since both x_1 and x_2 are non-singular and cannot be inserted into C_2, there exist vertices v_1 and v_2 on C_2 such that v_i is not adjacent to x_i and $v_i^- \rightarrow x_i \rightarrow v_i^+$, $i = 1, 2$. If $v_1 \rightarrow x_2$, then we obtain a Hamilton cycle. So we may assume that the only arc between x_2 and v_1 is x_2v_1. For the same reason, we may assume that v_2 dominates x_1 but is not dominated by x_1. Now the arc x_1x_2 can be inserted between v_1^- and v_1 and between v_2 and v_2^+.

Hence, x_1x_2 cannot be inserted between two pairs of vertices only in the case that $v_1^- = v_2$ and $v_1 = v_2^+$. We show that in this case D is hamiltonian. Construct, at first, a cycle $C^* = C_1[x_2, x_1]C_2[v_1^+, v_2^-]x_2$ which contains all the vertices of D but v_1^-, v_1. The arc $v_1^-v_1$ can be inserted into C_1, by the remark at the beginning of the proof. But $v_1^-v_1$ cannot be inserted between x_1 and x_2, since v_1 does not dominate x_2 and $v_1^- = v_2$ is not dominated by x_1. Hence, the arc $v_1^-v_1$ can be inserted into C^* to give a hamiltonian cycle of D. This completes the proof that either D is hamiltonian or every arc on C_i can be inserted between two different pairs of vertices in C_{3-i}.

Assume without loss of generality that the length of C_2 is not greater than that of C_1. Then C_1 has two arcs x_iy_i $(i = 1, 2)$ that can be inserted

between the same pair u, v of vertices in C_2. Since C_1 is odd, one of the paths $Q = C_1[y_1^+, x_2^-]$ and $C_1[y_2^+, x_1^-]$ has odd length. Without loss of generality, suppose that Q is odd. Obviously, $C^* = C_2[v, u]C_1[x_2, y_1]v$ is a cycle of D. By the fact shown above each arc of the path Q can be inserted into C_2 between a pair of vertices different from u, v. Therefore, each arc of Q can be inserted into C^*. Hence, by Corollary 6.4.19 we conclude that D has a hamiltonian cycle H. It is not difficult to verify that H can be found in time $O(|V(C_1)||V(C_2)|)$. □

Let D be a semicomplete multipartite digraph and let $C \cup C'$ be a cycle subdigraph of D. We write that $C \simeq\!> C'$ if C contains singular vertices with respect to C' and they all are out-singular, and C' has singular vertices with respect to C and they all are in-singular. A cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ is **good** if for every pair i, j, $1 \leq i < j \leq t$, neither $C_i \simeq\!> C_j$ nor $C_j \simeq\!> C_i$.

Since this definition and the proof of Lemma 6.6.12 are quite important, we illustrate them in Figure 6.6. Observe that if C, C' are a pair of disjoint cycles in a semicomplete multipartite digraph D, then (up to switching the role of the two cycles) at least one of the following four cases applies (see Figure 6.6):

(a) Every vertex on C has an arc to and from C'.
(b) There exist vertices $x \in V(C), y \in V(C')$ such that $x \Rightarrow V(C')$ and $y \Rightarrow V(C)$, or $V(C') \Rightarrow x$ and $V(C) \Rightarrow y$.
(c) C contains distinct vertices x, y such that $x \Rightarrow V(C')$ and $V(C') \Rightarrow y$.
(d) $C \simeq\!> C'$.

The alternatives (a)-(c) are covered by the definition of a good cycle factor (for cycle factors containing only two cycles); the alternative (d) is not.

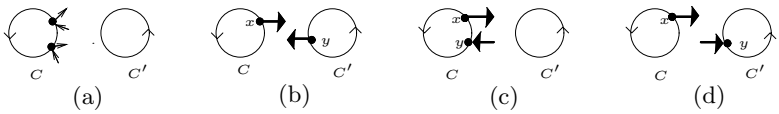


Figure 6.6 The four possible situations (up to switching the role of the two cycles or reversing all arcs) for arcs between two disjoint cycles in a semicomplete multipartite digraph. In (a) every vertex on C has arcs to and from C'. In (b)-(d) a fat arc indicates that all arcs go in the direction shown from or to the specified vertex (i.e., in (b) all arcs between x and C' leave x).

The following lemma gives the main result for a good cycle factor containing two cycles.

Lemma 6.6.12 *[93] If D is a semicomplete multipartite digraph containing a good factor $C_1 \cup C_2$, then D is hamiltonian. A Hamilton cycle in D can be constructed in time $O(|V(C_1)||V(C_2)|)$.*

Proof: The first case is that at least one of the cycles C_1 and C_2 has no singular vertices (Situation (a) in Figure 6.6). If both C_1, C_2 have no singular vertices, then D is hamiltonian by Lemma 6.6.11 and we can find a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Assume now that only one of them has no singular vertices. Suppose without loss of generality that C_1 contains an out-singular vertex x and C_2 has no singular vertices. Since C_2 contains no singular vertices, C_1 has at least one vertex which is not out-singular. Suppose that $x \in V(C_1)$ was chosen such that x^+ is not out-singular. Hence there is a vertex y on C_2 dominating x^+. If $x \rightarrow y$, then y can be inserted into C_1 and hence, by Lemma 6.6.10 and Corollary 6.4.18, D is hamiltonian (consider $C_2[y^+, y]$ and C_1). Otherwise, x is not adjacent to y. In this case, $x \rightarrow y^+$ and D has the hamiltonian cycle $C_1[x^+, x]C_2[y^+, y]x$. The above arguments can be easily converted into an $O(|V(C_1)||V(C_2)|)$ algorithm.

Consider the second case: each of C_1, C_2 has singular vertices with respect to the other cycle. Assume without loss of generality that C_1 has an out-singular vertex x_1. If C_2 also contains an out-singular vertex x_2 (Situation (b) in Figure 6.6), then x_1 is not adjacent to x_2 and $x_i \rightarrow x_{3-i}^+$ for both $i = 1, 2$. Hence D is hamiltonian. If C_2 contains no out-singular vertices, then it has in-singular vertices. Since $C_1 \cup C_2$ is a good factor, C_1 contains both out-singular and in-singular vertices (Situation (c) in Figure 6.6). Since both C_1 and C_2 have in-singular vertices, the digraph D' obtained from D by reversing the orientations of the arcs of D has two cycles C_1' and C_2' containing out-singular vertices. We conclude that D' (and hence D) is hamiltonian. Again, the above arguments can be converted into an $O(|V(C_1)||V(C_2)|)$ algorithm. □

The main result on good cycle factors is the following theorem by Bang-Jensen, Gutin and Huang. This theorem can be proved by induction on t, the number of cycles in a good cycle factor. We leave the details to the reader.

Theorem 6.6.13 (Bang-Jensen, Gutin and Huang) *[93] If D is a strong semicomplete multipartite digraph containing a good cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ ($t \geq 1$), then D is hamiltonian. Furthermore, given $\mathcal{F}$ one can find a hamiltonian cycle in D in time $O(n^2)$.* □

6.6.3 Consequences of Lemma 6.6.12

In this subsection mostly based on [93], we will show that several important results on semicomplete multipartite digraphs are consequences of Lemma 6.6.12.

Proof of Theorem 6.6.1: It is sufficient to prove that if P is a path and C is a cycle of D such that $V(P) \cap V(C) = \emptyset$, then D has a path P' with $V(P') = V(P) \cup V(C)$. Let P and C be such a pair, and let u be the initial and v the terminal vertex of P. If u is non-singular or in-singular with respect to C, then obviously the path P' exists. Similarly if v is non-singular or

out-singular with respect to C. Assume now that u is out-singular and v is in-singular with respect to C.

Add a new vertex w to D and the arcs zw, for all $z \neq u$, and the arc wu to obtain the semicomplete multipartite digraph D'. Then w forms a cycle C' with P in D' and $C \cup C'$ is a good cycle factor of D'. Therefore, by Lemma 6.6.12, D' has a hamiltonian cycle. Then D contains a hamiltonian path.

It is easy to see that the proof above supplies a recursive $O(n^2)$ algorithm for finding a hamiltonian path in D given a 1-path-cycle factor $\mathcal{F}$. Thus, the complexity result of this theorem is due to the fact that we can either construct a 1-path-cycle factor in a digraph or discover that it does not exist in time $O(n^{2.5})$: see Exercise 4.67. □

To obtain the rest of the proofs in this subsection, we need the following:

Lemma 6.6.14 *[93] Let D be a strong semicomplete multipartite digraph containing a cycle subdigraph $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ such that for every pair i, j $(1 \leq i \leq j \leq t)$ $C_i \Rightarrow C_j$ or $C_j \Rightarrow C_i$ holds. Then D has a cycle C of length at least $|V(\mathcal{F})|$ and one can find C in time $O(n^2)$ for a given $\mathcal{F}$. If D is an extended semicomplete digraph, then we can choose C such that $V(\mathcal{F}) \subseteq V(C)$.*

Proof: Define a tournament $T(\mathcal{F})$ as follows: $\{C_1, \ldots, C_t\}$ forms the vertex set of $T(\mathcal{F})$ and $C_i \rightarrow C_j$ in $T(F)$ if and only if $C_i \Rightarrow C_j$ in D. Let H be the subdigraph of D induced by the vertices of $\mathcal{F}$ and let W be a partite set of D having a representative in C_1.

First consider the case that $T(\mathcal{F})$ is strong. Then it has a hamiltonian cycle. Without loss of generality assume that $C_1 C_2 \ldots C_t C_1$ is a hamiltonian cycle in $T(\mathcal{F})$. If each of C_i $(i \in [t])$ has a vertex from W, then for every $i \in [t]$ choose any vertex w_i of $V(C_i) \cap W$. Then

$$C_1[w_1, w_1^-]C_2[w_2, w_2^-] \ldots C_t[w_t, w_t^-]w_1$$

is a hamiltonian cycle in H. If there exists a cycle C_i containing no vertices of W, then we may assume (shifting the cyclic order if needed) that C_t has no vertices from W. Obviously, H has a hamiltonian path starting at a vertex $w \in W \cap V(C_1)$ and finishing at some vertex v of C_t. Since $v \rightarrow w$, H is hamiltonian.

Now consider the case where $T(\mathcal{F})$ is not strong. Replacing in $\mathcal{F}$ every collection X of cycles which induce a strong component in $T(\mathcal{F})$ by a hamiltonian cycle in the subdigraph induced by X, we obtain a new cycle subdigraph $\mathcal{L}$ of D such that $T(\mathcal{L})$ has no cycles. The subdigraph $T(\mathcal{L})$ contains a unique hamiltonian path $Z_1 Z_2 \ldots Z_s$, where Z_i is a cycle of $\mathcal{L}$. Since D is strong there exists a path P in D with the first vertex in Z_s and the last vertex in Z_q $(1 \leq q < s)$ and the other vertices not in $\mathcal{L}$. Assume that q is as small as possible. Then we can replace the cycles $Z_q, \ldots, Z_s$ by a cycle consisting of all the vertices of $P \cup Z_q \cup \ldots \cup Z_s$ except maybe one and derive a new cycle

subdigraph with less cycles. Continuing in this manner, we obtain finally a single cycle.

In the case of an extended semicomplete digraph D, if $D\langle V(\mathcal{F})\rangle$ is not strong, then $T(\mathcal{F})$ is not strong. Also, $C_i \Rightarrow C_j$ implies that $C_i \mapsto C_j$. This, combined with the above argument on semicomplete multipartite digraphs, allows one to construct a cycle C such that $V(\mathcal{F}) \subset V(C)$.

Using the above proof together with an $O(n^2)$ algorithm for constructing a hamiltonian cycle in a strong tournament (see Theorem 6.3.2 or Exercise 6.5) and obvious data structures one can obtain an $O(n^2)$ algorithm. □

Lemma 6.6.15 *[93] Let $C \cup C'$ be a cycle factor in a strong semicomplete multipartite digraph D of order n. Then D has a cycle Z of length at least $n-1$ containing all vertices of C. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: Suppose that the (existence) claim is not true. By Lemma 6.6.12, this means that each of C and C' has singular vertices with respect to the other cycle, and all singular vertices on one cycle are out-singular and all singular vertices on the other cycle are in-singular. Assume without loss of generality that C has only out-singular vertices with respect to C'. Since D is strong C has a non-singular vertex x. Furthermore we can choose x such that its predecessor x^- on C is singular. Let y be some vertex of C' such that $y \rightarrow x$. If x^- is adjacent to y^+, the successor of y on C', then D has a hamiltonian cycle. Otherwise $x^- \rightarrow y^{++}$ and D has a cycle of length $n-1$ containing all vertices of C. The complexity result easily follows from the above arguments. □

The next two results due to Gutin are easy corollaries of Lemma 6.6.15:

Corollary 6.6.16 *[444] Let $C \cup C'$ be a cycle factor in a strong semicomplete bipartite digraph D. Then D has a hamiltonian cycle Z. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: Since D is bipartite, it cannot have a cycle of length $n-1$. □

Corollary 6.6.17 *[449] Let $C \cup C'$ be a cycle factor in a strong extended semicomplete digraph D. Then D has a hamiltonian cycle Z. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: If C and C' have a pair x, y of non-adjacent vertices ($x \in V(C)$, $y \in V(C')$), then obviously $x \rightarrow y^+$, $y \rightarrow x^+$ and D has a Hamilton cycle that can be found in time $O(|V(C)|V(C')|)$. Assuming that any pair of vertices from C and C' is adjacent, we complete the proof as in Lemma 6.6.15. □

Corollaries 6.6.16 and 6.6.17 imply immediately the following useful result.

Proposition 6.6.18 *If $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_k$ is a cycle factor in a digraph which is either semicomplete bipartite or extended semicomplete and there is no $\mathcal{F}' = C'_1 \cup C'_2 \cup \ldots \cup C'_r$ such that for every $i \in [k]$, $V(C_i) \subset V(C'_j)$ for some $j \in [r]$, then without loss of generality $C_i \Rightarrow C_j$ for every $i < j$.* □

Lemma 6.6.15 implies immediately the following result first proved by Ayel (see [555]).

Corollary 6.6.19 *If C is a longest cycle in a strong semicomplete multipartite digraph D, then $D - V(C)$ is acyclic.* □

Proof of Theorem 6.6.6: Let $\mathcal{F} = C_1 \cup \ldots \cup C_t$ be a cycle subdigraph of maximum order in a strong semicomplete bipartite digraph D. We construct a semicomplete digraph S, a generalization of the tournament T in Lemma 6.6.14, as follows. The vertices of S are the cycles in $\mathcal{F}$, $C_i \rightarrow C_j$ in S if and only if there is an arc from C_i to C_j in D. Cycles of length two in S indicate what cycles in $\mathcal{F}$ can be merged together by Corollary 6.6.16. Therefore, we can merge cycles in $\mathcal{F}$ till S becomes oriented, i.e., without 2-cycles. Now we can apply Lemma 6.6.14.

Complexity details are left to the reader. □

Proof of Theorem 6.6.7: The proof is similar to that of Theorem 6.6.6, applying Corollary 6.6.17 instead of Corollary 6.6.16. Details are left to the reader as Exercise 6.24. □

6.6.4 Yeo's Irreducible Cycle Subdigraph Theorem and Its Applications

While Lemma 6.6.12 is strong enough to imply short proofs of results on longest cycles in some special families of semicomplete multipartite digraphs such as semicomplete bipartite graphs and extended semicomplete digraphs, this lemma does not appear strong enough to be used in proofs of longest cycle structure results for other families of semicomplete multipartite digraphs. In this subsection, we formulate the very deep theorem of Yeo on irreducible cycle subdigraphs in semicomplete multipartite digraphs, the main theorem in [915], that is more powerful than Lemma 6.6.12. We provide short proofs of some important consequences of this theorem.

Recall that for two subdigraphs X, Y of D, a path P is an (X, Y)-path if P starts at a vertex $x \in V(X)$, terminates at a vertex $y \in V(Y)$ and $V(P) \cap (V(X) \cup V(Y)) = \{x, y\}$.

Theorem 6.6.20 (Yeo's irreducible cycle subdigraph theorem) *[915] Let D be a semicomplete multipartite digraph with partite sets $V_1, V_2, \ldots, V_c$. Let $X \subseteq V(D)$ and let $\mathcal{F}$ be a cycle subdigraph of D consisting of t cycles that covers X, such that t is minimum. Then the following holds.*

(a) We can label the cycles $C_1, C_2, \ldots, C_t$ of $\mathcal{F}$, such that $C_i \simeq\!> C_j$, whenever $1 \leq i < j \leq t$.

(b) Assume that $C_1, C_2, \ldots, C_t$ are ordered as stated in (a), then there are cycles $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$ ($n_0 = 1$, $n_m = t$), and integers $q_1, q_2, \ldots, q_m \in [c]$, such that the following is true. For every (C_j, C_i)-path P starting at

u and terminating at v with $V(P) \cap V(\mathcal{F}) = \{u, v\}$ and $1 \leq i < j \leq t$, there exists an integer $k \in [m]$, such that $n_{k-1} \leq i < j \leq n_k$ and $\{u^+_{C_j}, v^-_{C_i}\} \subseteq V_{q_k} \cap X$. □

By a careful analysis of the proof of Theorem 6.6.20 in [916] one can obtain the following:

Theorem 6.6.21 *[916] Let D be a semicomplete multipartite digraph, and let $X \subseteq V(D)$ be arbitrary. Let $\mathcal{F}$ be a cycle subdigraph of D that covers X. Then in $O(|V(D)|^3)$ time we can find a new cycle subdigraph, $\mathcal{F}'$, of D, that covers X, such that $\mathcal{F}'$ has the properties (a) and (b) given in Theorem 6.6.20. Furthermore we can find $\mathcal{F}'$, such that for every cycle C in $\mathcal{F}$, the vertices $X \cap V(C)$ are included in some cycle of $\mathcal{F}'$.* □

Theorems 6.6.20 and 6.6.21 are very important starting points of [97], where polynomial solvability of the Hamilton cycle problem for semicomplete multipartite digraphs is established. We will prove some important consequences of Theorem 6.6.20 and state several more of them.

Theorem 6.6.22 *[915] Every regular semicomplete multipartite digraph is hamiltonian.*

Proof: Let D be a regular semicomplete multipartite digraph. By Exercise 13.17, D contains a cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$. We may assume that $\mathcal{F}$ is chosen, such that t is minimum. If $t = 1$, then D is hamiltonian, so assume that $t > 1$.

Let $X = V(D)$. Let $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$ and $q_1, q_2, \ldots, q_m$ be defined as in Theorem 6.6.20. Let $yx \in A(D)$ be an arc from $y \in V(C_i)$, with $i \in \{2, 3, \ldots, t\}$ to $x \in V(C_1)$. Part (b) of Theorem 6.6.20 implies that $x^-, y^+ \in V_{q_1}$. Now we define the two distinct arcs $a_1(yx) = xy^+$ and $a_2(yx) = x^-y$. By Theorem 6.6.20, $a_1(yx)$ and $a_2(yx)$ are arcs in D. Indeed, x and y^+ (x^- and y) are adjacent. If $y^+ \rightarrow x$, then $y^{++} \in V_{q_1}$, which is impossible.

If $y'x'$ and yx are distinct arcs from $V(D) - V(C_1)$ to $V(C_1)$, then we see that $a_1(yx)$, $a_2(yx)$, $a_1(y'x')$ and $a_2(y'x')$ are four distinct arcs from $V(C_1)$ to $V(D) - V(C_1)$. We have now shown that the number of arcs leaving $V(C_1)$ is at least twice as large as the number of arcs entering $V(C_1)$. However, this contradicts the fact that D is an eulerian digraph (see Corollary 1.7.3). □

Theorem 6.6.23 (Yeo) *[915] Let D be a $(\lfloor k/2 \rfloor + 1)$-strong semicomplete multipartite digraph, and let X be an arbitrary set of vertices in D such that X includes at most k vertices from each partite set of D. If there is a cycle subdigraph $\mathcal{F} = C_1 \cup \ldots \cup C_t$, which covers X, then there is a cycle C in D, such that $X \subseteq V(C)$.*

Proof: We may clearly assume that $\mathcal{F}$ has the properties described in Theorem 6.6.20, and $t \geq 2$, since otherwise we are done. Let $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$

and $q_1, q_2, \ldots, q_m$ be defined as in Theorem 6.6.20. Since X contains at most k vertices from each partite set, we have that $\min\{|V_{q_1} \cap V(C_1) \cap X|, |V_{q_1} \cap V(C_{n_1}) \cap X|\} = r \leq \lfloor k/2 \rfloor$. Assume without loss of generality that $|V_{q_1} \cap V(C_{n_1}) \cap X| = r$. Since D is $(\lfloor k/2 \rfloor + 1)$-strong we get that there exists a $(V(C_{n_1}) - (V_{q_1} \cap V(C_{n_1}) \cap X)^-, V(C_1) \cup \ldots \cup V(C_{n_1-1}))$-path in $D - (V_{q_1} \cap V(C_{n_1}) \cap X)^-$, $P = p_1 \ldots p_l$. Assume that $p_l \in V(C_i)$ $(1 \leq i < n_1)$. By Theorem 6.6.20, the (C_{n_1}, C_i)-path P contradicts the minimality of $\mathcal{F}$, since $n_0 \leq i < n_1$ and $p_1^+ \notin X \cap V_{q_1}$. □

A family of semicomplete multipartite digraphs described in [915] shows that one cannot weaken the value $\lfloor k/2 \rfloor + 1$ of strong connectivity in this theorem. Using the fact that every k-strong digraph of independence number at most k has a cycle factor (see Proposition 13.8.2) and applying Theorem 6.6.23, we obtain the following two corollaries:

Corollary 6.6.24 *[915] If a k-strong semicomplete multipartite digraph D has at most k vertices in each partite set, then D contains a Hamilton cycle.* □

Corollary 6.6.25 *[915] A k-strong semicomplete multipartite digraph has a cycle through any set of k vertices.* □

Theorem 6.6.22 was generalized by Yeo [919] as follows (its proof also uses Theorem 6.6.20). Let $i_l(D) = \max\{|d^+(x) - d^-(x)| : x \in V(D)\}$ and $i_g(D) = \Delta^0(D) - \delta^0(D)$ for a digraph D (the two parameters are called the **local irregularity** and the **global irregularity**, respectively, of D [919]). Clearly, $i_l(D) \leq i_g(D)$ for every digraph D.

Theorem 6.6.26 *[919] Let D be a semicomplete c-partite digraph of order n with partite sets of cardinalities $n_1, n_2, \ldots, n_c$ such that $n_1 \leq n_2 \leq \ldots \leq n_c$. If $i_g(D) \leq (n - n_{c-1} - 2n_c)/2 + 1$ or $i_l(D) \leq \min\{n - 3n_c + 1, (n - n_{c-1} - 2n_c)/2 + 1\}$, then D is hamiltonian.* □

The result of this theorem is best possible in a sense: Yeo [919] constructed an infinite family $\mathcal{D}$ of non-hamiltonian semicomplete multipartite digraphs such that every $D \in \mathcal{D}$ has $i_l(D) = i_g(D) = (n - n_{c-1} - 2n_c + 1)/2 + 1 \leq n - 3n_c + 2$.

There are many corollaries of this theorem including the following one by Volkmann and Yeo:

Theorem 6.6.27 *[897] Every arc of a regular multipartite tournament is contained in a Hamilton path.* □

Another generalization of Theorem 6.6.22, whose proof is based on Theorem 6.6.20, was obtained by Guo, Tewes, Volkmann and Yeo [439]. For a digraph D and a positive integer k, define

$$f(D,k) = \sum_{x \in V(D), d^+(x) > k} (d^+(x) - k) + \sum_{x \in V(D), d^-(x) < k} (k - d^-(x)).$$

Theorem 7.5.3 in Ore's book [733] on the existence of a perfect matching in a bipartite graph can easily be transformed into a sufficient condition for a digraph to contain a cycle factor. This condition is as follows. If, for a digraph D and positive integer k, we have $f(D,k) \leq k-1$, then D has a cycle factor. For a positive integer $k \geq 2$, let G'_k be a semicomplete 3-partite digraph with the partite sets $V_1 = \{x\}$, $V_2 = \{y_1, y_2, \ldots, y_{k-1}\}$ and $V_3 = \{z_1, z_2, \ldots, z_k\}$ and arc set

$$\{yx, xz, zy, yv : \ y \in V_2, z \in V_3, v \in V_3 - z_1\} \cup \{z_1 x\}.$$

The digraph G''_k is the converse of G'_k. We observe that $f(G'_k, k) = k-1$ (Exercise 6.28), but G'_k is not hamiltonian, as a hamiltonian cycle would contain the arc xz_1 and every second vertex on the cycle would belong to the partite set V_3. Since x has no in-neighbour in $V_3 - z_1$, this is not possible. Clearly, G''_k is not hamiltonian either.

Theorem 6.6.28 *[439] Let D be a semicomplete multipartite digraph such that $f(D,k) \leq k-1$ for some positive integer k. If D is not isomorphic to G'_k or G''_k, then D is Hamiltonian.* □

The authors of [439] introduced the following family of semicomplete multipartite digraphs. Let D be a semicomplete multipartite digraph with partite sets $V_1, V_2, \ldots, V_k$. If $\min\{|(x_i, V_j)|, |(V_j, x_i)|\} \geq \frac{1}{2}|V_j|$ for every vertex $x_i \in V_i$ and for every $1 \leq i, j \leq k$, $j \neq i$, then D is called a **semi-partitioncomplete digraph**. Several sufficient conditions to guarantee hamiltonicity of semi-partitioncomplete digraphs were derived in [439]. In particular, the following result was proved.

Theorem 6.6.29 *If a strong semi-partitioncomplete digraph D of order n has less than $n/2$ vertices in every partite set, then D is hamiltonian.* □

6.7 Hamilton Paths and Cycles in Quasi-Transitive Digraphs

The methods developed in [103] by Bang-Jensen and Huang and [454] by Gutin to characterize hamiltonian and traceable quasi-transitive digraphs as well as to construct polynomial algorithms for verifying the existence of Hamilton paths and cycles in quasi-transitive digraphs can be easily generalized to much wider classes of digraphs [89]. Thus, in this section, along with quasi-transitive digraphs, we consider totally Φ-decomposable digraphs for various sets Φ of digraphs.

By Theorem 2.7.5, every strong quasi-transitive digraph D has a decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$, where S is a strong semicomplete digraph, $s = |V(S)|$, and each Q_i, $i \in [s]$, is either just a single vertex or a non-strong quasi-transitive digraph. Also, a non-strong quasi-transitive digraph D with at least two vertices has a decomposition $D = T[H_1, H_2, \ldots, H_t]$, where T is a transitive oriented graph, $t = |V(T)|$, and every H_i is a strong semicomplete digraph. These decompositions are called canonical decompositions. The following characterization of hamiltonian quasi-transitive digraphs is due to Bang-Jensen and Huang [103].

Theorem 6.7.1 *[103] A strong quasi-transitive digraph D with canonical decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$ is hamiltonian if and only if it has a cycle factor $\mathcal{F}$ such that no cycle of $\mathcal{F}$ is a cycle of some Q_i.*

Proof: Clearly, a Hamilton cycle in D crosses every Q_i. Thus, it suffices to show that if D has a cycle factor $\mathcal{F}$ such that no cycle of $\mathcal{F}$ is a cycle of some Q_i, then D is hamiltonian. Observe that $V(Q_i) \cap \mathcal{F}$ is a path factor $\mathcal{F}_i$ of Q_i for every $i \in [s]$. For every $i \in [s]$, delete the arcs between end-vertices of all paths in $\mathcal{F}_i$ except for the paths themselves, and then perform the operation of path-contraction for all paths in $\mathcal{F}_i$. As a result, one obtains an extended semicomplete digraph S' (since S is semicomplete). Clearly, S' is strong and has a cycle factor. Hence, by Theorem 6.6.5, S' has a Hamilton cycle C. After replacing every vertex of S' with the corresponding path from $\mathcal{F}$, we obtain a Hamilton cycle in D. □

Similarly to Theorem 6.7.1, one can prove the following characterization of traceable quasi-transitive digraphs (see Exercise 6.29).

Theorem 6.7.2 *[103] A quasi-transitive digraph D with at least two vertices and with canonical decomposition $D = R[G_1, G_2, \ldots, G_r]$ is traceable if and only if it has a 1-path-cycle factor $\mathcal{F}$ such that no cycle or path of $\mathcal{F}$ is completely in some $D\langle V(G_i)\rangle$.* □

It appears that Theorems 6.7.1 and 6.7.2 do not imply polynomial algorithms to verify hamiltonicity and traceability, respectively. The following characterization of hamiltonian quasi-transitive digraphs is given implicitly in the paper [454] by Gutin:

Theorem 6.7.3 (Gutin) *[454] Let D be a strong quasi-transitive digraph with canonical decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$. Let $n_1, \ldots, n_s$ be the orders of the digraphs $Q_1, Q_2, \ldots, Q_s$, respectively. Then D is hamiltonian if and only if the extended semicomplete digraph $S' = S[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_s}]$ has a cycle subdigraph which covers at least $\mathrm{pc}(Q_j)$ vertices of $\overline{K}_{n_j}$ for every $j \in [s]$.*

Proof: Suppose that D has a Hamilton cycle H. For every $j \in [s]$, $V(Q_j) \cap H$ is a k_j-path factor $\mathcal{F}_j$ of Q_j. By the definition of the path covering number,

we have $k_j \geq \mathrm{pc}(Q_j)$. For every $j \in [s]$, the deletion of the arcs between end-vertices of all paths in $\mathcal{F}_j$ except for the paths themselves, and then path-contraction of all paths in $\mathcal{F}_j$ transforms H into a cycle of S' having at least $\mathrm{pc}(Q_j)$ vertices of $\overline{K}_{n_j}$ for every $j \in [s]$.

Suppose now that S' has a cycle subdigraph $\mathcal{L}$ containing $p_j \geq \mathrm{pc}(Q_j)$ vertices of $\overline{K}_{n_j}$ for every $j \in [s]$. Since S' is a strong extended semicomplete digraph, by Theorem 6.6.7, S' has a cycle C such that $V(C) = V(\mathcal{L})$. Clearly, every Q_j has a p_j-path factor $\mathcal{F}_j$. Replacing, for every $j \in [s]$, the p_j vertices of $\overline{K}_{n_j}$ in C with the paths of $\mathcal{F}_j$, we obtain a hamiltonian cycle in D. □

Theorem 6.7.3 can be used to show that the Hamilton cycle problem for quasi-transitive digraphs is polynomial time solvable.

Theorem 6.7.4 (Gutin) *[454] There is an $O(n^4)$ algorithm which, given a quasi-transitive digraph D, either returns a hamiltonian cycle in D or verifies that no such cycle exists.* □

The approach used in the proofs of Theorems 6.7.3 and 6.7.4 in [454] can be generalized to a much wider class of digraphs as was observed by Bang-Jensen and Gutin [89]. We follow the main ideas of [89].

Theorem 6.7.5 *Let Φ be an extension-closed set of digraphs, i.e., $\Phi^{ext} = \Phi$, including the trivial digraph $\overline{K}_1$ on one vertex. Suppose that for every digraph $H \in \Phi$ we have $\mathrm{pcc}(H) = \mathrm{pc}(H)$. Let D be a totally Φ-decomposable digraph. Then, given a total Φ-decomposition of D, the path covering number of D can be calculated and a minimum path factor found in time $O(n^4)$.*

Proof: We prove this theorem by induction on n. For $n = 1$ the claim is trivial.

Let D be a totally Φ-decomposable digraph and let $D = R[H_1, \ldots, H_r]$ be a Φ-decomposition of D such that $R \in \Phi$, $r = |V(R)|$ and every H_i (of order n_i) is totally Φ-decomposable. A $\mathrm{pc}(D)$-path factor of D restricted to every H_i corresponds to a disjoint collection of some p_i paths covering $V(H_i)$. Hence, we have $\mathrm{pc}(H_i) \leq p_i \leq n_i$. Therefore, arguing similarly to that in the proof of Theorem 6.7.3, we obtain

$$\mathrm{pc}(D) = \min\{\mathrm{pc}(R[\overline{K}_{p_1}, \ldots, \overline{K}_{p_r}]) : \ \mathrm{pc}(H_i) \leq p_i \leq n_i, \ i \in [r]\}. \qquad (6.9)$$

Since Φ is extension-closed, and since, for every digraph $Q \in \Phi$, $\mathrm{pc}(Q) = \mathrm{pcc}(Q)$, we obtain

$$\mathrm{pc}(D) = \min\{\mathrm{pcc}(R[\overline{K}_{p_1}, \ldots, \overline{K}_{p_r}]) : \ \mathrm{pc}(H_i) \leq p_i \leq n_i, \ i \in [r]\}. \qquad (6.10)$$

By the result of Exercise 4.70, given the lower and upper bounds $\mathrm{pc}(H_i)$ and n_i ($i \in [r]$), we can find the minimum in (6.10) and thus $\mathrm{pc}(D)$ in time $O(n^3)$. Let $T(n)$ be the time needed to find the path covering number of a totally Φ-decomposable digraph of order n. Then, by (6.10),

$$T(n) = O(n^3) + \sum_{i=1}^{r} T(n_i).$$

Furthermore, $T(1) = O(1)$. Hence $T(n) = O(n^4)$. □

Recall (see Section 2.11) that Φ_0 (Φ_2) is the family of all semicomplete multipartite, extended locally semicomplete and acyclic digraphs (semicomplete bipartite, extended locally semicomplete and acyclic digraphs). Clearly, both families of digraphs are extension-closed. As we know, $\mathrm{pc}(D) = \mathrm{pcc}(D)$ for every semicomplete multipartite digraph D (see Theorem 6.6.2), for every extended locally semicomplete digraph D (by Theorem 5.8.1 in [91]) and every acyclic digraph D (which is trivial). Notice that one can check whether a digraph D is totally Φ_0-decomposable (totally Φ_2-decomposable) and, if this is the case, find a total Φ_0-decomposition (Φ_2-decomposition) in time $O(n^4)$ (see Section 2.11). Therefore, Theorem 6.7.5 implies the following theorem by Bang-Jensen and Gutin:

Theorem 6.7.6 *[90] The path covering number of a totally Φ_0-decomposable digraph can be calculated in time $O(n^4)$.* □

Corollary 6.7.7 *[90] One can verify whether a totally Φ_2-decomposable digraph is hamiltonian in time $O(n^4)$.*

Proof: Let $D = R[H_1, \ldots, H_r]$, $r = |R|$, be a decomposition of a strong digraph D ($r \geq 2$). Then, D is hamiltonian if and only if the following family $\mathcal{S}$ of digraphs contains a hamiltonian digraph:

$$\mathcal{S} = \{R[\overline{K}_{p_1}, \ldots, \overline{K}_{p_r}] : \ \mathrm{pc}(H_i) \leq p_i \leq |V(H_i)|, \ i \in [r]\}.$$

Now suppose that D is a totally Φ_2-decomposable digraph. Then, every digraph of the form $R[\overline{K}_{p_1}, \ldots, \overline{K}_{p_r}]$ is in Φ_2. We know (see Theorem 6.6.4 and Theorem 5.8.1 in [91]) that every digraph in Φ_2 is hamiltonian if and only if it is strong and contains a cycle factor. Thus, all we need is to verify whether there is a digraph in $\mathcal{S}$ containing a cycle factor. It is easily seen that there is a digraph in $\mathcal{S}$ containing a cycle factor if and only if there is a circulation in the network formed from R by adding lower bounds $\mathrm{pc}(H_i)$ and upper bounds $|V(H_i)|$ to the vertex v_i of R for every $i \in [r]$. Since the lower bounds can be found in time $O(n^4)$ (see Theorem 6.7.5) and the existence of a circulation checked in time $O(n^3)$ (see Exercise 4.31), we obtain the required complexity $O(n^4)$. □

Since every quasi-transitive digraph is totally Φ_2-decomposable this theorem immediately implies Theorem 6.7.4. Note that the minimum path factors in Theorem 6.7.5 can be found in time $O(n^4)$. Also, a hamiltonian cycle in a hamiltonian totally Φ_2-decomposable digraph can be constructed in time $O(n^4)$.

6.8 Vertex-Cheapest Paths and Cycles

In this section, we consider problems that generalize the Hamilton path and cycle problems in a significant way. We prove that the problems of finding vertex-cheapest paths and cycles in vertex-weighted quasi-transitive digraphs are polynomial time solvable. The values of the weights can be any reals, positive or negative. Thus, we can conclude that the longest and shortest path and cycle problems for quasi-transitive digraphs are polynomial time solvable. The same result holds for acyclic digraphs as the only non-trivial problem from the above four is the longest path problem and it is well-known that it can be solved in polynomial time, see Section 3.3.2. Notice that for the quasi-transitive digraphs three of the above four problems are non-trivial (the shortest and longest cycles and longest path) and, in fact, much more difficult than the longest path problem for acyclic digraphs as the reader can see in the rest of the section. It appears that the problems are non-trivial even for semicomplete digraphs. The following results were proved by Bang-Jensen, Gutin and Yeo for extended semicomplete and locally semicomplete digraphs.

Theorem 6.8.1 *[100] Let $D = (V, A)$ be an extended semicomplete digraph with a cycle and real-valued costs on the vertices. In time $O(n^3m + n^4 \log n)$ we can find a minimum cost cycle in D.* □

Theorem 6.8.2 *[98] Let D be a locally semicomplete digraph with real-valued costs on the vertices. In time $O(n(m+n \log n))$ we can find a minimum cost cycle of D.* □

The approach described in the previous section seems too weak to allow us to construct polynomial time algorithms for vertex-cheapest paths and cycles in quasi-transitive digraphs. A more powerful method that leads to such algorithms was first suggested by Bang-Jensen, Gutin and Yeo [100] and, in the rest of this section, we describe this method.

6.8.1 Vertex-Cheapest Paths and Cycles in Quasi-Transitive Digraphs

Recall that the cost of a subset of vertices is the sum of the costs of its vertices and the cost of a subdigraph is the sum of the costs of its vertices. For a digraph D of order n and $i \in [n]$ we define $mp_i(D)$ ($mpc_i(D)$) to be the minimum cost of an i-path (i-path-cycle) subdigraph of D. We set $mp_0(D) = 0$ and $mpc_0(D)$ is zero if D has no negative cycle and otherwise it is the minimum cost of a cycle subdigraph in D. Note that $mp_0(D)$ and $mpc_0(D)$ always exist as we may take single vertices as paths and we always have $mpc_i(D) \leq mp_i(D)$. For any digraph D with at least one cycle we denote by $mc(D)$ the minimum cost of a cycle in D.

Let $D = (V, A)$ be a digraph and let X be a non-empty subset of V. We say that a cycle C in D is an X-**cycle** if C contains all vertices of X. In this section, we consider the following problems for a digraph $D = (V, A)$ with n vertices and real-valued costs on the vertices:

(P1) Determine $mp_i(D)$ for all $i \in [n]$.
(P2) Find a cheapest cycle in D or determine that D has no cycle.

Clearly, problems (P1) and (P2) are NP-hard as determining the numbers $mp_1(D)$ and $mc(D)$ generalize the hamiltonian path and cycle problems (assign cost -1 to each vertex of D). The problem (P2) can be solved in time $O(n^3)$ when all costs are non-negative using an all pairs shortest path calculation. The problems (P1) and (P2) were solved in [87] for the special case when all costs are non-negative. However, the approach of [87] cannot be used or modified to work with negative costs. Bang-Jensen, Gutin and Yeo [100] managed to obtain an approach suitable for arbitrary real costs.

6.8.2 Minimum Cost k-Path-Cycle Subdigraphs

In this subsection, we will use certain notions and results on network flows from Chapter 4. As in Chapter 4, we will allow capacities and costs on the vertices in our networks. This makes it easier to model certain problems for digraphs and it is easy to transform such a network into one where all capacities and costs are on the arcs (see Subsection 4.2.4 for details). With these remarks in mind, the following lemma follows directly from Lemma 4.2.4 and Proposition 4.10.7.

Lemma 6.8.3 *Let $N = (V, A)$ be a network with source s and sink t, capacities on arcs and vertices and a real-valued cost $c(v)$ for each vertex $v \in V$. For all integers i such that there exists a feasible (s,t)-flow of value i in N, let f_i be a minimum cost (s,t)-flow in N of value i and let $c(f_i)$ be the cost of f_i. Then, for all i where all of f_{i-1}, f_i, f_{i+1} exist, we have*

$$c(f_{i+1}) - c(f_i) \geq c(f_i) - c(f_{i-1}). \tag{6.11}$$

□

Recall that a cycle subdigraph of a digraph D is a collection of vertex-disjoint cycles of D.

Lemma 6.8.4 *Let $D = (V, A)$ be a digraph with real-valued cost function c on the vertices. In time $O(n(m + n \log n))$ we can determine the number $mpc_0(D)$ and find a cycle subdigraph of cost $mpc_0(D)$ if $mpc_0(D) < 0$.*

Proof: Let $H(w)$ be the digraph on 4 vertices w_1, w_2, w_3, w_4 and the following arcs $w_1w_2, w_2w_1, w_2w_3, w_3w_4, w_4w_3$. Let $D^* = (V^*, A^*)$ be obtained from D as follows: replace every vertex v by the digraph $H(v)$. Furthermore, for

every original arc $uv \in A$, D^* contains the arc u_4v_1. There are no costs on the vertices and all arcs have cost 0 except the arcs of the form v_2v_3 which have cost $c(v)$. Observe that $mpc_0(D)$ is precisely the minimum cost of a spanning cycle subdigraph in D^*. Let $V^* = \{x_1, x_2, \ldots, x_{4n}\}$. Construct a bipartite graph B with partite sets $L = \{\ell_1, \ldots, \ell_{4n}\}$ and $R = \{r_1, \ldots, r_{4n}\}$, in which $\ell_i r_j$ is an edge if and only if $x_i x_j \in A^*$. Moreover, the cost of $\ell_i r_j$ is equal to the cost of $x_i x_j$. Observe that a minimum cost perfect matching in B corresponds to a minimum cost cycle subdigraph in D^*. We can find a minimum cost perfect matching in B in time $O(n(m+n\log n))$, see the remark after the proof of Theorem 11.1 in [622]. Using the transformation from B to D^*, we can compute the minimum cost of a spanning cycle subdigraph F in D^* in time $O(n(m+n\log n))$. If this cost is negative, we can find a minimum cost cycle subdigraph of D within the same time. □

Lemma 6.8.5 *Let $D = (V, A)$ be a vertex-weighted digraph.*

(a) In total time $O(n^2m + n^3)$ we can determine the numbers $\{mpc_1(D), mpc_2(D), \ldots, mpc_n(D)\}$ and find j-path-cycle subdigraphs F_j, $j = 1, 2, \ldots, n$, where F_j has cost $mpc_j(D)$.

(b) The costs $mpc_i(D)$ satisfy the following inequality for every $i \in [n-1]$:

$$mpc_{i+1}(D) - mpc_i(D) \geq mpc_i(D) - mpc_{i-1}(D). \tag{6.12}$$

Proof: Form a network $N(D)$ from D by adding a pair s, t of new vertices along with arcs $\{(s, v), (v, t) : v \in V\}$. Let all vertices and all arcs of D have lower bound 0 and capacity 1. Let $c(s) = c(t) = 0$, let each other vertex of $N(D)$ inherit its cost from D and let all arcs have cost 0.

Suppose F_j is a j-path-cycle subdigraph of D. Using F_j we can obtain a feasible flow f_j of value j in $N(D)$ if we assign $f_j(a) = 1$ to all arcs a in F_j and those arcs a of $N(D)$ that start (terminate) at s (t) and terminate (start) at the initial (terminal) vertex of a path in F_j, and $f_j(a) = 0$ for all other arcs of $N(D)$. Similarly, by Theorem 4.3.1, we can transform a feasible integer-valued (s, t)-flow of value j in $N(D)$ into a j-path-cycle subdigraph of D.

Notice that $N(D)$ has a feasible integer-valued (s, t)-flow of value k for any integer $k = 0, 1, \ldots, n$. Thus it follows from the observations above that for every $j = 0, 1, \ldots, n$ the value $mpc_j(D)$ is exactly the minimum cost of a flow of value j in $N(D)$. Now (6.11) implies that the inequality (6.12) is valid.

It remains to prove (a). It follows from Lemma 6.8.4 that we can find a minimum cost flow f of value 0 in time $O(n^3)$. Now we can use the Buildup algorithm from Subsection 4.10.2 starting from f. Using the Buildup algorithm we can find feasible integer-valued flows f_j for all $j \in [n]$, such that f_j is a minimum cost feasible (s, t)-flow of value j in $N(D)$, in total time $O(n^2m)$ (the complexity of obtaining f_{j+1} starting from f_j is $O(nm)$). This proves (a). □

6.8.3 Cheapest i-Path Subdigraphs in Quasi-Transitive Digraphs

By Theorem 6.6.1, in a semicomplete multipartite digraph D all cycles of a k-path-cycle subdigraph with $k \geq 1$ can be merged with one of the paths to form a new path. This easily implies the following lemma which plays an important role in our algorithms.

Lemma 6.8.6 *Let D be a semicomplete multipartite digraph. Then for every $i \in [n]$ we have $mp_i(D) = mpc_i(D)$.* □

The next theorem shows that (P1) is polynomially solvable for quasi-transitive digraphs.

Theorem 6.8.7 *Let $D = (V, A)$ be a vertex-weighted quasi-transitive digraph. Then the following holds:*

(a) In total time $O(n^2m + n^3)$ we can find for every $i \in [n]$, the value of $mp_i(D)$ and an i-path subdigraph F_i of cost $mp_i(D)$.
(b) For all $i \in [n-1]$ we have

$$mp_{i+1}(D) - mp_i(D) \geq mp_i(D) - mp_{i-1}(D). \tag{6.13}$$

Proof: We prove (b) by induction on n. The statement vacuously holds for $n = 1$ and is easy to verify for $n = 2$ (recall that, by definition, $mp_0(D) = 0$). This proves the basis of induction and we now assume that $n \geq 3$.

By Theorem 2.7.5, D has a decomposition $D = T[Q_1, \ldots, Q_t]$, $t = |T| \geq 2$, where T is an acyclic digraph or a semicomplete digraph. Let $D' = T[\overline{K}_{n_1}, \ldots, \overline{K}_{n_t}]$ be obtained from D by deleting all arcs inside each Q_i, $i \in [t]$. Assign costs to the vertices $v_1^k, \ldots, v_{n_k}^k$ of $\overline{K}_{n_k}$, as follows:

$$c'(v_j^k) = mp_j(Q_k) - mp_{j-1}(Q_k). \tag{6.14}$$

By the induction hypothesis (b) holds for Q_k implying that we have

$$c'(v_j^k) \leq c'(v_{j+1}^k) \text{ for every } j \geq 1. \tag{6.15}$$

Let F_i' be an i-path-cycle subdigraph of D'. If T is acyclic, then D' is acyclic and, thus, F_i' is an i-path subdigraph of D'. If T is semicomplete, then D' is extended semicomplete and, thus, by Theorem 6.6.1 and Lemma 6.8.6, we may assume that F_i' is an i-path subdigraph of D'. Hence, $mp_i(D') = mpc_i(D')$ and it follows from Lemma 6.8.5(b) that (6.13) holds for D'. Thus it suffices to prove that $mp_i(D) = mp_i(D')$.

Let F_i' be an i-path subdigraph of D' and let p_k denote the number of vertices from $\overline{K}_{n_k}$ which are covered by F_i'. Since all vertices of $\overline{K}_{n_k}$ are similar it follows from (6.15) that we may assume (by making the proper replacements if necessary) that F_i' includes $v_1^k, \ldots, v_{p_k}^k$. For each k, replace the vertices $v_1^k, \ldots, v_{p_k}^k$ in F_i' by a p_k-path subdigraph of Q_k with cost

$mp_{p_k}(Q_k) = \sum_{i=1}^{p_k} c'(v_i^k)$. As a result, we obtain, from F_i', an i-path subdigraph F_i of D for which we have $c'(F_i') = \sum_{k=1}^{t} mp_{p_k}(Q_k) = c(F_i)$ and, thus, $c(F_i) = c'(F_i')$. Reversing the process above it is easy to get, from an i-path subdigraph of D, an i-path subdigraph F_i' of D' such that $c(F_i) = c'(F_i')$. This shows that $mp_i(D) = mp_i(D')$ and hence (6.13) holds for D by the remark above.

We prove the complexity by induction on n. Let m' be the number of arcs in D' and recall that all these arcs are also in D. Clearly when a digraph H has $|V(H)| \leq 2$ we can choose a constant c_1 so that we can determine the numbers $mp_i(H)$, $i = 1, 2, \ldots, |V(H)|$, in time at most $c_1|V(H)|^2(|A(H)| + |V(H)|)$. Now assume by induction that for each Q_i we can determine the desired numbers inside Q_i in time at most $c_1 n_i^2(m_i + n_i)$. This means that we can find the numbers $mp_i(Q_j)$ for all $j \in [t]$ and $i \in [n_j]$ in total time

$$\sum_{j=1}^{t} c_1 n_j^2(m_j + n_j) \leq c_1 n^2 \sum_{j=1}^{t}(m_j + n_j) = c_1 n^2(m - m' + n).$$

By Lemma 6.8.5(a), Theorem 6.6.1 and Lemma 6.8.6, there is a constant c_2 such that in total time at most $c_2 n^2(m' + n)$ we can find, for every $j \in [n]$, a j-path-cycle subdigraph of cost $mp_j(D')$ in D'. It follows from the way we construct F_i above from F_i' that if we are given for each $k \in [t]$ and each $1 \leq j \leq n_k$ a j-path subdigraph in Q_k of cost $mp_j(Q_k)$, then we can construct all the path subdigraphs F_r, $1 \leq r \leq n$, in time at most $c_3 n^3$ for some constant c_3. Hence the total time needed by the algorithm is at most

$$c_1 n^2(m - m' + n) + c_2 n^2(m' + n) + c_3 n^3 =$$
$$c_1 n^2(m + n) + (c_2 - c_1) n^2 m' + (c_2 + c_3) n^3,$$

which is at most $c_1 n^2(m + n)$ for c_1 sufficiently large. □

The next theorem is an easy consequence of Theorem 6.8.7 (assign all vertices cost -1).

Theorem 6.8.8 *One can find a longest path in any quasi-transitive digraph in time $O(n^2 m + n^3)$.* □

Sometimes, one is interested in finding path subdigraphs that include maximum number of vertices from a given set X or avoid as many vertices of X as possible. We consider a minimum cost extension of this problem in the next result.

Theorem 6.8.9 *Let $D = (V, A)$ be a vertex-weighted quasi-transitive digraph and let $X \subseteq V$ be non-empty. Let p_j be the maximum possible number of vertices from X in a j-path subdigraph and let q_j be maximum possible number of vertices from X not in a j-path subdigraph. In total time $O(n^2 m + n^3)$ we can find, for all $j \in [n]$, a cheapest j-path subdigraph which includes p_j (avoids q_j, respectively) vertices of X.*

Proof: Let $C = \sum_{v \in V} |c(v)|$ and subtract $C+1$ from the cost of every vertex in X. Now, for each $j \in [n]$, every cheapest j-path subdigraph F_j must cover as many vertices from X as possible, i.e., p_j vertices. Furthermore, since the new cost of F_j is exactly the original one minus $p_j(C+1)$, cheapest j-path subdigraphs covering p_j vertices from X are preserved under this transformation. Now the 'including' part of the claim follows from Theorem 6.8.7(a). The 'avoiding' part can be proved similarly, by adding $C+1$ to every vertex of X. □

6.8.4 Finding a Cheapest Cycle in a Quasi-Transitive Digraph

Bang-Jensen, Gutin and Yeo obtained the following:

Theorem 6.8.10 *[100] For quasi-transitive digraphs with vertex-weights the minimum cost cycle problem can be solved in time* $O(n^5 \log n)$.

Proof: Let D be a quasi-transitive digraph. If D is not strong, then we simply look at the strong components, so assume that D is strong. By Theorem 2.7.5, $D = T[Q_1, \ldots, Q_t]$, where T is a strong semicomplete digraph, and each Q_i is either a single vertex or a non-strong quasi-transitive digraph.

Suppose we have found a minimum cost cycle C_i in each Q_i which contains a cycle. Then clearly the minimum cost of a cycle in D is the minimum cost cycle among those cycles C_i that exist and the minimum cost of a cycle C which intersects at least two Q_i's. Hence it follows that applying this approach recursively we can find the minimum cost cycle in D. Now we show how to compute a minimum cost cycle C as above.

Let D' be defined as in the proof of Theorem 6.8.7 including the vertex-costs. It is easy to show using the same approach as when we converted between i-path subdigraphs of D' and D in the proof of Theorem 6.8.7, that the cost of C is precisely $mc(D')$. Now it follows from Theorem 6.8.1 that we can find the cycle C in time $O(n^3m + n^4 \log n)$.

Since we can construct D' including finding the costs for all the vertices in time $O(n^2m + n^3)$ by Theorem 6.8.7 and there are at most $O(n)$ recursive calls, the approach above will lead to a minimum cost cycle of D in time $O(n^4m + n^5 \log n)$. In fact, we can bound the first term as we did in the proof of Theorem 6.8.7 and obtain $O(n^3m + n^5 \log n) = O(n^5 \log n)$ rather than $O(n^4m + n^5 \log n)$. This completes the proof. □

6.9 Hamilton Paths and Cycles in Various Classes of Digraphs

Let us start from the following simple observation.

Proposition 6.9.1 *The line digraph $L(D)$ is hamiltonian if and only if D is eulerian.* □

Grötschel and Harary [427] showed that only very few bridgeless graphs have the property that every strong orientation is hamiltonian.

Theorem 6.9.2 *[427] Let G be a bridgeless graph. If G is neither a cycle nor a complete graph, then G contains a strong non-hamiltonian orientation.* □

However, Thomassen proved in the following result that there are many more graphs with the property that every strong orientation is traceable.

Theorem 6.9.3 *[857] Let G be a 2-edge-connected undirected graph such that every connected component of $\overline{G}$ is either bipartite or an odd cycle of length at least 5. Also assume that $\overline{G}$ has at most one non-bipartite component. Then every strong orientation of G is traceable.*

To prove Theorem 6.9.3, we need the following lemma whose proof is left as Exercise 6.32.

Lemma 6.9.4 *Let L be the complement of an odd cycle $u_1u_2\ldots u_{2k+1}u_1$, $k \geq 2$, and let F be an orientation of L. Then, there are $i \neq j \in \{1, 2, \ldots, 2k+1\}$ such that $u_iu_ju_{i+1}$ or $u_{i+1}u_ju_i$ is a path in F.* □

Proof of Theorem 6.9.3: Let $G_1, \ldots, G_r$ be bipartite connected components of $\overline{G}$ such that A_i, B_i are partite sets of G_i, $i \in [r]$. Let $Z = u_1u_2\ldots u_{2k+1}u_1$ be the odd cycle in $\overline{G}$, if one exists.

Let H be a strong orientation of G. Define a partition A, B of $V(G)$ as follows. Let $A^* = A_1 \cup \ldots \cup A_r$ and $B^* = B_1 \cup \ldots \cup B_r$. If Z does not exist (in $\overline{G}$), then $A = A^*$, $B = B^*$. Otherwise, by Lemma 6.9.4, without loss of generality, we have that there exists a j such that $u_1u_ju_2$ is a directed path in H. Let $A = A^* \cup \{u_3, u_5, \ldots, u_{2k+1}\}$, $B = B^* \cup \{u_2, u_4, \ldots, u_{2k}\} \cup \{u_1\}$. By this construction, $H\langle A\rangle$ is a tournament and $H\langle B\rangle$ is either a tournament (if Z does not exist) or H has a path xzy such that $x, y \in B$ and $xy \notin G\langle B\rangle$.

We now show that H has a cycle C including all vertices of A. If $H\langle A\rangle$ is strong, then C exists by Camion's theorem (Corollary 1.5.2). If $H\langle A\rangle$ is not strong, then there is a shortest path P in H from the terminal strong component of $H\langle A\rangle$ to its initial strong component. Let P start at u and terminate at w. (Clearly, P does not have vertices other than u and w in these two components.) It is easy to check that $H\langle(A - V(P)) \cup \{u, w\}\rangle$ has a hamiltonian (w, u)-path Q. The paths P and Q form a cycle containing A. Let C be a longest cycle containing A.

If $H - V(C)$ is a tournament, then some vertex of C dominates a vertex v of the initial strong component of $H - V(C)$. The tournament $H - V(C)$ has a hamiltonian path starting at v; this path can be extended to a hamiltonian path in H. Thus, we may assume that $H - V(C)$ is not a tournament. In particular, $x, y \in V(H) - V(C)$. Let $C = v_1v_2\ldots v_mv_1$. We consider two cases.

Case 1: $z \in V(C)$. We first prove that C contains vertices v_i, v_{i+j} such that v_i dominates one of x, y and v_{i+j} is dominated by the other one and $1 \leq j \leq m-1$. Since $\overline{G}$ has no triangles, each of z^+ and z^- is adjacent to at least one of x, y. By the maximality of C, if z^+ and y are adjacent, we must have $z^+ \rightarrow y$ and then z, z^+ is the desired pair. Hence, we may assume that z^+ is adjacent to x and, hence, either z, z^+ is the desired pair or $z^+ \rightarrow x$. Now considering z^- one can prove that either z^-, z is the desired pair or z^-, z^+ is the desired pair.

Among all pairs v_i, v_{i+j} satisfying the above property choose one such that j is the smallest possible. We may assume (by interchanging x and y if needed) that $v_i \rightarrow x$ and $y \rightarrow v_{i+j}$. We show that $j = 1$. Assume that $j > 1$. Because of the minimality of j, x is not dominated by v_{i+s} when $1 \leq s < j$ and because of the maximality of C, x does not dominate v_{i+1}. Hence, x is not adjacent to v_{i+1}. Similarly, we can see that y is not adjacent to v_{i+j-1} and none of the vertices v_{i+s}, $1 \leq s < j$, is dominated by y. Since $\overline{G}$ has no triangle, $j \geq 3$ and $v_{i+1} \rightarrow y$ and $x \rightarrow v_{i+j-1}$; a contradiction to the minimality of j. Thus, we may assume that $v_i \rightarrow x,\ y \rightarrow v_{i+1}$.

We add to the oriented graph $H - V(C)$ the arc yx obtaining a tournament T. Let v be a vertex in the initial strong component of T dominated by a vertex u in C. By Camion's theorem, T has a hamiltonian path P starting at v and terminating at some vertex w. If yx is not on P, then $C[u^+, u]P$ is a hamiltonian path of H. If yx is on P, then $P[v, y]C[v_{i+1}, v_i]P[x, w]$ is a hamiltonian path of H.

Case 2: $z \notin V(C)$. If $H - V(C)$ is strong, then we consider any arc of H between x and C (such an arc exists as the degree of x in $\overline{G}$ equals 2). If this arc starts (terminates) at x, we add to $H - V(C)$ the arc xy (yx) and consider a hamiltonian cycle in the resulting tournament. Using this together with C and the arc between x and C, it is easy to find a hamiltonian path in H.

So we assume that $H - V(C)$ is not strong. Let $H_1, H_2, \ldots, H_p$ be an acyclic ordering of strong components of $H - V(C)$. We may assume without loss of generality (consider the converse of H if needed) that at most one of x, y belongs to $V(H_1)$. Clearly, some vertex v in H_1 is dominated by a vertex in C. We can find a hamiltonian path in H as in the case when $H - V(C)$ is a tournament unless for some i, $V(H_i) = \{x\}$ and $V(H_{i+1}) = \{y\}$ or $V(H_{i-1}) = \{y\}$. But this is impossible due to the existence of xzy. □

In this theorem it is important that $\overline{G}$ does not contain a 3-cycle. Indeed, let M be a multipartite tournament consisting of a strong tournament T with fixed vertex y and triple x_1, x_2, x_3 of independent vertices such that $N^+(x_i) = \{y\}$ for every $i = 1, 2, 3$. Since $|N^+(\{x_1, x_2, x_3\})| < 2$ (see Exercise 4.71), M has no 1-path-cycle factor. (Recall that a multipartite tournament is traceable if and only if it has a 1-path-cycle factor, see Theorem 6.6.1.) However, Thomassen [857] remarks that Theorem 6.9.3 is perhaps far from being the best possible. He claims that by using the method of the proof

of this theorem, it is not difficult to show that any strong orientation of a graph, whose complement is a disjoint union of two 5-cycles and independent vertices, has a hamiltonian path.

Problem 6.9.5 *Find a non-trivial extension of Theorem 6.9.3.*

We recall that a digraph D is unilateral if for every pair x, y of distinct vertices of D there is a path between x and y (not necessarily both (x, y)-path and (y, x)-path). For some of the graphs in Theorem 6.9.3 not only all strong orientations are traceable, but also all unilateral ones satisfy this property. This was shown by Fink and Lesniak-Foster in the following theorem.

Theorem 6.9.6 *[315] Let G be a graph and let $\mathcal{F} = Q_1 \cup \ldots \cup Q_k$ be a path subgraph of G in which every path Q_i is of length 1 or 2. Then an orientation of $G - \cup_{i=1}^{k} E(Q_i)$ is traceable if and only if it is unilateral.* □

Erdős and Trotter [301] investigated when the Cartesian product of two directed cycles is hamiltonian. They proved the following (gcd means the greatest common divisor):

Theorem 6.9.7 *Let $d = \gcd(k, m)$. The Cartesian product $\vec{C}_k \times \vec{C}_m$ is hamiltonian if and only if $d \geq 2$ and there exist positive integers d_1, d_2 such that $d_1 + d_2 = d$ and $\gcd(k, d_1) = \gcd(m, d_2) = 1$.* □

For a generalization of Theorem 6.9.7, see Theorem 15.6.2.

In Section 2.5, we introduced de Bruijn digraphs $D_B(d, t)$, Kautz digraphs $D_K(d, t)$ as well as their generalizations: $D_G(d, n)$, $D_I(d, n)$, $D(d, n, q, r)$. The consecutive-d digraphs $D(d, n, q, r)$ are the most general among the digraphs listed above. Thus, we restrict our attention to these digraphs. Du, Hsu and Hwang [278] proved the following result for digraphs $D(d, n, q, r)$.

Theorem 6.9.8 *If $\gcd(n, q) \geq 2$, or $\gcd(n, q) = 1$ and $q \geq 5$, then $D(d, n, q, r)$ is hamiltonian.* □

Hwang [545] as well as Du and Hsu [277] characterized hamiltonian digraphs $D(d, n, q, r)$ for $\gcd(n, q) = 1$ and $d = 1$ ($d = 2$, respectively). Chang, Hwang and Tong [195] showed that every digraph $D(4, n, q, r)$ is hamiltonian. They also gave examples of digraphs $D(3, n, q, r)$, which are not hamiltonian [194].

Several authors considered hamiltonicity for circulant digraphs. In particular, Rankin [763] proved the following classic result:

Theorem 6.9.9 *A strong circulant digraph $C_n(a, b)$ is hamiltonian if and only if there are two non-negative integers s and t, such that $s + t = \gcd(sa + tb, n) = \gcd(a - b, n)$.* □

Recall that according to Part (c) of Proposition 2.14.1 $C_n(a_1, a_2, \ldots, a_p)$ is strong if and only if $\gcd(n, a_1, a_2, \ldots, a_p) = 1$. There is no characterization of hamiltonian circulant digraphs $C_n(S)$ with $|S| \geq 3$ [568, 650]. Locke and Witte [650] gave an infinite family of non-hamiltonian circulant digraphs of out-degree 3. Curran and Witte [235] obtained sufficient conditions for a circulant digraph of out-degree at least 3 to be hamiltonian and stated the following conjecture:

Conjecture 6.9.10 *Suppose that $C_n(S)$ is strong and $|S| \geq 3$. If $C_n(B)$ is not strong for every proper subset B of S, then $C_n(S)$ is hamiltonian.*

Jirásek [568] showed that for infinitely many non-hamiltonian circulant digraphs constructed by Locke and Witte [650] the reversal of any arc produces a Hamilton cycle. This solved a problem of C. Thomassen stated in [169], whether there is a non-hamiltonian oriented graph in which the reversal of any arc results in a hamiltonian graph.

In Section 2.14.2 we introduced arc-locally semicomplete (ALS) digraphs and formulated the following characterization of strong ALS digraphs by Bang-Jensen: a strong ALS digraph is either semicomplete or semicomplete bipartite or an extended cycle. Clearly, an extended cycle is hamiltonian if and only if it has a cycle factor. Thus, this characterization and Theorems 1.5.3 and 6.6.4 imply the following:

Corollary 6.9.11 *[75] An ALS digraph is hamiltonian if and only if it is strong and contains a cycle factor.* □

Motivated by arc-locally semicomplete digraphs, Bang-Jensen [75] introduced the following family of digraphs, which we will call Q-digraphs. A digraph D is a **Q-digraph** if the following condition holds for every four distinct vertices x, y, z, w of D: if xy, zy and zw are arcs, then vertices x and w are adjacent. There are Q-digraphs which are neither semicomplete digraphs nor semicomplete bipartite digraphs nor extended cycles, see Figure 6.7.

Nevertheless, Bang-Jensen [75] raised the following two conjectures:

Conjecture 6.9.12 *A Q-digraph is hamiltonian if and only if it is strong and has a cycle factor.*

Conjecture 6.9.13 *The hamiltonian path and cycle problems are polynomial time solvable for Q-digraphs.*

Generalizing the notion of f-connectivity of undirected graphs introduced in [179], Bang-Jensen and Brandt [77] came up with the following notion. Let f be a monotone increasing function $f : \mathbb{Z}_+ \to \mathbb{R}$ (i.e., $f(x) \leq f(y)$ for each $x < y$). A digraph $D = (V, A)$ is **f-expanding** if for every $\emptyset \neq X \subset V$ we have the following: if $|X| \leq |V - N^+[X]|$, then $|N^+(X)| \geq f(|X|)$, and if $|X| \leq |V - N^-[X]|$, then $|N^-(X)| \geq f(|X|)$. It is not difficult to show that every f-expanding digraph D is $\lceil f(1) \rceil$-strong (Exercise 6.38) and, thus, if $f(1) > 0$, then D is strong. Bang-Jensen and Brandt [77] proved the following:

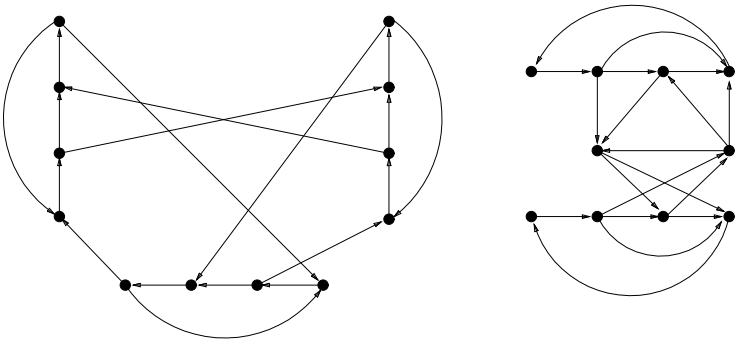


Figure 6.7 Two hamiltonian Q-digraphs.

Theorem 6.9.14 *If $f(k) \geq 3^{k+1}$, then every f-expanding digraph of sufficiently large order is hamiltonian.* □

Bang-Jensen and Brandt believe that some degree of polynomial expansion suffices for guaranteeing hamiltonicity of digraphs.

Conjecture 6.9.15 *[77] There exists a natural number r so that if $f(k) \geq k^r$, then every f-expanding digraph is hamiltonian.*

Bang-Jensen and Brandt [77] proved that if $f(k) \geq ck$, $c > 0$ and D is an f-expanding digraph of order n, then D contains a path of length $\frac{c}{c+2}n$ and a cycle of length at least $n(\frac{c}{c+2})^2$.

For a digraph $D = (V, A)$, a set $S \subseteq V$ is called a **q^+-set** (**q^--set**, respectively) if S has at least two vertices and, for every $u \in S$, there exists $v \in S, v \neq u$ such that $N^+(u) \cap N^+(v) \neq \emptyset$ ($N^-(u) \cap N^-(v) \neq \emptyset$, respectively). A digraph D is called **s-quadrangular** if, for every q^+-set S, we have

$$|\cup \{N^+(u) \cap N^+(v) : u \neq v,\ u, v \in S\}| \geq |S|$$

and, for every q^--set S, we have

$$|\cup \{N^-(u) \cap N^-(v) : u \neq v,\ u, v \in S)\}| \geq |S|.$$

Gutin, Jones, Rafiey, Severini and Yeo formulated the following:

Conjecture 6.9.16 *[464] Every strong s-quadrangular digraph is hamiltonian.*

It was shown by Severini [807] that the digraph of a unitary matrix is s-quadrangular. It follows that if Conjecture 6.9.16 is true, then the digraph of an irreducible unitary matrix is hamiltonian. Unitary matrices are important in quantum mechanics and, at present, are central in the theory of quantum computation [728].

It was proved in [464] that Conjecture 6.9.16 holds for each strong s-quadrangular digraph D with $\Delta^0(D) \leq 3$ and every strong s-quadrangular digraph D has a cycle factor.

6.10 Exercises

6.1. Using the proof of Theorem 6.2.2, Lemma 6.2.1 and Proposition 2.8.3, prove Corollary 6.2.3.

6.2. Prove that every strong locally in-semicomplete digraph has a 2-connected underlying graph.

6.3. Give a direct proof of the following result. A locally semicomplete digraph has a hamiltonian cycle if and only if it is strong (Bang-Jensen [66]).

6.4. Give a direct proof of the following result. A locally semicomplete digraph has a hamiltonian path if and only if it is connected (Bang-Jensen [66]). Hint: use Proposition 2.9.2.

6.5. Give a direct proof of the following result. One can find a longest cycle in a semicomplete digraph in time $O(n^2)$ (Manoussakis [681]). Hint: start by finding a hamiltonian path P and show that using P we can construct a hamiltonian cycle in the desired time.

6.6. Using Proposition 6.1.5 and Theorem 6.4.1 prove the following:

Proposition 6.10.1 *Let D be a digraph of order n. Suppose that, for every dominated pair of non-adjacent vertices $\{x, y\}$, either $d(x) \geq n-1$ and $d(y) \geq n-2$ or $d(x) \geq n-2$ and $d(y) \geq n-1$. Then D is traceable.*

6.7. Prove that the digraph Q_n introduced before Theorem 6.4.1 is strong and non-hamiltonian.

6.8. Prove Lemma 6.4.21.

6.9. Find an infinite family of hamiltonian digraphs that satisfy the conditions of both Theorems 6.4.1 and 6.4.5, but do not satisfy the conditions of Theorem 6.4.7 and are neither locally out-semicomplete nor locally in-semicomplete (Bang-Jensen, Gutin and Li [94]).

6.10. Find an infinite family of hamiltonian digraphs that satisfy the conditions of Theorem 6.4.12, but do not satisfy the conditions of Theorem 6.4.7 (Zhao and Meng [933]).

6.11. Prove Corollaries 6.4.18 and 6.4.19.

6.12. Using Meyniel's theorem, prove that if a strong digraph D has at least $n^2 - 3n + 5$ arcs, then D is hamiltonian (Lewin [641]).

6.13. Prove that every digraph with more than $(n-1)^2$ arcs is hamiltonian (Lewin [641]).

6.14. Prove that if the minimum semi-degree of a digraph D of order n is at least $(n+1)/2$, then every arc of D is contained in a Hamilton cycle of D.

6.15. Construct an infinite family of non-hamiltonian strong digraphs that satisfy both (6.1) and (6.2) (Bermond and Thomassen [152]).

6.16. Let $P = v_1 v_2 \dots v_p$ be a longest path in an oriented graph D. Prove that if $d^-(v_1) > 0$, then D contains a cycle of length at least $d^-(v_1) + 2$. Deduce from this that every oriented graph D of positive minimum in-degree contains a cycle of length at least $\delta^-(D) + 2$ (Jackson [555]).

6.17. For each integer $k \geq 1$, construct oriented graphs of minimum in-degree k that have no cycle of length greater than $k+2$ (Jackson [555]).

6.18. Prove that every vertex of a semicomplete multipartite digraph D belongs to a longest path in D (Volkmann [888]).

6.19. **(+)** Give a direct proof of the first (non-algorithmic) part of Theorem 6.6.1 (Gutin [448, 452]).

6.20. Show that the multipartite tournament in Figure 6.5 is non-hamiltonian.

6.21. Show that the analogue of Theorem 6.6.7 for semicomplete bipartite digraphs does not hold, i.e., there are a strong semicomplete bipartite digraph D and a maximum cycle subdigraph $\mathcal{F}$ in D such that $D\langle V(\mathcal{F})\rangle$ is not hamiltonian (Gutin [451]).

6.22. Prove Theorem 6.6.13 by induction on t.

6.23. By inspecting all intermediate steps in the proof of Corollary 6.6.16, show that the following statement holds. Let D be a bipartite digraph obtained by taking two disjoint even cycles $C = u_1u_2 \dots u_{2k-1}u_{2k}u_1$ and $Z = v_1v_2 \dots v_{2r-1}v_{2r}v_1$ and adding an arc between v_{2i-1} and u_{2j} and between v_{2i} and u_{2j-1} (in any direction, possibly one in each direction) for all $i \in [k]$ and $j \in [r]$. D is hamiltonian if and only if it is strong. Moreover, if D is strong, then, given cycles C and Z as above, a hamiltonian cycle of D can be found in time $O(|V(C)||V(Z)|)$ (Gutin [451]).

6.24. Prove Theorem 6.6.7.

6.25. Prove the following proposition. Let D be a strong semicomplete multipartite digraph of order n and let r be the cardinality of minimum partite set of D. If for each pair of dominated non-adjacent vertices x, y, $d(x) + d(y) \geq \min\{2(n-r)+3, 2n-1\}$, then D is hamiltonian (Zhou and Zhang [934]).

6.26. Prove that every oriented graph of minimum in-degree and out-degree $k \geq 2$, on at most $2k+2$ vertices, is a multipartite tournament with at most two vertices in each partite set.

6.27. Prove the following theorem due to Jackson:

Theorem 6.10.2 *[555] Every oriented graph of minimum in-degree and out-degree $k \geq 2$, on at most $2k+2$ vertices, is hamiltonian.*

6.28. Check that $f(G'_k, k) = k-1$, where the digraph G'_k and the function f are introduced after Theorem 6.6.26.

6.29. **Characterization of traceable quasi-transitive digraphs.** Prove Theorem 6.7.2 using Theorem 6.6.1. Hint: see the proof of Theorem 6.7.1.

6.30. **Another characterization of traceable quasi-transitive digraphs.** Formulate and prove a characterization of traceable quasi-transitive digraphs similar to Theorem 6.7.3.

6.31. Prove that if D is a non-strong quasi-transitive digraph with a hamiltonian path, then $U\overline{G}(D)$ is not connected.

6.32. Prove Lemma 6.9.4.

6.33. Prove that if D is a strong oriented graph of order at least three and D does not contain, as induced subdigraph, any digraph in Figure 6.8, then

D is hamiltonian (Kemnitz and Greger [590]). Hint : show that D is locally out-semicomplete and use the characterization of hamiltonian locally out-semicomplete digraphs (Gutin and Yeo [477]).

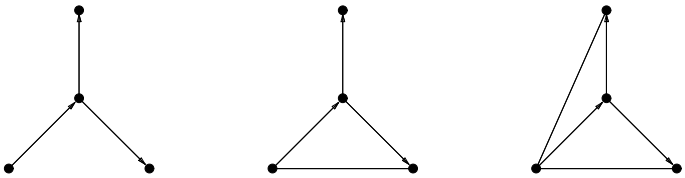


Figure 6.8 Forbidden digraphs in Exercise 6.33. Unoriented arcs can be oriented arbitrarily.

6.34. **A counterexample to a conjecture from** [590]. Consider the tournament D with $V(D) = \{x_1, x_2, x_3, x_4, x_5\}$ and

$$A(D) = \{x_1x_2, x_2x_3, x_3x_4, x_4x_5, x_5x_1, x_1x_3, x_2x_4, x_3x_5, x_4x_1, x_5x_2\}$$

and any 2-strong tournament T, containing three vertices y_1, y_2, y_3 such that

$$\{y_1y_2, y_2y_3, y_3y_1\} \subseteq A(T).$$

Let us construct an oriented graph T^* with vertex set $V(D) \cup V(T)$ and arc set

$$A(D) \cup A(T) \cup \{y_1x_2, x_4y_1, y_2x_2, x_4y_2, y_3x_4, x_2y_3\}.$$

Prove that

(a) T^* is strong.
(b) T^* does not contain, as induced subdigraph, any orientation of $K_{1,3}$.
(c) For every vertex v in T^*, $T^*\langle N(v)\rangle$ is strong.
(d) T^* is not hamiltonian.

(Gutin and Yeo [477])

6.35. **Connected (g, f)-factors in some semicomplete multipartite digraphs.** Given a digraph D and two positive integers $f(x), g(x)$ for every $x \in V(D)$, a subgraph H of D is called a **(g, f)-factor** if $g(x) \leq d^+_H(x) = d^-_H(x) \leq f(x)$ for every $x \in V(D)$. If $f(x) = g(x) = 1$ for every x, then a connected (g, f)-factor is a hamiltonian cycle. Prove the following result by Gutin [459]:

Theorem 6.10.3 *Let D be a semicomplete bipartite digraph or an extended locally in-semicomplete digraph. Then D has a connected (g, f)-factor if and only if D is strongly connected and contains a (g, f)-factor. One can check whether D has a connected (g, f)-factor in $O(n^3)$ time.* □

6.36. **Connected (g, f)-factors in quasi-transitive digraphs.** The additional terminology used in this exercise is introduced in the previous exercise. Prove the following assertion. The connected (g, f)-factor problem is polynomial time solvable for quasi-transitive digraphs (Gutin [459]).

6.37. Formulate and prove a 'cycle' analog of Theorem 6.8.9.

6.38. Prove that every f-expanding digraph is $\lceil f(1) \rceil$-strong.

6.39. Prove that every strong s-quadrangular digraph D has a cycle factor. Hint: use Proposition 4.11.7(b) (Gutin et al. [464]).

7. Restricted Hamiltonian Paths and Cycles

In this chapter we discuss results on hamiltonian paths and cycles with special properties. We start by studying hamiltonian paths with one or more end-vertices prescribed, that is, we study paths which start in a given vertex, paths which connect two prescribed vertices and, finally, paths which start and end in specified vertices. Not surprisingly, the level of difficulty of these problems increases when we fix more and more end-vertices. Even for tournaments the last problem is still not completely solved.

The next topic covered is hamiltonian cycles which either avoid or contain certain prescribed arcs. These problems are very difficult even for tournaments. As we will show in Section 7.4, some of these results imply that the problem of deciding the existence of a hamiltonian cycle in a digraph obtained from a semicomplete digraph by adding just a few new vertices and some arcs is already very difficult. In fact, the problem is highly non-trivial, even if we add just one extra vertex.

The last topic covered in the chapter is orientations of hamiltonian cycles. We discuss in some detail one of the main tools in a proof by Havet and Thomassé of the deep result that every tournament on at least eight vertices contains every orientation of a hamiltonian undirected path.

7.1 Hamiltonian Paths with a Prescribed End-Vertex

We begin with hamiltonian paths starting or ending at a prescribed vertex. Besides being of independent interest, results of this type are also useful in connection with results on hamiltonian paths with both end-vertices prescribed (but the direction of the path is not necessarily given).

To get a feeling for arguing with extended tournament structure, we start with the following easy result.

Proposition 7.1.1 *Suppose that a strong extended tournament D has an (x,y)-path P such that $D-P$ has a cycle factor. Then D has a hamiltonian path starting at x and a hamiltonian path ending at y.*

Proof: Choose a path P' starting at x to be as long as possible so that $D-P'$ has a cycle factor consisting of the cycles $C_1, C_2, \ldots, C_q$, $q \geq 0$. By

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_7,

Proposition 6.6.18, we may assume that $C_i \Rightarrow C_j$ when $1 \leq i < j \leq q$. Let $P' = u_1 u_2 \dots u_r$ where $u_1 = x$. If $q \neq 0$, then, by the choice of P', u_r is completely dominated by C_1. Since D is strong, there is an arc from P' to C_1. Let u_i be the vertex of P' with largest index $i < r$ such that there is an arc $u_i z$ from u_i to C_1 and let z^- be the predecessor of z on C_1. Since u_{i+1} has no arc to C_1, we obtain $z^- \rightarrow u_{i+1}$. Here we used the property that nonadjacent vertices of an extended semicomplete digraph are similar (defined in Chapter 1). Hence $C_1[z, z^-]$ can be inserted between u_i and u_{i+1}, contradicting the choice of P'. So $q = 0$ and P' is a hamiltonian path starting at x. An analogous argument can be applied to show that D has a hamiltonian path ending at y. □

The following result, due to Bang-Jensen and Gutin, shows that, for digraphs that are either semicomplete bipartite or extended locally semicomplete, there is a nice necessary and sufficient condition for the existence of a hamiltonian path starting at a prescribed vertex.

Theorem 7.1.2 *[90] Let $D = (V, A)$ be a digraph which is either semicomplete bipartite or extended locally out-semicomplete and let $x \in V$. Then D has a hamiltonian path starting at x if and only if D contains a 1-path-cycle factor $\mathcal{F}$ of D such that the path of $\mathcal{F}$ starts at x, and, for every vertex y of $V - \{x\}$, there is an (x, y)-path[1] in D. Moreover, if D has a hamiltonian path starting at x, then, given a 1-path-cycle factor $\mathcal{F}$ of D such that the path of $\mathcal{F}$ starts at x, the desired hamiltonian path can be found in time $O(n^2)$.*

Proof: As the necessity is clear, we will only prove the sufficiency. Suppose that $\mathcal{F} = P \cup C_1 \cup ... \cup C_t$ is a 1-path-cycle factor of D that consists of a path P starting at x and cycles C_i, $i = 1, ..., t$. Suppose also that every vertex of D is reachable from x. Then, without loss of generality, there is a vertex of P that dominates a vertex of C_1. Let $P = x_1 x_2 \dots x_p$, $C_1 = y_1 y_2 \dots y_q y_1$, where $x = x_1$ and $x_k \rightarrow y_s$ for some $k \in [p]$, $s \in [q]$. We show how to find a new path starting at x which contains all the vertices of $V(P) \cup V(C_1)$. Repeating this process we obtain the desired path. Clearly, we may assume that $k < p$ and that x_p has no arc to $V(C_1)$.

Assume first that D is an extended locally out-semicomplete digraph. If P has a vertex x_i which is similar to a vertex y_j in C_1, then $x_i y_{j+1}, y_j x_{i+1} \in A$ and using these arcs we see that $P[x_1, x_i]C[y_{j+1}, y_j]P[x_{i+1}, x_p]$ is a path starting from x and containing all the vertices of $P \cup C_1$. If P has no vertex that is similar to a vertex in C_1, then we can apply the result of Exercise 2.38 to $P[x_k, x_p]$ and $x_k C_1[y_s, y_{s-1}]$ and merge these two paths into a path R starting from x_k and containing all the vertices of $P[x_k, x_p] \cup C_1$. Now, $P[x_1, x_{k-1}]R$ is a path starting at x and containing all the vertices of $P \cup C_1$.

Suppose now that D is semicomplete bipartite. Then either $y_{s-1} \rightarrow x_{k+1}$, which implies that $P[x_1, x_k]C_1[y_s, y_{s-1}]P[x_{k+1}, x_p]$ is a path starting at x

[1] This is equivalent to saying that D has an out-branching with root x.

and covering all the vertices of $P \cup C_1$, or $x_{k+1} \rightarrow y_{s-1}$. In the latter case, we consider the arc between x_{k+2} and y_{s-2}. If $y_{s-2} \rightarrow x_{k+2}$, we can construct the desired path, otherwise we continue to consider arcs between x_{k+3} and y_{s-3} and so on. If we do not construct the desired path in this way, then we find that the last vertex of P dominates a vertex in C_1, contradicting our assumption above.

Using the process above and breadth-first search, one can construct an $O(n^2)$ algorithm for finding the desired hamiltonian path starting at x. □

Just as the problem of finding a minimum path factor generalizes the hamiltonian path problem, we may generalize the problem of finding a hamiltonian path starting at a certain vertex to the problem of finding a path factor with as few paths as possible such that one of these paths starts at a specified vertex x. We say that a path factor **starts at** $\boldsymbol{x}$ if one of its paths starts at x and denote by $\mathrm{pc}_x(D)$ the minimum number of paths in a path factor that starts at x. The problem of finding a path factor with $\mathrm{pc}_x(D)$ paths which starts at x in a digraph D is called the PFx PROBLEM[2].

Let Φ_1 be the union of all semicomplete bipartite, extended locally semicomplete and acyclic digraphs. Using an approach similar to that taken in Section 6.8, Bang-Jensen and Gutin proved the following.

Theorem 7.1.3 *[90] Let D be a totally Φ_1-decomposable digraph. Then the PFx problem for D can be solved in time $O(|V(D)|^4)$.* □

7.2 Weakly Hamiltonian-Connected Digraphs

Recall that an $[x, y]$-path in a digraph $D = (V, A)$ is a path which either starts at x and ends at y or oppositely. We say that D is **weakly hamiltonian-connected** if it has a hamiltonian $[x, y]$-path (also called an **$[x, y]$-hamiltonian path**) for every choice of distinct vertices $x, y \in V$. Obviously, deciding whether a digraph contains an $[x, y]$-hamiltonian path for some x, y is not easier than determining whether D has any hamiltonian path and hence for general digraphs this is an $\mathcal{NP}$-complete problem by Theorem 6.1.2 (see also Exercise 7.2). In this section we discuss various results that have been obtained for generalizations of tournaments. All of these results imply polynomial algorithms for finding the desired paths.

7.2.1 Results for Extended Tournaments

We start with a theorem, due to Thomassen [856], which has been generalized to several super-classes of tournaments as will be seen in the following subsections.

[2] Observe that $pc(D) \leq \mathrm{pc}_x(D) \leq \mathrm{pc}(D) + 1$ holds for every digraph D.

Theorem 7.2.1 *[856] Let $D = (V, A)$ be a tournament and let x_1, x_2 be distinct vertices of D. Then D has an $[x_1, x_2]$-hamiltonian path if and only if none of the following holds.*

(a) D is not strong and either none of x_1, x_2 belongs to the initial strong component of D or none of x_1, x_2 belongs to the terminal strong component (or both).
(b) D is strong and for $i = 1$ or 2, $D - x_i$ is not strong and x_{3-i} belongs to neither the initial nor the terminal strong component of $D - x_i$.
(c) D is isomorphic to one of the two tournaments in Figure 7.1 (possibly after interchanging the names of x_1 and x_2).

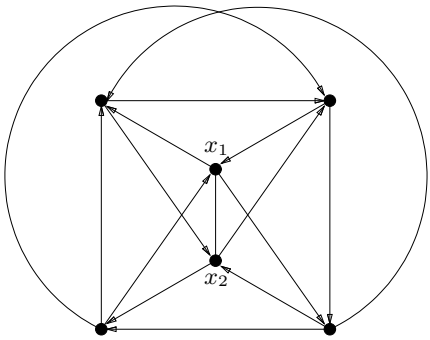


Figure 7.1 The exceptional tournaments in Theorem 7.2.1. The edge between x_1 and x_2 can be oriented arbitrarily.

The following easy corollary is left as Exercise 7.4:

Corollary 7.2.2 *[856] Let D be a strong tournament and let x, y, z be distinct vertices of D. Then D has a hamiltonian path connecting two of the vertices in the set $\{x, y, z\}$.* □

Thomassen [856] used a nice trick in his proof of Theorem 7.2.1 by using Corollary 7.2.2 in the induction proof. We will give his proof below.

Proof of Theorem 7.2.1: Let x_1, x_2 be distinct vertices in a tournament D. It is easy to check that if any of (a)-(c) holds, then there is no $[x_1, x_2]$-hamiltonian path in D.

Suppose now that none of (a)-(c) hold. We prove, by induction on n, that D has an $[x_1, x_2]$-hamiltonian path. This is easy to show when $n \leq 4$, so assume now that $n \geq 5$ and consider the induction step with the obvious induction hypothesis. If D is not strong, then let $D_1, D_2, \ldots, D_s$, $s \geq 2$, be the acyclic ordering of the strong components of D. Since (a) does not hold,

we may assume without loss of generality that $x_1 \in V(D_1)$ and $x_2 \in V(D_s)$. Observe that D_1 has a hamiltonian path P_1 starting at x_1 and D_s has a hamiltonian path P_s ending at x_2. Let P_i be a hamiltonian path in D_i for each $i = 2, 3, \ldots, s-1$. Then $P_1P_2 \ldots P_{s-1}P_s$ is an (x_1, x_2)-hamiltonian path.

If $D - x_i$ is not strong for $i = 1$ or 2, then we may assume without loss of generality that $i = 1$. Let $D'_1, \ldots, D'_p$, $p \geq 2$, be the acyclic ordering of the strong components of $D - x_1$. Since (b) does not hold we may assume, by considering the converse of D if necessary, that x_2 belongs to D'_p. Let y be any out-neighbour of x_1 in D'_1. Our argument for the previous case implies that there is a (y, x_2)-hamiltonian path P in $D - x_1$, implying that x_1P is an (x_1, x_2)-hamiltonian path in D. Hence we may assume that $D - x_i$ is strong for $i = 1, 2$.

If $D - \{x_1, x_2\}$ is not strong, then it is easy to prove that D has an (x_i, x_{3-i})-hamiltonian path for $i = 1, 2$ (Exercise 7.1). Hence we only need to consider the case when $D' = D - \{x_1, x_2\}$ is strong. Let $u_1u_2 \ldots u_{n-2}u_1$ be a hamiltonian cycle of D'. By considering the converse if necessary, we may assume that x_1 dominates u_1. Then D has an (x_1, x_2)-hamiltonian path unless x_2 dominates u_{n-2} so we may assume that is the case. By the same argument we see that either the desired path exists or x_1 dominates u_{n-3} and x_2 dominates u_{n-4}. Now it is easy to see that either the desired path exists, or $n-2$ is even and we have $x_1 \mapsto \{u_1, u_3, \ldots, u_{n-3}\}$, $x_2 \mapsto \{u_2, u_4, \ldots, u_{n-2}\}$. If x_1 or x_2 dominates any vertex other than those described above, then, by repeating the argument above, we see that either the desired path exists or $\{x_1, x_2\} \mapsto V(C)$, which is impossible since D is strong. Hence we may assume that

$$\begin{aligned} \{u_2, u_4, \ldots, u_{n-2}\} &\mapsto x_1 \mapsto \{u_1, u_3, \ldots, u_{n-3}\}, \\ \{u_1, u_3, \ldots, u_{n-3}\} &\mapsto x_2 \mapsto \{u_2, u_4, \ldots, u_{n-2}\}. \end{aligned} \tag{7.1}$$

If $n = 6$, then using that (c) does not hold, it is easy to see that the desired path exists. So we may assume that $n \geq 8$. By induction, the theorem and hence also Corollary 7.2.2 holds for all tournaments on $n-2$ vertices. Thus D' has a hamiltonian path P which starts and ends in the set $\{u_1, u_3, u_5\}$ and by (7.1), P can be extended to an (x_1, x_2)-hamiltonian path of D. □

We now turn to extended tournaments. An extended tournament D does not always have a hamiltonian path, but, as we saw in Theorem 6.6.1, it does when the following obviously necessary condition is satisfied: there is a 1-path-cycle factor in D. Thus if we are looking for a sufficient condition for the existence of an $[x, y]$-hamiltonian path, we must require the existence on an $[x, y]$-path P such that $D - P$ has a cycle factor (this includes the case when P is already hamiltonian). Checking for such a path factor in an arbitrary digraph can be done in polynomial time using flows, see Exercise 7.3.

The next result is similar to the structure we found in the last part of the proof of Theorem 7.2.1.

Lemma 7.2.3 *[92] Suppose that D is a strong extended tournament containing two adjacent vertices x and y such that $D - \{x, y\}$ has a hamiltonian cycle C but D has no hamiltonian $[x, y]$-path. Then C is an even cycle, $N^+(x) \cap V(C) = N^-(y) \cap V(C)$, $N^-(x) \cap V(C) = N^+(y) \cap V(C)$, and the neighbours of x alternate between in-neighbours and out-neighbours around C.*

Proof: Exercise 7.5. □

Bang-Jensen, Gutin and Huang obtained the following characterization for the existence of an $[x, y]$-hamiltonian path in an extended tournament. Note the strong similarity with Theorem 7.2.1.

Theorem 7.2.4 *[92] Let D be an extended tournament and x_1, x_2 be distinct vertices of D. Then D has an $[x_1, x_2]$-hamiltonian path if and only if D has an $[x_1, x_2]$-path P such that $D - P$ has a cycle factor and D does not satisfy any of the conditions below:*

(a) D is not strong and either the initial or the terminal component of D (or both) contains none of x_1 and x_2;
(b) D is strong and the following holds for $i = 1$ or $i = 2$: $D - x_i$ is not strong and either x_{3-i} belongs to neither the initial nor the terminal component of $D - x_i$, or x_{3-i} does belong to the initial (terminal) component of $D - x_i$ but there is no (x_{3-i}, x_i)-path ((x_i, x_{3-i})-path) P' such that $D - P'$ has a cycle factor.
(c) $D, D - x_1$ and $D - x_2$ are all strong and D is isomorphic to one of the tournaments in Figure 7.1. □

The proof of this theorem in [92] is constructive and implies the following result (the proof is much more involved than that of Theorem 7.2.1). We point out that the proof in [92] makes explicit use of the fact that the digraphs have no 2-cycles. Hence the proof is only valid for extended tournaments and not for general extended semicomplete digraphs, for which the problem is still open.

Theorem 7.2.5 *[92] There exists an $O(\sqrt{n}m)$ algorithm to decide if a given extended tournament has a hamiltonian path connecting two specified vertices x and y. Furthermore, within the same time bound a hamiltonian $[x, y]$-path can be found if it exists.* □

Theorem 7.2.4 implies the following characterization of extended tournaments which are weakly hamiltonian-connected (see Exercise 7.7).

Theorem 7.2.6 *[92] Let D be an extended tournament. Then D is weakly hamiltonian-connected if and only if it satisfies each of the conditions below.*

(a) D is strongly connected.
(b) For every pair of distinct vertices x and y of D, there is an $[x, y]$-path P such that $D - P$ has a cycle factor.
(c) For each vertex x of D, $D - x$ has at most two strong components and if $D - x$ is not strong, then for each vertex y in the initial (respectively terminal) strong component, there is a (y, x)-path (respectively an (x, y)-path) P' such that $D - P'$ has a cycle factor.
(d) D is not isomorphic to any of the two tournaments in Figure 7.1. □

The following result generalizes Corollary 7.2.2. Note that we must assume the existence of the paths described below in order to have any chance of having a hamiltonian path with end-vertices in the set $\{x, y, z\}$. The proof below illustrates how to argue with extended tournament structure.

Corollary 7.2.7 *[92] Let x, y and z be three vertices of a strong extended tournament D. Suppose that, for every choice of distinct vertices $u, v \in \{x, y, z\}$, there is a $[u, v]$-path P in D so that $D - P$ has a cycle factor. Then there is a hamiltonian path connecting two of the vertices in $\{x, y, z\}$.*

Proof: If both $D - x$ and $D - y$ are strong, then, by Theorem 7.2.4, either D has a hamiltonian path connecting x and y, or D is isomorphic to one of the tournaments in Figure 7.1, in which case there is a hamiltonian path connecting x and z. There is a similar argument if both $D - x$ and $D - z$, or $D - y$ and $D - z$ are strong. So, without loss of generality, assume that neither $D - x$ nor $D - y$ is strong. Let $S_1, S_2, \ldots, S_t$ be an acyclic ordering of the strong components of $D - x$. Note that S_t has an arc to x, since D is strong.

Suppose first that $y \in V(S_i)$ for some $1 < i < t$. We show that this implies that $D - y$ is strong, contradicting our assumption. Consider an $[x, y]$-path P and a cycle factor $\mathcal{F}$ of $D - P$. It is easy to see that P cannot contain any vertex of $S_{i+1}, \ldots, S_t$. Hence each of these strong components contains a cycle factor consisting of those cycles from $\mathcal{F}$ that are in S_j for $j = i+1, \ldots, t$. In particular (since it contains a cycle), each S_j has size at least 3 for $j = i+1, \ldots, t$. It also follows from the existence of P and $\mathcal{F}$ that every vertex in S_i is dominated by at least one vertex from $U = V(S_1) \cup \ldots \cup V(S_{i-1})$. Indeed, if some vertex $z \in V(S_i)$ is not dominated by any vertex from U, then using that $S_r \Rightarrow S_p$ for all $1 \leq r < p \leq t$ we get that z is similar to all vertices in U. However, this contradicts the existence of P and $\mathcal{F}$. Now it is easy to see that $D - y$ is strong since every vertex of $S_i - y$ is dominated by some vertex from $V(S_1) \cup \ldots \cup V(S_{i-1})$ and dominates a vertex in $V(S_{i+1}) \cup \ldots \cup V(S_t)$. Hence we may assume that y belongs to S_1 or S_t.

By considering the converse of D if necessary, we may assume that $y \in V(S_1)$. By Theorem 7.2.4(b), we may assume that there is no (y, x)-path W

such that $D - W$ has a cycle factor. Thus it follows from the assumption of the corollary that there is an (x, y)-path $P' = v_1 v_2 \dots v_r$, $v_1 = x, v_r = y$ such that $D - P'$ has a cycle factor $\mathcal{F}'$. Since $P' - x$ is contained in S_1, we can argue as above that each S_i, $i > 1$, has a cycle factor (inherited from $\mathcal{F}'$) and hence each S_i contains a hamiltonian cycle C_i, by Theorem 6.6.7.

Note that every vertex of S_1 which is not on P' belongs to some cycle of $\mathcal{F}'$ that lies entirely inside S_1. Hence, if $r = 2$ (that is, P' is just the arc $x \to y$), then it follows from Proposition 7.1.1 (which is also valid when the path in question has length zero) that S_1 contains a hamiltonian path starting at y. This path can easily be extended to a (y, x)-hamiltonian path in D, since each S_i, $i > 1$, is hamiltonian. Thus we may assume that $r \geq 3$.

If $S_1 - y$ is strong, then $D - y$ is strong, contradicting our assumption above. Let $T_1, T_2, \dots, T_s$, $s \geq 2$, be an acyclic ordering of the strong components of $S_1 - y$. Note that each $V(T_i)$ is either covered by some cycles from the cycle factor $\mathcal{F}'$ of $D - P'$ and hence T_i has a hamiltonian cycle (by Theorem 6.6.5), or is covered by a subpath of $P'[v_2, v_{r-1}]$ and some cycles (possibly zero) from $\mathcal{F}'$ and hence T_i has a hamiltonian path (by Theorem 6.6.1). Note also that there is at least one arc from y to T_1 and at least one arc from T_s to y. If T_1 contains a portion of $P'[v_2, v_{r-1}]$, then it is clear that T_1 contains v_2. But then $D - y$ is strong since $x \to v_2$, contradicting our assumption. So T_1 contains no vertices of $P'[v_1, v_{r-1}]$ and hence, by the remark above, T_1 has a hamiltonian cycle to which there is at least one arc from y. Using the structure derived above, it is easy to show that D has a (y, x)-hamiltonian path (Exercise 7.6). □

It can be seen from the results above that when we consider weak hamiltonian-connectedness, extended tournaments have a structure which is closely related to that of tournaments. To see that Theorem 7.2.4 does not extend to general multipartite tournaments, consider the multipartite tournament D obtained from a hamiltonian bipartite tournament B with classes X and Y, by adding two new vertices x and y along with the following arcs: all arcs from x to X and from Y to x, all arcs from y to Y and X to y and an arc between x and y in any direction. It is easy to see that D satisfies none of the conditions (a)-(c) in Theorem 7.2.4, yet there can be no hamiltonian path with end-vertices x and y in D because any such path would contain a hamiltonian path of B starting and ending in X or starting and ending in Y. Such a path cannot exist for parity reasons ($|X| = |Y|$). Note also that we can choose B so that the resulting multipartite tournament is highly connected.

Bang-Jensen and Manoussakis [110] characterized weakly hamiltonian-connected bipartite tournaments. In particular, they proved a necessary and sufficient condition for the existence of an $[x, y]$-hamiltonian path in a bipartite tournament. The statement of this characterization turns out to be quite similar to that of Theorem 7.2.4. The only difference between the statements of these two characterizations is in Condition (c): in the characterization for

bipartite tournaments the set of forbidden digraphs is absolutely different and moreover infinite.

7.2.2 Results for Locally Semicomplete Digraphs

Our next goal is to describe the solution of the $[x, y]$-hamiltonian path problem for locally semicomplete digraphs. Notice that this solution also covers the case of semicomplete digraphs and so, in particular, it generalizes Theorem 7.2.1 to semicomplete digraphs.

We start by establishing notation for some special locally semicomplete digraphs. Up to isomorphism there is a unique strong tournament with four vertices. We denote this by T_4^1. It has the following vertices and arcs:

$$V(T_4^1) = \{a_1, a_2, a_3, a_4\}, A(T_4^1) = \{a_1a_2, a_2a_3, a_3a_4, a_4a_1, a_1a_3, a_2a_4\}.$$

The semicomplete digraphs T_4^2, T_4^3, and T_4^4 are obtained from T_4^1 by adding some arcs, namely:

$$A(T_4^2) = A(T_4^1) \cup \{a_3a_1, a_4a_2\},$$

$$A(T_4^3) = A(T_4^1) \cup \{a_3a_1\}, A(T_4^4) = A(T_4^1) \cup \{a_1a_4\}.$$

Let $\mathcal{T}_4 = \{T_4^1, T_4^2, T_4^3, T_4^4\}$. It is easy to check that every digraph of $\mathcal{T}_4$ has a unique hamiltonian cycle and has no hamiltonian path between two vertices which are not consecutive on this hamiltonian cycle (two such vertices are called **opposite**).

Let $\mathcal{T}_6$ be the set of semicomplete digraphs with the vertex set $\{x_1, x_2, a_1, a_2, a_3, a_4\}$ such that each member D of $\mathcal{T}_6$ has a cycle $a_1a_2a_3a_4a_1$ and the digraph $D\langle\{a_1, a_2, a_3, a_4\}\rangle$ is isomorphic to one member of $\mathcal{T}_4$, in addition, $x_i \to \{a_1, a_3\} \to x_{3-i} \to \{a_2, a_4\} \to x_i$ for $i = 1$ or $i = 2$. It is straightforward to verify that $\mathcal{T}_6$ contains only two tournaments (denoted by T_6' and T_6''), namely, the ones shown in Figure 7.1, and that $|\mathcal{T}_6| = 11$. Since none of the digraphs of $\mathcal{T}_4$ has a hamiltonian path connecting any two opposite vertices, no digraph of $\mathcal{T}_6$ has a hamiltonian path between x_1 and x_2.

For every even integer $m \geq 4$ there is only one 2-strong, 2-regular locally semicomplete digraph on m vertices, namely, the second power $\vec{C}_m^2$ of an m-cycle (Exercise 7.8). We define

$$\mathcal{T}^* = \{\, \vec{C}_m^2 \mid m \text{ is even and } m \geq 4\}.$$

It is not difficult to prove that every digraph of $\mathcal{T}^*$ has a unique hamiltonian cycle and is not weakly hamiltonian-connected (Exercise 7.9, see also

[69]). For instance, if the unique hamiltonian cycle of $\vec{C}_6^2$ is denoted by $u_1u_2u_3u_4u_5u_6u_1$, then $u_1u_3u_5u_1$ and $u_2u_4u_6u_2$ are two cycles of $\vec{C}_6^2$ and there is no hamiltonian path between any two vertices of $\{u_1, u_3, u_5\}$ or of $\{u_2, u_4, u_6\}$.

Let T_8^1 be the digraph consisting of $\vec{C}_6^2$ together with two new vertices x_1 and x_2 such that $x_1 \to \{u_1, u_3, u_5\} \to x_2 \to \{u_2, u_4, u_6\} \to x_1$. Furthermore, T_8^2 (T_8^3, respectively) is defined as the digraph obtained from T_8^1 by adding the arc x_1x_2 (the arcs x_1x_2 and x_2x_1, respectively). Let $\mathcal{T}_8 = \{T_8^1, T_8^2, T_8^3\}$. It is easy to see that every element of $\mathcal{T}_8$ is a 3-strong locally semicomplete digraph and has no hamiltonian path between x_1 and x_2.

Before we present the main result, we state the following two lemmas that were used in the proof of Theorem 7.2.10 by Bang-Jensen, Guo and Volkmann in [81]. The first lemma generalizes the structure found in the last part of the proof of Theorem 7.2.1.

Lemma 7.2.8 *[81] Let D be a strong locally semicomplete digraph on $n \geq 4$ vertices and x_1, x_2 two distinct vertices of D. If $D - \{x_1, x_2\}$ is strong, and $N^+(x_1) \cap N^+(x_2) \neq \emptyset$ or $N^-(x_1) \cap N^-(x_2) \neq \emptyset$, then D has a hamiltonian path connecting x_1 and x_2.*

Proof: Exercise 7.10. □

Another useful ingredient in the proof of Theorem 7.2.10 is the following linking result. An **odd chain** is the second power, $\vec{P}_{2k+1}^2$ for some $k \geq 1$, of a path on an odd number of vertices.

Lemma 7.2.9 *[81] Let D be a connected, locally semicomplete digraph with $p \geq 4$ strong components and acyclic ordering $D_1, D_2, ..., D_p$ of these. Suppose that $V(D_1) = \{u_1\}$ and $V(D_p) = \{v_1\}$ and that $D - x$ is connected for every vertex x. Then, for every choice of $u_2 \in V(D_2)$ and $v_2 \in V(D_{p-1})$, D has two vertex disjoint paths P_1 from u_2 to v_1 and P_2 from u_1 to v_2 with $V(P_1) \cup V(P_2) = V(D)$ if and only if D is not an odd chain from u_1 to v_1.*

Proof: If D is an odd chain, it is easy to see that D has no two vertex-disjoint (u_i, v_{3-i})-paths, for $i = 1, 2$ (Exercise 7.11). We prove by induction on p that the converse is true as well. Suppose that D is not an odd chain from u_1 to v_1. Since the subdigraph $D - x$ is connected for every vertex x, $|N^+(D_i)| \geq 2$ for all $i \leq p - 2$ and $|N^-(D_j)| \geq 2$ for all $j \geq 3$. If $p = 4$, then it is not difficult to see that D has two vertex-disjoint paths P_1 from u_2 to v_1 and P_2 from u_1 to v_2 with $V(P_1) \cup V(P_2) = V(D)$ (Exercise 7.13). If $p = 5$, it is also not difficult to check that D has the desired paths, unless D is a chain on five vertices. So we assume that $p \geq 6$. Now we consider the digraph D', which is obtained from D by deleting the vertex sets $\{u_1, v_1\}$, $V(D_2 - u_2)$ and $V(D_{p-1} - v_2)$.

Using the assumption on D, it is not difficult to show that D' is a connected, but not strongly connected locally semicomplete digraph with the

acyclic ordering $\{u_2\}, D_3, D_4, \ldots, D_{p-2}, \{v_2\}$ of its strong components. Furthermore, for every vertex y of D', the subdigraph $D' - y$ is still connected. Let u be an arbitrary vertex of D_3 and v an arbitrary vertex of D_{p-2}. Note that there is a (u_1, u)-hamiltonian path P in $D\langle\{u_1, u\} \cup V(D_2 - u_2)\rangle$ and similarly there is a (v, v_1)-hamiltonian path Q in $D\langle\{v, v_1\} \cup V(D_{p-1} - v_2)\rangle$. Hence if D' has disjoint (u_2, v)-, (u, v_2)-paths which cover all vertices of D', then D has the desired paths. So we can assume D' has no such paths. By induction, D' is an odd chain from u_2 to v_2. Now using that D is not an odd chain from u_1 to v_1 it is easy to see that D has the desired paths. We leave the details to the reader. □

A weaker version of Lemma 7.2.9 was proved in [69, Theorem 4.5].

Below we give a characterization, due to Bang-Jensen, Guo and Volkmann, for the existence of an $[x, y]$-hamiltonian path in a locally semicomplete digraph. Note again the similarity to Theorem 7.2.1.

Theorem 7.2.10 *[81] Let D be a connected locally semicomplete digraph on n vertices and x_1 and x_2 be two distinct vertices of D. Then D has no hamiltonian $[x_1, x_2]$-path if and only if one of the following conditions is satisfied:*

(1) D is not strong and either the initial or the terminal component of D (or both) contains none of x_1, x_2.
(2) D is strongly connected, but not 2-strong,
  (2.1) there is an $i \in \{1, 2\}$ such that $D - x_i$ is not strong and x_{3-i} belongs to neither the initial nor the terminal component of $D - x_i$;
  (2.2) $D - x_1$ and $D - x_2$ are strong, s is a separating vertex of D, $D_1, D_2, ..., D_p$ is the acyclic ordering of the strong components of $D - s$, $x_i \in V(D_\alpha)$ and $x_{3-i} \in V(D_\beta)$ with $\alpha \leq \beta - 2$. Furthermore, $V(D_{\alpha+1}) \cup V(D_{\alpha+2}) \cup ... \cup V(D_{\beta-1})$ contains a separating vertex of D, or $D' = D\langle V(D_\alpha) \cup V(D_{\alpha+1}) \cup ... \cup V(D_\beta)\rangle$ is an odd chain from x_i to x_{3-i} with $N^-(D_{\alpha+2}) \cap V(D - V(D')) = \emptyset$ and $N^+(D_{\beta-2}) \cap V(D - V(D')) = \emptyset$.
(3) D is 2-strong and is isomorphic to T_4^2 or to one member of $\mathcal{T}_6 \cup \mathcal{T}_8 \cup \mathcal{T}^*$ and x_1, x_2 are the corresponding vertices in the definitions. □

As an easy consequence of Theorem 7.2.10, we obtain a characterization of weakly hamiltonian-connected locally semicomplete digraphs. The proof is left to the interested reader as Exercise 7.12.

Theorem 7.2.11 *[81] A locally semicomplete digraph D with at least three vertices is weakly hamiltonian–connected if and only if it satisfies (a), (b) and (c) below:*

(a) D is strong,
(b) For every $x \in V(D)$, $D - x$ has at most two strong components,
(c) D is not isomorphic to any member of $\mathcal{T}_6 \cup \mathcal{T}_8 \cup \mathcal{T}^*$. □

7.3 Hamiltonian-Connected Digraphs

We now turn to hamiltonian paths with specified initial and terminal vertices. An **(x, y)-hamiltonian path** is a hamiltonian path from x to y. Clearly, asking for such a path in an arbitrary digraph is an even stronger requirement than asking for an $[x, y]$-hamiltonian path[3]. A digraph $D = (V, A)$ is **hamiltonian-connected** if D has an (x, y)-hamiltonian path for every choice of distinct vertices $x, y \in V$.

We can check, in polynomial time, whether a digraph of bounded directed tree-width[4] is hamiltonian-connected due the the following theorem by Johnson, Robertson, Seymour and Thomas.

Theorem 7.3.1 *[573] For a digraph D of bounded directed tree-width and a pair x, y of distinct vertices of D, we can check, in polynomial time, whether D has an (x, y)-hamiltonian path.* □

By Lemma 2.13.9, all of the above holds for DAG-width and directed path-width.

No characterization[5] for the existence of an (x, y)-hamiltonian path is known, even for the case of tournaments. However, by Theorem 7.3.6, there is a polynomial algorithm for the problem in the next section, so in the algorithmic sense a good characterization does exist. The following very important partial result, due to Thomassen, was used in the algorithm of Theorem 7.3.6.

Theorem 7.3.2 (Thomassen) *[856] Let $D = (V, A)$ be a 2-strong semicomplete digraph with distinct vertices x, y. Then D contains an (x, y)-hamiltonian path if either (a) or (b) below is satisfied.*

(a) D contains three internally disjoint (x, y)-paths each of length at least two,

(b) D contains a vertex z which is dominated by every vertex of $V - x$ and D contains two internally disjoint (x, y)-paths each of length at least two.

□

In his proof Thomassen explicitly uses the fact that the digraph is allowed to have cycles of length 2. This simplifies the proof (which is still far from trivial), since one can use contraction to reduce to a smaller instance and then use induction.

[3] We know of no class of digraphs for which the $[x, y]$-hamiltonian path problem is polynomially solvable, but the (x, y)-hamiltonian path problem is $\mathcal{NP}$-complete. For arbitrary digraphs they are equivalent from a complexity point of view (see Exercise 7.2).

[4] See Section 2.13 for definitions of directed width parameters.

[5] By this we mean a structural characterization involving only conditions that can be checked in polynomial time.

An important ingredient in the proof of Theorem 7.3.2, as well as in several other proofs concerning the existence of an (x, y)-hamiltonian path in a semicomplete digraph D, is to prove that D contains a spanning acyclic graph in which x can reach all other vertices and y can be reached by all other vertices. The reason for this can be seen from the following result which generalizes an observation by Thomassen in [856].

Proposition 7.3.3 *[72] Let D be a path-mergeable digraph. Then D has a hamiltonian (x, y)-path if and only if D contains a spanning acyclic digraph H in which $d_H^-(x) = d_H^+(y) = 0$ and such that, for every vertex $z \in V(D)$, H contains an (x, z)-path and a (z, y)-path.*

Proof: Exercise 7.15. □

Theorem 7.3.2 and Menger's theorem (see Theorem 5.4.1) immediately imply the following result. For another nice consequence see Exercise 7.16.

Theorem 7.3.4 *[856] If a semicomplete digraph D is 4-strong, then D is hamiltonian-connected.* □

Thomassen constructed an infinite family of 3-strongly connected tournaments with two vertices x, y for which there is no (x, y)-hamiltonian path [856]. Hence, from a connectivity point of view, Theorem 7.3.4 is the best possible.

Theorem 7.3.4 is a very important result with several consequences. Thomassen has shown in several papers how to use Theorem 7.3.4 to obtain results on spanning collections of paths and cycles in semicomplete digraphs. See, e.g., the papers [857, 859] by Thomassen and also Section 7.4.

The next theorem by Bang-Jensen, Manoussakis and Thomassen generalizes Theorem 7.3.2. Recall that for specified distinct vertices s, t, an (s, t)-separator is a subset $S \subseteq V - \{s, t\}$ such that $D - S$ has no (s, t)-path. An (s, t)-separator is **trivial** if either s has out-degree zero or t has in-degree zero in $D - S$.

Theorem 7.3.5 *[111] Let T be a 2-strong semicomplete digraph on at least 10 vertices and let x, y be vertices of T such that $y \mapsto x$. Suppose that both of $T - x$ and $T - y$ are 2-strong. If all (x, y)-separators consisting of two vertices (if any exist) are trivial, then T has an (x, y)-hamiltonian path.* □

Based on Theorem 7.3.5 and several other structural results on 2-strong semicomplete digraphs Bang-Jensen, Manoussakis and Thomassen proved the following.

Theorem 7.3.6 *[111] The (x, y)-hamiltonian path problem is polynomially solvable for semicomplete digraphs.* □

The algorithm uses a divide-and-conquer approach and cannot be easily modified to find a longest (x, y)-path in a semicomplete digraph. There also does not seem to be any simple reduction of this problem to the problem of deciding the existence of a hamiltonian path from x to y.

Conjecture 7.3.7 *[89] There exists a polynomial algorithm which, given a semicomplete digraph D and two distinct vertices x and y of D, finds a longest (x,y)-path.*

Note that if we ask for the longest $[x,y]$-path in a tournament, then this can be answered using Theorem 7.2.1 (see Exercise 7.18).

Conjecture 7.3.8 *[89] There exists a polynomial algorithm which, given a digraph D that is either extended semicomplete or locally semicomplete, and two distinct vertices x and y of D, decides whether D has an (x,y)-hamiltonian path and finds such a path if one exists.*

The following extension of Theorem 7.3.4 to extended tournaments has been conjectured by Bang-Jensen, Gutin and Huang:

Conjecture 7.3.9 *[92] If D is a 4-strong extended tournament with an (x,y)-path P such that $D-P$ has a cycle factor, then D has an (x,y)-hamiltonian path.*

Guo [433] extended Theorem 7.3.4 to locally semicomplete digraphs.

Theorem 7.3.10 (Guo) *[433] Let D be a 2-strong locally semicomplete digraph and let x,y be two distinct vertices of D. Then D contains a hamiltonian path from x to y if (a) or (b) below is satisfied.*

(a) There are three internally disjoint (x,y)-paths in D, each of which is of length at least 2 and D is not isomorphic to any of the digraphs T_8^1 and T_8^2 (see the definition in the preceding section).
(b) The digraph D has two internally disjoint (x,y)-paths P_1,P_2, each of which is of length at least 2 and a path P which either starts at x or ends at y and has only x or y in common with P_1,P_2 such that $V(D)=V(P_1)\cup V(P_2)\cup V(P)$. Furthermore, for any vertex $z\notin V(P_1)\cup V(P_2)$, z has a neighbour on $P_1-\{x,y\}$ if and only if it has a neighbour on $P_2-\{x,y\}$. □

Since neither of the two exceptions in (a) is 4-strong, Theorem 7.3.10 implies the following:

Corollary 7.3.11 *[433] Every 4-strong locally semicomplete digraph is hamiltonian-connected.* □

In [432] Guo used Theorem 7.3.10 to give a complete characterization of those 3-strongly connected arc-3-cyclic (that is, every arc is in a 3-cycle) locally tournament digraphs with no hamiltonian path from x to y for specified vertices x and y. In particular this characterization shows that there exist infinitely many 3-strongly connected digraphs which are locally tournament digraphs (but not semicomplete digraphs) and are not hamiltonian-connected.

Thus, as far as this problem is concerned, it is not only the subclass of semicomplete digraphs which contain difficult instances within the class of locally semicomplete digraphs. It should be noted that Guo's proof does not rely on Theorem 7.3.4. However, due to the non-semicomplete exceptions mentioned above, it seems unlikely that a much simpler proof of Corollary 7.3.11 can be found using Theorems 7.3.4 and 2.10.15.

Not surprisingly, there are also several results on hamiltonian-connectivity in digraphs with many arcs. One example is the following result by Lewin.

Theorem 7.3.12 *[641] If a digraph on $n \geq 3$ vertices has $(n-1)^2 + 1$ or more arcs, then it is hamiltonian-connected.* □

If a digraph D is hamiltonian-connected, then D is also hamiltonian (since every arc is in a hamiltonian cycle). The next result, due to Bermond, shows that we only need a slight strengthening of the degree condition in Corollary 6.4.3 to get a sufficient condition for strong hamiltonian-connectivity.

Theorem 7.3.13 *[144] Every digraph D on n vertices with minimum semi-degree at least $\frac{n+1}{2}$ is hamiltonian-connected.* □

Overbeck-Larisch showed that if we just ask for weak hamiltonian-connectedness, then we can replace the condition on the semi-degrees by a condition on the degrees:

Theorem 7.3.14 *[735] Every 2-strong digraph on n vertices and minimum degree at least $n+1$ is weakly hamiltonian-connected.* □

Thomassen asked whether all 3-strong digraphs $D = (V, A)$ on n vertices with $d^+(x) + d^-(x) \geq n + 1$ for all $x \in V$ are necessarily hamiltonian-connected. However, this is not the case, as was shown by Darbinyan [247].

For further results on hamiltonian cycles containing a subset of the vertices in a prescribed order see Section 10.3.

7.4 Hamiltonian Cycles Containing or Avoiding Prescribed Arcs

We now turn our attention to hamiltonian cycles in digraphs with the extra condition that these cycles must either contain or avoid all arcs from a prescribed subset A' of the arcs. Not surprisingly, problems of this type are quite difficult even for semicomplete digraphs. If we have no restriction on the size of A', then we may easily formulate the hamiltonian cycle problem for arbitrary digraphs as an avoiding problem for semicomplete digraphs. Hence the avoiding problem without any restrictions is certainly $\mathcal{NP}$-complete. Below, we study both types of problems from a connectivity as well as from a complexity point of view. We also show that when the number of arcs to

be avoided, respectively, contained in a hamiltonian cycle, is some constant, then, from a complexity point of view, the avoiding version is no harder than the containing version. Finally, we show that for digraphs which can be obtained from a semicomplete digraph by adding a few new vertices and some arcs, the hamiltonian cycle problem is very hard and even if we just added one new vertex, the problem is highly non-trivial.

7.4.1 Hamiltonian Cycles Containing Prescribed Arcs

We start by studying the problem of finding a hamiltonian cycle that contains certain prescribed arcs $e_1, e_2, \ldots, e_k$. This problem, which we call the k-HCA PROBLEM, is clearly very hard for general digraphs. We show below that even for semicomplete digraphs this is a difficult problem. For $k = 1$ the k-HCA problem is a special case of the (x, y)-hamiltonian path problem and it follows from Theorem 7.3.6 that there is a polynomial algorithm to decide the existence of a hamiltonian cycle containing one prescribed arc in a semicomplete digraph.

Based on the evidence from Theorem 7.3.6, Bang-Jensen, Manoussakis and Thomassen raised the following conjecture. As mentioned above, when $k = 1$ the conjecture follows from Theorem 7.3.6.

Conjecture 7.4.1 *[111] For each fixed k, the k-HCA problem is polynomially solvable for semicomplete digraphs.*

When $k = 2$ the problem already seems very difficult. This is interesting, especially in view of the discussion below concerning hamiltonian cycles in digraphs obtained from semicomplete digraphs by adding a few new vertices. Bang-Jensen and Thomassen proved that when k is not fixed the k-HCA problem becomes $\mathcal{NP}$-complete even for tournaments [118]. The proof of this result in [118] contains an interesting idea which was generalized by Bang-Jensen and Gutin in [84]. Consider a digraph D containing a set W of k vertices such that $D - W$ is semicomplete. Construct a new semicomplete digraph D_W as follows. First, split every vertex $w \in W$ into two vertices w_1, w_2 such that all arcs entering w now enter w_1 and all arcs leaving w now leave w_2. Add all possible arcs from vertices of index 1 to vertices of index 2 (whenever the arcs in the opposite direction are not already present). Add all edges between vertices of the same index and orient them randomly. Finally, add all arcs of the kind $w_1 z$ and $z w_2$, where $w \in W$ and $z \in V(D) - W$. See Figure 7.2. It is easy to show that the following holds:

Proposition 7.4.2 *[84] Let W be a set of k vertices of a digraph D such that $D - W$ is a semicomplete digraph. Then D has a cycle of length $c \geq k$ containing all vertices of W, if and only if the semicomplete digraph D_W has a cycle of length $c + k$ through the arcs $\{w_1 w_2 : w \in W\}$.*

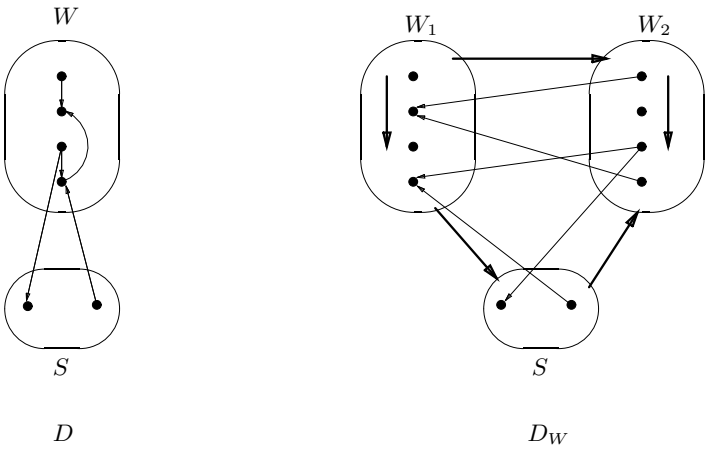


Figure 7.2 The construction of D_W from D and W. The bold arc from W_1 to W_2 indicates that all arcs not already going from W_2 to W_1 (as copies of arcs in D) go in the direction shown. The four other bold arcs indicate that all possible arcs are present in the direction shown.

Proof: Exercise 7.20. □

Let $D = (V, A)$ be a semicomplete digraph and $A' = \{u_1v_1, \ldots, u_kv_k\}$ be a subset of A. Let D' be the digraph obtained from D by replacing each arc $u_iv_i \in A'$ by a path $u_iw_iv_i$, $i \in [k]$, where w_i is a new vertex. Then every cycle C in D that uses all arcs in A' corresponds to a cycle C' in D' which contains all vertices of $W = \{w_1, w_2, \ldots, w_k\}$ and conversely. This observation and Proposition 7.4.2 allows us to study cycles through a specified set W of vertices in digraphs D such that $D - W$ is semicomplete instead of studying cycles containing $k = |W|$ fixed arcs in semicomplete digraphs.

Note that if k is not fixed, then it is $\mathcal{NP}$-complete to decide the existence of a cycle through k given vertices in a digraph which can be obtained from a semicomplete digraph by adding k new vertices and some arcs. Indeed, take $k = |V(D)|$, then this is the Hamilton cycle problem for general digraphs. This proves that the k-HCA is $\mathcal{NP}$-complete for semicomplete digraphs.

Now we can reformulate Conjecture 7.4.1 to the following equivalent statement:

Conjecture 7.4.3 *[84] Let k be a fixed natural number. There exists a polynomial algorithm to decide if there is a hamiltonian cycle in a given digraph D which is obtained from a semicomplete digraph by adding at most k new vertices and some arcs.*

The truth of this conjecture when $k = 1$ follows from Proposition 7.4.2 and Theorem 7.3.6. Surprisingly, when $|W| = 2$ the problem already seems very difficult.

We conclude this subsection with some results on the k-HCA problem for highly connected tournaments. Thomassen [859] obtained the following theorem for tournaments with large strong connectivity (the function $f(k)$ is defined recursively by $f(1) = 1$ and $f(k) = 2(k-1)f(k-1) + 3$ for $k \geq 2$). The proof is by induction on k and uses Theorem 7.3.4 to establish the case $k = 1$ (this is another illustration of the importance of Theorem 7.3.4).

Theorem 7.4.4 *[859] If $\{x_1, y_1, ..., x_k, y_k\}$ is a set of distinct vertices in an $h(k)$-strong tournament T, where $h(k) = f(5k)+12k+9$, then T has a k-path factor $P_1 \cup P_2 \cup ... \cup P_k$ such that P_i is an (x_i, y_i)-path for $i = 1, ..., k$.* □

Theorem 7.4.4 implies the following:

Theorem 7.4.5 *[859] If $a_1, ..., a_k$ are arcs with no common head or tail in an $h(k)$-strong tournament T, then T has a hamiltonian cycle containing $a_1, ..., a_k$ in that cyclic order.* □

Combining the ideas of avoiding and containing, Thomassen proved the following (where the function h was defined in Theorem 7.4.4):

Theorem 7.4.6 *[859] For any set A_1 of at most k arcs in an $h(k)$-strong tournament T and for any set A_2 of at most k independent[6] arcs of $T - A_1$, the digraph $T - A_1$ has a hamiltonian cycle containing all arcs of A_2.* □

7.4.2 Avoiding Prescribed Arcs with a Hamiltonian Cycle

How many arcs can we delete from a strong tournament and still have a hamiltonian cycle, no matter what set of arcs is deleted? This is a difficult question, but it is easy to see that for some tournaments the answer is that even one missing arc may destroy all hamiltonian cycles. If some vertex has in- or out-degree 1, then deleting that arc clearly suffices to destroy all hamiltonian cycles. On the other hand, for every p, it is also easy to construct an infinite set $\mathcal{S}$ of strong tournaments in which $\delta^0(T) \geq p$ for every $T \in \mathcal{S}$ and yet there is some arc of T which is on every hamiltonian cycle of T (see Exercise 7.19). It follows from Theorem 7.4.7 below that all such tournaments are strong but not 2-strong.

We can generalize the question to k-strong tournaments and again it is obvious that if some vertex v has in- or out-degree k (this is the smallest possible by the connectivity assumption), then deleting all k arcs out of or into v, we can obtain a digraph with no hamiltonian cycle. Thomassen [857] conjectured that in a k-strong tournament, k is the minimum number of arcs one can delete in order to destroy all hamiltonian cycles. The next theorem due to Fraisse and Thomassen answers this in the affirmative.

[6] A set of arcs is independent if no two of the arcs share a vertex.

Theorem 7.4.7 *[334] For every k-strong tournament $D = (V, A)$ and every set $A' \subset A$ such that $|A| \leq k-1$, there is a hamiltonian cycle C in $D-A'$.* □

The proof is long and non-trivial; in particular it uses Theorem 7.3.4. Below we describe a stronger result due to Bang-Jensen, Gutin and Yeo [96]. The authors proved Theorem 7.4.8 using results on irreducible cycle factors in multipartite tournaments, in particular Yeo's irreducible cycle factor theorem (Theorem 6.6.20). This is just one more illustration of the power of Theorem 6.6.20.

Theorem 7.4.8 *[96] Let $T = (V, A)$ be a k-strong tournament on n vertices, and let $X_1, X_2, \ldots, X_p$ $(p \geq 1)$ be a partition of V such that $1 \leq |X_1| \leq |X_2| \leq \ldots \leq |X_p|$. Let D be the digraph obtained from T by deleting all arcs which have both head and tail in the same X_i (i.e., $D = T - \cup_{i=1}^{p} A(T\langle X_i \rangle)$). If $|X_p| \leq n/2$ and $k \geq |X_p| + \sum_{i=1}^{p-1} \lfloor |X_i|/2 \rfloor$, then D is hamiltonian. In other words, T has a hamiltonian cycle which avoids all arcs with both head and tail in some X_i.* □

We will not give the proof here since it is quite technical, but we give the main idea of the proof. The first observation is that D is a multipartite tournament, which follows from the way we constructed it. Our goal is to apply Theorem 6.6.20 to D. Hence we need to establish that D is strong (see Exercise 7.24) and has a cycle factor (Exercise 7.25). Now we can apply Theorem 6.6.20 to prove that every irreducible cycle factor in D is a hamiltonian cycle. This last step is non-trivial (Exercise 7.26).

The following result shows that the bound for k in Theorem 7.4.8 is sharp:

Theorem 7.4.9 *[96] Let $2 \leq r_1 \leq r_2 \leq \ldots \leq r_p$ be arbitrary integers. Then there exists a tournament T and a collection $X_1, X_2, ..., X_p$ of disjoint sets of vertices in T such that*

(a) T is $(r_p - 1 + \sum_{i=1}^{p-1} \lfloor r_i/2 \rfloor)$-strong;
(b) $|X_i| = r_i$ for $i \in [p]$;
(c) $D = T - \cup_{i=1}^{p} A(T\langle X_i \rangle)$ is not hamiltonian. □

In fact, the paper [96] is concerned with aspects of the following more general problem:

Problem 7.4.10 *[96] Which sets B of edges of the complete graph K_n have the property that every k-strong orientation of K_n induces a hamiltonian digraph on $K_n - B$?*

The Fraisse-Thomassen theorem says that this is the case whenever B contains at most $k-1$ edges. Theorem 7.4.8 says that a union of disjoint cliques of sizes $r_1, \ldots, r_p$ has the property whenever $\sum_{i=1}^{l} \lfloor r_i/2 \rfloor + \max_{1 \leq i \leq l} \{\lceil r_i/2 \rceil\} \leq k$. By Theorem 7.4.9, this is the best possible result for unions of cliques.

Let us show that Theorem 7.4.8 implies Theorem 7.4.7. Let T be a k-strong tournament on n vertices and let $A' = \{e_1, e_2, \ldots, e_{k-1}\}$ be a given

set of $k-1$ arcs of T. In $UG(T)$ these arcs induce a number of connected components $X_1, X_2, \ldots, X_p$, $1 \leq p \leq k-1$. Denote by a_i, $i \in [p]$, the number of arcs from A' which join two vertices from X_i. Then we have $\sum_{i=1}^{p} a_i = k-1$ and $|X_i| \leq a_i + 1$, $i \in [p]$. We may assume that the numbering is chosen so that $|X_1| \leq |X_2| \leq \ldots \leq |X_p|$. Note that $|X_p| \leq k < n/2$. Furthermore, since each $a_i \geq 1$, we also have $|X_p| \leq (k-1) - (p-1) + 1 = k - p + 1$. Now we can make the following calculation:

$$\begin{aligned}
|X_p| + \sum_{i=1}^{p-1} \lfloor \frac{|X_i|}{2} \rfloor &= \lceil \frac{|X_p|}{2} \rceil + \sum_{i=1}^{p} \lfloor \frac{|X_i|}{2} \rfloor \\
&\leq \lceil \frac{|X_p|}{2} \rceil + \lfloor \frac{1}{2} \sum_{i=1}^{p} |X_i| \rfloor \\
&\leq \lceil \frac{k-p+1}{2} \rceil + \lfloor \frac{1}{2} \sum_{i=1}^{p} (a_i + 1) \rfloor \\
&= \lceil \frac{k-p+1}{2} \rceil + \lfloor \frac{k-1+p}{2} \rfloor \\
&= k.
\end{aligned}$$

Now it follows from Theorem 7.4.8 that T has a hamiltonian cycle which avoids every arc with both head and tail in some X_i and in particular it avoids all arcs in A'. This shows that Theorem 7.4.8 implies Theorem 7.4.7.

Note that if A' induces a tree and possibly some disjoint edges in $UG(T)$, then Theorem 7.4.8 is no stronger than Theorem 7.4.7. This can be seen from the fact that in this case we have equality everywhere in the calculation above. In all other cases Theorem 7.4.8 provides a stronger bound.

In relation to Problem 7.4.10, it seems natural to investigate bounds for k in different cases of the set B. In particular, one may consider the following problems.

Problem 7.4.11 *What are sharp bounds for k in Problem 7.4.10 when B is a spanning forest of K_n consisting of m disjoint paths containing r_1, ..., r_m vertices, respectively?*

Problem 7.4.12 *What are sharp bounds for k in Problem 7.4.10 when B is a spanning forest of K_n consisting of m disjoint stars containing r_1, ..., r_m vertices, respectively?*

Problem 7.4.13 *What are sharp bounds for k in Problem 7.4.10 when B is a spanning cycle subdigraph of K_n consisting of m disjoint cycles containing r_1, ..., r_m vertices, respectively?*

How easy is it to decide, given a semicomplete digraph $D = (V, A)$ and a subset $A' \subseteq A$, whether D has a hamiltonian cycle C which avoids all arcs of A'? As we mentioned earlier, this problem is $\mathcal{NP}$-complete if we pose no restriction on the arcs in A'. In the case when A' is precisely the set of those arcs that lie inside the sets of some partition $X_1, X_2, \ldots, X_r$ of V, then the existence of C can be decided in polynomial time. This follows from the fact that $D\langle A - A'\rangle$ is a semicomplete multipartite digraph and, by Theorem 6.6.9, the hamiltonian cycle problem is polynomially solvable for semicomplete multipartite digraphs. The same argument also covers the case when $k = 1$ in the conjecture below.

Conjecture 7.4.14 *For every k there exists a polynomial algorithm which, for a given semicomplete digraph $D = (V, A)$ and a subset $A' \subseteq A$ such that $|A'| = k$, decides whether D has a hamiltonian cycle that avoids all arcs in A'.*

At first glance, cycles that avoid certain arcs seem to have very little to do with cycles that contain certain specified arcs. Hence, somewhat surprisingly, if Conjecture 7.4.1 is true, then so is[7] Conjecture 7.4.14: Suppose that Conjecture 7.4.1 is true. Then it follows from the discussion of Subsection 7.4.1 that also Conjecture 7.4.3 holds. Hence, for fixed k, there is a polynomial algorithm $\mathcal{A}_k$ which, given a digraph $D = (V, A)$ and a subset $W \subset V$ for which $D - W$ is semicomplete and $|W| \leq k$, decides whether or not D has a hamiltonian cycle. Let k be fixed and D be a semicomplete digraph and let A', $|A'| \leq k$, be a prescribed set of arcs in D. Let W be the set of all vertices such that at least one arc of A' has head or tail in W. Then $|W| \leq 2|A'|$ and D has a hamiltonian cycle avoiding all arcs in A' if and only if the digraph $D - A'$ has a hamiltonian cycle. By the remark above we can test this using the polynomial algorithm $\mathcal{A}_r$, where $r = |W|$.

7.4.3 Hamiltonian Cycles Avoiding Arcs in 2-Cycles

Recall from Chapter 2 that we call an arc xy ordinary if it is not contained in a 2-cycle. Deciding whether a given digraph has a hamiltonian cycle C such that all arcs of C are ordinary is of course an $\mathcal{NP}$-complete problem since the hamiltonian cycle problem for oriented graphs is $\mathcal{NP}$-complete. This implies that the problem is $\mathcal{NP}$-complete even for semicomplete digraphs.

Tuza [881] studied this problem for semicomplete digraphs and posed the following conjecture:

Conjecture 7.4.15 *[881] Let s be a positive integer and suppose that $D = (V, A)$ is a semicomplete digraph such that for every $Y \subset V$, $|Y| < s$, the*

[7] We thank Thomassen for pointing out this consequence to us (private communication, August 1999).

induced semicomplete digraph $D\langle V - Y\rangle$ is strong and has at least one ordinary arc. Then there exists a hamiltonian cycle in D which has at least s ordinary arcs.

The following result shows that it is enough to prove that there is a cycle of length at least $s + 1$ with this property.

Proposition 7.4.16 *[881] If a strong semicomplete digraph T has a cycle of length at least $s + 1$ which contains at least s ordinary arcs, then T has a hamiltonian cycle with at least s ordinary arcs.* □

Tuza has proved the existence of such a cycle for $s = 1, 2$, see [881]. It is easy to see that $s+1$ cannot be replaced by s in Proposition 7.4.16 (Exercise 7.29).

7.5 Arc-Traceable Digraphs

A digraph $D = (V, A)$ is **arc-traceable** if every arc $xy \in A$ is contained in a hamiltonian path of D. In this short section we briefly discuss results on semicomplete digraphs.

First observe that arc-traceable semicomplete digraphs can be recognized in polynomial time. This follows from the fact that there is a polynomial algorithm for checking whether there is a hamiltonian path through a given arc in a semicomplete digraph (Exercise 7.28).

Since every strong in-semicomplete digraph is hamiltonian (Theorem 6.3.1) we easily get the following observation (Exercise 7.27).

Proposition 7.5.1 *[693] Every 2-strong in-semicomplete digraph is arc-traceable.* □

In [185] Busch, Jacobson and Reid studied strong tournaments which are not arc-traceable. Although they were not able to characterize these, they found some necessary conditions for a strong tournament to be non-arc-traceable, one of which is the following.

Theorem 7.5.2 *[185] Let T be a strong tournament containing an arc xy which is not in any hamiltonian path of T. Then the following holds:*

(1) T has a vertex z such that $T - z$ is not strong.
(2) T has k strong components, $T_1, T_2, \ldots, T_k$, $k \geq 4$, where T_1 is the initial and T_k the terminal strong component of $T - z$.
(3) The vertex x is in T_1 and y is in T_k.
(4) The vertex z has no arc to T_2 and no arc from T_{k-1}. □

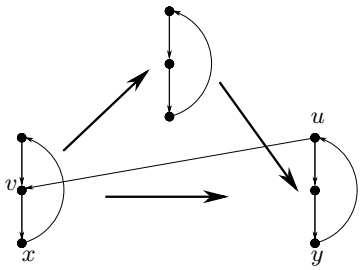


Figure 7.3 A strong tournament T with no hamiltonian path including the arc xy. The bold arcs indicate that all arcs between the three 3-cycles, except the arc uv, have the direction shown.

Figure 7.3 illustrates the theorem. It also shows that the following necessary condition for a tournament to be arc-traceable, is not sufficient: for every arc xy, T contains a 1-path-cycle factor $\mathcal{F}$ in which xy is an arc of the unique path in $\mathcal{F}$.

Theorem 7.5.3 *[185] If T is a strong tournament with $\delta^0(T) \geq 2$ and for every arc $xy \in A(T)$, $d^-(x) + d^+(y) \geq \frac{n}{2} - 2$, then T is arc-traceable.* □

In [693] Meierling and Volkmann studied arc-traceable locally semicomplete digraphs and obtained a number of results which generalize those in [185], including Theorem 7.5.2.

7.6 Oriented Hamiltonian Paths and Cycles

Since every tournament has a hamiltonian directed path, it is natural to ask whether every tournament contains every orientation of a hamiltonian undirected path. This is not true, as one can see from the examples in Figure 7.4.

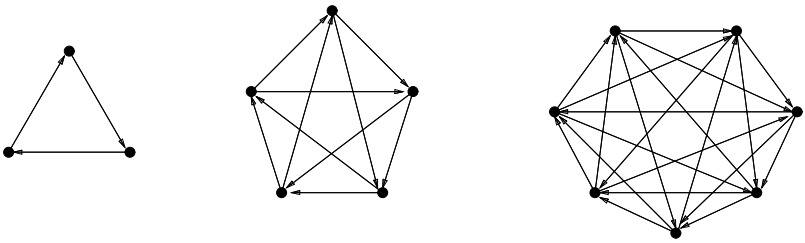


Figure 7.4 The unique tournaments with no anti-directed hamiltonian path.

A path is **anti-directed** if the orientation of each arc on the path is opposite to that of its predecessor. The reader can easily verify that none of the three tournaments in Figure 7.4 contains an anti-directed hamiltonian path. Grünbaum [431] proved that, except for the three tournaments of Figure 7.4, every tournament contains an anti-directed hamiltonian path. Rosenfeld [787] strengthened this to the following statement:

Theorem 7.6.1 *[787] In a tournament on at least nine vertices, every vertex is the origin of an anti-directed hamiltonian path.* □

Rosenfeld conjectured that there exists a natural number N such that every tournament on at least N vertices contains every orientation of a hamiltonian undirected path. Grünbaum's examples show that we must have $N \geq 8$. Rosenfeld's conjecture has been studied extensively and many partial results were obtained until it was proved by Thomason [851] (see also Theorem 7.6.3). We will mention one of these partial results here (see also the papers [37] by Alspach and Rosenfeld and [835] by Straight).

Forcade found the following beautiful result which generalizes Redei's theorem for tournaments whose number of vertices is a power of two.

Theorem 7.6.2 *[329] If T is a tournament on $n = 2^r$ vertices for some r, then for every orientation P of a path on n vertices, T contains an odd number of occurrences of P.* □

Thomason [851] proved Rosenfeld's conjecture by showing that N is less than 2^{128}. He also conjectured that $N = 8$ should be the right number. This was confirmed by Havet and Thomassé [509].

Theorem 7.6.3 (Havet-Thomassé theorem) *[509] Every tournament on at least eight vertices contains every orientation of a hamiltonian path.* □

The proof of Theorem 7.6.3 in [509] is very long (involving a lot of cases), but it uses a very nice partial result which we shall describe below. First we need some new notation. Let $P = u_1u_2 \dots u_n$ be an oriented path. The vertex u_1 (u_n) is the **origin** (**terminus**) of P. An **interval** of P is a maximal subpath $P' = P[u_i, u_j]$[8] such that P' is a directed path (i.e., either a (u_i, u_j)-path or a (u_j, u_i)-path). See an illustration in Figure 7.5. The intervals are labeled $I_1, I_2, \dots, I_{t(P)}$ starting from u_1. The **length** $\ell_i(P)$ of the ith interval is the number of arcs in the directed subpath corresponding to I_i. If the first interval of P is directed out of u_1, then P is an **out-path**, otherwise P is an **in-path**. Now we can describe any oriented path P by a signed sequence $sgn(P)(\ell_1, \ell_2, \dots, \ell_{t(P)})$, where $sgn(P)$ is '+' if P is an out-path and otherwise $sgn(P)$ is '−'. We also use the notation $*P$ to denote the subpath $P[u_2, u_n]$.

[8] We use the same notation here as for directed paths, i.e., $P[u_i, u_j] = u_iu_{i+1} \dots u_j$ when $i \leq j$.

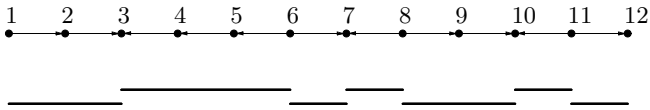


Figure 7.5 An oriented path with intervals $[1,3], [3,6], [6,7], [7,8], [8,10], [10,11], [11,12]$.

For every set $X \subseteq V$ in a tournament $T = (V, A)$, we define the sets $R^+(X)$ ($R^-(X)$) to be those vertices that can be reached from (can reach) the set X by a directed path. By definition $X \subseteq R^+(X) \cap R^-(X)$. A vertex u is an **out-generator** (**in-generator**) of T if $R^+(u) = V$ ($R^-(u) = V$). Recall that by Theorem 1.4.2, every tournament T has at least one out-generator and at least one in-generator. In fact, by Proposition 2.9.2, a vertex is an out-generator (in-generator) if and only if it is the initial (terminal) vertex of at least one hamiltonian path in T.

The next result, due to Havet and Thomassé, deals with oriented paths covering all but one vertex in a tournament. It plays an important role in the proof of Theorem 7.6.3 in [509].

Theorem 7.6.4 *[509] Let $T = (V, A)$ be a tournament on $n + 1$ vertices. Then*

(1) For every out-path P on n vertices and every choice of distinct vertices x, y such that $|R^+(\{x, y\})| \geq \ell_1(P) + 1$, either x or y is an origin of (a copy of) P in T.
(2) For every in-path P on n vertices and every choice of distinct vertices x, y such that $|R^-(\{x, y\})| \geq \ell_1(P) + 1$, either x or y is an origin of (a copy of) P in T.

The following is an easy corollary of Theorem 7.6.4. We state it now since we shall use it in the inductive proof below.

Corollary 7.6.5 *[851] Every tournament T on n vertices contains every oriented path P on $n-1$ vertices. Moreover, every subset of $\ell_1(P)+1$ vertices contains an origin of P. In particular, there are at least two distinct origins of P in T.* □

Proof of Theorem 7.6.4: (We follow the proof in [509]). The proof is by induction on n and clearly holds for $n = 1$. Now suppose that the theorem holds for all tournaments on at most n vertices. It suffices to prove (1) since (2) can be reduced to (1) by considering the converses of T and P.

Let $P = u_1 u_2 \dots u_n$ be given and let x, y be distinct vertices such that $|R^+(\{x, y\})| \geq \ell_1(P) + 1$. We may assume that $x \rightarrow y$ and hence $R^+(x) = R^+(\{x, y\})$. We consider two cases.

Case 1 $\ell_1(P) \geq 2$: If $|N^+(x)| \geq 2$, let $z \in N^+(x)$ be an out-generator of $T\langle R^+(x) - x\rangle$ and let $t \in N^+(x)$ be distinct from z. By the definition of z we have that $|R^+_{T-x}(\{t,z\})| = |R^+(x)| - 1 > \ell_1(*P)$. Note that $*P$ is an out-path, since $\ell_1(P) > 1$. By the induction hypothesis, either z or t is the origin of $*P$ in $T - x$, implying that x is an origin of P in T.

Thus we may assume that $N^+(x) = \{y\}$. Since $|R^+(\{x,y\})| \geq \ell_1(P)+1 \geq 3$ we see that $N^+(y) \neq \emptyset$. Let q be an out-generator of $T\langle N^+(y)\rangle$. Then q is also an out-generator of $T\langle R^+(\{x,y\}) - y\rangle$, $q \rightarrow x$ and $|R^+_{T-y}(\{x,q\})| = |R^+(\{x,y\})| - 1 > \ell_1(*P)$. By induction, either x or q is the origin of $*P$ in $T - y$ and since x has no out-neighbour in $T - y$ it must be q that is the origin. Now we see that y is the origin of P in T.

Case 2 $\ell_1(P) = 1$: We consider first the subcase when $|N^+(x)| \geq 2$. Let $X := R^-_{T-x}(N^+(x))$ and consider the partition $(X, Y, \{x\})$ of V, where $Y = V - X - x$. By the definition of these sets, we have $Y \mapsto x$, $X \mapsto Y$ and $y \in X$. If $|X| \geq \ell_2(P) + 1$, then we claim that x is an origin of P in T; indeed, let $p \in N^+(x)$ be an in-generator of $T\langle X\rangle$ and take $u \in N^+(x) - p$. By the induction hypothesis, either p or u is an origin of $*P$ in $T - x$ and hence x is an origin of P in T.

So we may assume that $|X| \leq \ell_2(P)$. Note that $\ell_2(P) \leq n - 2$ holds always (remember we count arcs). Hence $|Y| > 1$, since T has $n + 1$ vertices. Let s be an in-generator of $T\langle Y\rangle$. Since $d^+(x) > 1$ and $X \mapsto Y$ we have $R^-_{T-y}(s) = V - y$. Let $w \in Y - s$ be arbitrary. By the induction hypothesis either w or s is an origin of $*P$ in $T - y$ and hence y is an origin of P in T.

Now consider the case when $N^+(x) = \{y\}$. Suppose first that $|N^-_{T-x}(y)| \geq n - 2$. By induction, Theorem 7.6.4 and hence Corollary 7.6.5 holds for $T - \{x, y\}$. Thus some vertex in $N^-_T(y)$ is an origin of $**P$. Hence x is an origin of P in T (using $x \rightarrow y$ and an arc into y from the origin of $**P$ in $T - \{x, y\}$). So we may assume that $|N^+(y)| \geq 2$. Let $U = R^-_{T-y}(N^+(y))$ and $W = V - U - \{x, y\}$. Then $W \mapsto \{x, y\}$ and $U \mapsto W \cup \{x\}$. If $|U| \geq \ell_2(P) + 1$, then by the same proof as we used above (beginning of Case 2), we get that y is an origin of P. So suppose $|U| \leq \ell_2(P)$. This implies in particular that $\ell_2(P) \geq |N^+(y)| \geq 2$.

If $|W| \geq 2$, then we let $w \in W$ be an in-generator of $T - \{x, y\}$ and take $w' \in W - w$ arbitrary. By induction either w or w' is an origin of the in-path $**P$ (recall that $\ell_2(P) \geq 2$ and hence $**P$ is an in-path). Thus using the arc xy and an arc into y from the origin of $**P$ in W we see that x is the origin of P. Finally consider the case when $|W| = 1$ (note that $|W| = n - 1 - |U| \geq 1$, since $|U| \leq \ell_2(P) \leq n - 2$). Then $|U| = n - 2$ and $\ell_2(P) = n - 2$ (since we assumed above that $\ell_2(P) \geq |U|$). Thus $*P$ is a directed in-path. Using that y is an in-generator of $T - x$, we get that x is an origin of P. This completes the proof of the theorem. □

If the path in Theorem 7.6.4 has $n+1$ vertices instead of n, then the statement is no longer true. However, the exceptions (to the $n+1, n+1$ version of

Theorem 7.6.4) can be characterized [509] and based on this characterization Havet and Thomassé were able to prove that the tournaments in Figure 7.4 are indeed the only tournaments that do not contain every orientation of a hamiltonian path.

In [509] Havet and Thomassé also proved the following result which is of independent interest.

Proposition 7.6.6 *[509] Let P be an out-path on n_1 vertices and Q an in-path on n_2 vertices. Let $T = (V, A)$ be a tournament on $n = n_1 + n_2$ vertices. If $x \in V$ is the origin of a copy of P and of Q in T, then we may choose copies of P and Q such that $V(P) \cap V(Q) = \{x\}$ and x is the origin of both copies.* □

How easy is it to find an occurrence of a prescribed orientation of a hamiltonian path P in a tournament? If P is a directed path, then this can be done in time[9] $O(n \log n)$ (see Section 18.1). Some patterns can be found faster; Bampis, Hell, Manoussakis and Rosenfeld [64] showed that one can find an anti-directed hamiltonian path in $O(n)$ time. This is the best possible as shown in [514]. The following somewhat surprising result by Hell and Rosenfeld shows that finding distinct patterns requires quite different complexities:

Theorem 7.6.7 *[514] For every $0 \leq \alpha \leq 1$ there exists an orientation P of a path on n vertices so that every algorithm which checks for an occurrence of P in a tournament T with n vertices must make $\Omega(n \log^{\alpha} n)$ references to the adjacency matrix of T in the worst case.* □

Based on Theorem 7.6.3 Havet proved the following result:

Theorem 7.6.8 *[505] There is an $O(n^2)$ algorithm that takes as input a tournament on $n \geq 8$ vertices and an oriented path P on at most n vertices and returns an occurrence of P in T.* □

It is not known whether there are orientations of paths that in the worst case need $\Omega(n^{1+\epsilon})$ references (for some $\epsilon > 0$) to the adjacency matrix to be found in a tournament. By this we mean that in some cases one needs that many steps to either find the desired path or conclude that no such path exists.

Instead of considering orientations of hamiltonian paths in tournaments, one may just as well consider orientations of hamiltonian cycles in tournaments. However, one particular cycle, namely, the directed hamiltonian cycle, can only be found in strong tournaments. Rosenfeld [788] conjectured that the directed hamiltonian cycle is the only orientation of a hamiltonian cycle that can be avoided by tournaments on arbitrarily many vertices. This conjecture was settled by Thomason who proved the following:

[9] We remind the reader that in measuring the complexity, we only count how many times we have to ask about the orientation of a given arc.

Theorem 7.6.9 *[851] Every tournament on $n \geq 2^{128}$ vertices contains every oriented cycle of length n except possibly the directed hamiltonian cycle.* □

Thomason also conjectured that the correct value of the lower bound on n is 9. One easily obtains a tournament with 8 vertices having no anti-directed hamiltonian cycle by adding a new vertex v to the tournament on 7 vertices in Figure 7.4 and joining v arbitrarily to the other 7 vertices. Hence 9 would be best possible if true.

Using the methods developed in [509] along with a number of new ideas, Havet [506] proved the following result. Recall that every strong tournament has a hamiltonian cycle.

Theorem 7.6.10 *[506] Every tournament T on $n \geq 68$ vertices contains every oriented cycle of length n, except possibly the directed hamiltonian cycle.* □

Hell and Rosenfeld [515] gave a polynomial algorithm for testing the existence of an anti-directed hamiltonian path P with prescribed end-vertices in a tournament. Note that it is not specified that P must be an out-path.

Problem 7.6.11 *[515] Extend the method of [515] to obtain a polynomial algorithm for deciding whether a given tournament T with vertices x, y has an ADH-path P starting in x and ending in y such that P is an out-path from x.*

Bampis, Hell, Manoussakis and Rosenfeld proved that if the last vertex is not specified, then the problem is polynomially solvable [64].

Problem 7.6.12 *[515] Is it true that for any out-path P on n vertices such that no interval of P is larger than k, any tournament T on n vertices and any prescribed vertex x of T which has out-degree at least $k+1$, there is an occurrence of P in T which starts in x on a forward arc?*

As mentioned in Section 7.3, Thomassen [856] proved that there are 3-strong tournaments with no (x, y)-hamiltonian path for some choice of vertices x and y, but that every 4-strong tournament contains such a path. As pointed out in [515] a much weaker condition suffices to guarantee that a tournament T contains an anti-directed hamiltonian path with prescribed end vertices.

Theorem 7.6.13 *[515] Every tournament T with $\delta^0(T) \geq 4$ contains an anti-directed hamiltonian path with prescribed end-vertices.* □

Problem 7.6.14 *[515] Can the minimum degree bound of 4 above be lowered to 3 or even 2?*

Problem 7.6.15 *[76] Find a sufficient condition for a tournament to contain two arc-disjoint ADH-paths with the same end-vertices.*

Not surprisingly, if a digraph is almost complete, then it will contain all orientations of a hamiltonian undirected path. The following result is due to Heydemann, Sotteau and Thomassen:

Theorem 7.6.16 *[526] Let D be a digraph on n vertices and at least $(n-1)(n-2)+3$ arcs and let C be an arbitrary orientation of a cycle of length n. Then D contains a copy of C, except for the case when D is not strong and C is a directed hamiltonian cycle.* □

7.7 Exercises

7.1. Prove that if D is a strong semicomplete digraph with distinct vertices x, y such that $D-x$ and $D-y$ are strong but $D-\{x,y\}$ is not strong, then D has an (x,y)-hamiltonian path and a (y,x)-hamiltonian path.

7.2. **(−)** Prove that, from a complexity point of view, the hamiltonian path problem, the $[x,y]$-hamiltonian path problem and the (x,y)-hamiltonian path problem are all equivalent. That is, each of them can be reduced in polynomial time to each of the two others.

7.3. Show how to decide in time $O(\sqrt{n}m)$ whether or not a given input digraph D with special vertices x, y contains a 1-path-cycle factor such that the path is a path between x and y. Hint: use flows.

7.4. Derive Corollary 7.2.2 from Theorem 7.2.1.

7.5. Prove Lemma 7.2.3.

7.6. Prove the last claim in the proof of Corollary 7.2.7.

7.7. Derive Theorem 7.2.6 from Theorem 7.2.4.

7.8. **2-regular 2-strong locally semicomplete digraphs.** Prove that for every $n \geq 5$ there exists (up to isomorphism) precisely one 2-strong and 2-regular locally semicomplete digraph, namely, the second power $\vec{C}_n^2$ of an n-cycle.

7.9. Prove that the second power $\vec{C}_n^2$ of an n-cycle has a unique hamiltonian cycle. Next, prove that $\vec{C}_n^2$ is not weakly hamiltonian-connected.

7.10. Prove Lemma 7.2.8.

7.11. Prove that if D is the second power $\vec{P}_{2k+1}^2$ of an odd path $P = u_1u_2 \dots u_{2k+1}$, then there is no pair of disjoint (u_1, u_{2k})-, (u_2, u_{2k+1})-paths in D.

7.12. Prove Theorem 7.2.11.

7.13. Suppose $D = (V, A)$ is a non-strong locally semicomplete digraph with strong decomposition D_1, D_2, D_3, D_4 such that $D-x$ is connected for every $x \in V$. Let $u_i \in V(D_i)$ be specified for each $i = 1, 2, 3, 4$. Prove that D contains disjoint (u_1, u_3)-,(u_2, u_4)-paths P, Q so that $V = V(P) \cup V(Q)$.

7.14. (+) Prove the following: Let T be a 2-strong semicomplete digraph and x, y vertices of T, such that $T-x$ and $T-y$ are both 2-strong, $x \not\to y$, and neither x nor y is contained in a 2-cycle. If $T-\{x,y\}$ is not 2-strong, then T has an (x,y)-hamiltonian path. Hint: consider a minimal separator of the form $\{u,x,y\}$.

7.15. (+) Prove Proposition 7.3.3.

7.16. (−) **Hamiltonian cycles containing a prescribed arc in semicomplete digraphs**. Use Theorem 7.3.2 to show that every 3-strong semicomplete digraph $D=(V,A)$ has a cycle containing the arc a for any prescribed arc $a \in A$.

7.17. (++) Prove Theorem 7.3.5.

7.18. **Longest $[x,y]$-paths in tournaments**. Find a characterization for the length of a longest $[x,y]$-path in a tournament. Hint: use Theorem 7.2.1.

7.19. For every $p \geq 1$, construct an infinite family $\mathcal{S}$ of strong tournaments which satisfy that $\delta^0(T) \geq p$ for each $T \in \mathcal{S}$ and there is some arc $a \in A(T)$ which belongs to every hamiltonian cycle of T. Extend your construction to work also for arbitrary high arc-strong connectivity.

7.20. Prove Proposition 7.4.2.

7.21. (+) **Hamiltonian cycles in almost acyclic digraphs.** Prove that for every fixed k there is a polynomial algorithm to decide whether there is a hamiltonian cycle in a given digraph D, which is obtained from an acyclic digraph $H=(V,A)$ by adding a set S of k new vertices and some arcs of the form st where $s \in S$ and $t \in V \cup S$. Hint: use the fact that the k-linkage problem is polynomial for acyclic digraphs (see Theorem 10.4.1).

7.22. Let D be constructed as in Exercise 7.21. Show that if k is not fixed (that is, k is part of the input), then the problem above is $\mathcal{NP}$-complete.

7.23. Let T be a tournament, let $Y_1, Y_2, \ldots, Y_s$ $(s \geq 1)$ be disjoint sets of vertices in T and let x and y be arbitrary distinct vertices in $V(T)-(Y_1 \cup Y_2 \cup \ldots \cup Y_s)$. Prove that if there exist k disjoint (x,y)-paths in T, then there exist at least $k-\sum_{i=1}^{s} \lfloor |Y_i|/2 \rfloor$ disjoint (x,y)-paths in $T-\cup_{i=1}^{s} A(T\langle Y_i \rangle)$.

7.24. (+) Let $X_1, X_2, \ldots, X_p$ and D be defined as in Theorem 7.4.8. Prove that D is strong. Hint: first prove the following two claims and then combine them into a proof that D is strong:

(a) If $x \in X_i$ and $y \in X_j$ $(1 \leq i \neq j \leq l)$, then there are $\lfloor |X_i|/2 \rfloor + \lfloor |X_j|/2 \rfloor + \lceil |X_l|/2 \rceil$ disjoint (x,y)-paths in $D_{i,j}$.

(b) If $x, y \in X_i$ $(x \neq y)$, then there are $|X_i|$ disjoint (x,y)-paths in D_i. Furthermore there is an (x,y)-path in D (Bang-Jensen, Gutin and Yeo [96]).

7.25. (+) Prove that the digraph D in Theorem 7.4.8 has a cycle factor [96]. Hint: let D' be obtained from D by the vertex-splitting technique (Section 4.2). Form a network from D' by putting lower bound 1 on arcs of the kind $v_t v_s$, $v \in V(D)$ and zero elsewhere. Put capacity 1 on arcs of the kind $v_t v_s$ and ∞ on all other arcs. Now apply Theorem 4.8.2 and deduce the result from the structure one can derive using a presumed bad cut $(S, \overline{S})$.

7.26. (+) Prove that the digraph D in Theorem 7.4.8 is hamiltonian [96]. Hint: consider any irreducible factor. Apply Theorem 6.6.20 and conclude that the cycle factor is a hamiltonian cycle.

7.27. Prove Proposition 7.5.1.

7.28. Show that there is a polynomial algorithm for checking whether a semicomplete digraph d has a hamiltonian path through a given arc xy. Hint: you can reduce the problem to the problem of checking for a hamiltonian cycle through a given arc in a semicomplete digraph.

7.29. Show by an example that $s+1$ cannot be replaced by s in Proposition 7.4.16.

7.30. **Orientations of paths in strong tournaments.** Prove the following statement: Let T be a strong tournament on n vertices and P an out-path on $n-1$ vertices. Then
(a) every vertex of T except possibly one is an origin of P and
(b) if $\ell_1(P) \geq 2$, then every vertex of out-degree at least 2 is an origin of P.

7.31. **Orientations of paths in 2-strong tournaments.** Let T be a 2-strong tournament on n vertices and let P be an oriented path on $n-1$ vertices. Prove that every vertex of T is an origin of P.

8. Paths and Cycles of Prescribed Lengths

In this chapter we study the following five topics:

- Pancyclicity, vertex-pancyclicity and arc-pancyclicity of digraphs (Section 8.1). Most of the results are on tournaments and their generalizations.
- Efficient algorithms for finding short paths and cycles (Section 8.2). We demonstrate how the colour coding technique can be applied to design efficient algorithms for short paths and cycles.
- Cycles of length k modulo p (Section 8.3). We discuss the even cycle problem and sufficient conditions for the existence of cycles of length k modulo p.
- Girth (Section 8.4). We present mainly results related to the Caccetta-Häggkvist and Hoáng-Reed conjectures.
- Short cycles in semicomplete digraphs (Section 8.5). We overview results on k-cycles in p-partite tournaments for $k \leq p$.

8.1 Pancyclicity of Digraphs

A digraph D of order n is **pancyclic** if it has cycles of all lengths $3, 4, \ldots, n$. We say that D is **vertex-pancyclic** (**arc-pancyclic**, respectively) if for every $v \in V(D)$ ($a \in A(D)$, respectively) and every $k \in \{3, 4, \ldots, n\}$ there is a cycle of length k containing v. We also say that D is **(vertex-)m-pancyclic** if D contains a k-cycle (every vertex of D is on a k-cycle) for each $k = m, m+1, \ldots, n$. Note that some early papers on pancyclicity in digraphs require that D is (vertex-)2-pancyclic in order to be (vertex-)pancyclic (see e.g. the survey [152] by Bermond and Thomassen). We feel that this definition is too restrictive, since often one can prove pancyclicity results for much broader classes of digraphs when the 2-cycle is omitted from the requirement.

In Subsection 8.1.1, we consider pancyclicity in degree-constrained digraphs. Pancyclicity in extended semicomplete, quasi-transitive and locally semicomplete digraphs is studied in Subsections 8.1.2 and 8.1.3. Pancyclicity in other classes of digraphs is overviewed in Subsection 8.1.4. Cycle extendability is introduced in Subsection 8.1.5 and arc-pancyclicity is studied in Subsection 8.1.6.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_8,

8.1.1 (Vertex-)Pancyclicity in Degree-Constrained Digraphs

The following assertion is due to Alon and Gutin:

Proposition 8.1.1 *[24] Every digraph $D = (V, A)$ on n vertices for which $\delta^0(D) \geq n/2 + 1$ is vertex-2-pancyclic.*

Proof: Let $v \in V$ be arbitrary. By Corollary 6.4.3 there is a Hamilton cycle $u_1u_2 \ldots u_{n-1}u_1$ in $D - v$. If there is no cycle of length k through v, then for every i, $|N^+(v) \cap \{u_i\}| + |N^-(v) \cap \{u_{i+k-2}\}| \leq 1$, where the indices are computed modulo $n-1$. By summing over all values of i, $1 \leq i \leq n-1$, we conclude that $|N^-(v)| + |N^+(v)| \leq n-1$, contradicting the assumption that all in-degrees and out-degrees exceed $n/2$. □

The following analogue of Proposition 8.1.1 is an easy consequence of Corollary 6.4.6. It was proved by Randerath, Schiermeyer, Tewes and Volkmann [762].

Proposition 8.1.2 *Every digraph $D = (V, A)$ on $n \geq 3$ vertices for which $\delta^0(D) \geq (n+1)/2$ is vertex-pancyclic.* □

Thomassen [853] proved that just by adding one to the degree condition for hamiltonicity in Theorem 6.4.7 one obtains cycles of all possible lengths in the digraphs satisfying the degree condition.

Theorem 8.1.3 *[853] Let D be a strong digraph on n vertices such that $d(x) + d(y) \geq 2n$ whenever x and y are nonadjacent. Then either D has cycles of all lengths $2, 3, \ldots, n$, or D is a tournament (in which case it has cycles of all lengths $3, 4, \ldots, n$) or n is even and D is isomorphic to $\overleftrightarrow{K}_{\frac{n}{2},\frac{n}{2}}$.* □

The following example from [853] shows that $2n$ cannot be replaced by $2n-1$ in Theorem 8.1.3. For some $m \leq n$ let $D_{n,m} = (V, A)$ be the digraph with vertices $V = \{v_1, v_2, \ldots, v_n\}$ and arcs $A = \{v_iv_j | i < j \text{ or } i = j+1\} - \{v_iv_{i+m-1} | 1 \leq i \leq n-m+1\}$. We leave it as Exercise 8.1 to show that $D_{n,m}$ is strong, has no m-cycle and if $m > (n+1)/2$, then $D_{n,m}$ satisfies Meyniel's condition for hamiltonicity (Theorem 6.4.7). In [244] Darbinyan characterizes those digraphs which satisfy Meyniel's condition, but are not pancyclic.

Theorem 8.1.3 extends Moon's theorem (Theorem 1.5.1) and Corollaries 6.4.2 and 6.4.6. However, as pointed out by Bermond and Thomassen in [152], Theorem 8.1.3 does not imply Meyniel's theorem (Theorem 6.4.7). The following result is due to Häggkvist:

Theorem 8.1.4 *[490] Every hamiltonian digraph on n vertices and at least $\frac{1}{2}n(n+1) - 1$ arcs is pancyclic.* □

Song [830] generalized the result of Jackson given in Theorem 6.10.2 and proved the following theorem.

Theorem 8.1.5 *[830] Let $D = (V, A)$ be an oriented graph on $n \geq 9$ vertices with minimum degree $n - 2$. Suppose that D satisfies the following property:*

$$xy \notin A \Rightarrow d^+(x) + d^-(y) \geq n - 3. \tag{8.1}$$

Then D is pancyclic. □

Song [830] pointed out that if the minimum degree condition in Theorem 8.1.5 is relaxed, then it is no longer guaranteed that D is hamiltonian.

Using Theorems 8.1.5 and 8.5.3, Bang-Jensen and Guo proved that under the same conditions as in Theorem 8.1.5 the digraph is in fact vertex-pancyclic.

Theorem 8.1.6 *[79] Let D be an oriented graph on $n \geq 9$ vertices and suppose that D satisfies the conditions in Theorem 8.1.5. Then D is vertex pancyclic.* □

It should be noted that every digraph which satisfies the condition of Theorem 8.1.5 is a multipartite tournament with independence number at most 2.

There are several other results on pancyclicity of digraphs with large minimum degrees, see e.g. the papers [242, 243, 246] by Darbinyan.

8.1.2 Pancyclicity in Extended Semicomplete and Quasi-Transitive Digraphs

In this subsection we show how to use the close relationship between the class of quasi-transitive digraphs and the class of extended semicomplete digraphs to derive results on pancyclic and vertex-pancyclic quasi-transitive digraphs from analogous results for extended semicomplete digraphs.

A digraph D is **triangular with partition** V_0, V_1, V_2, if the vertex set of D can be partitioned into three disjoint sets V_0, V_1, V_2 with $V_0 \mapsto V_1 \mapsto V_2 \mapsto V_0$. Note that this is equivalent to saying that $D = \vec{C}_3[D\langle V_0\rangle, D\langle V_1\rangle, D\langle V_2\rangle]$.

Gutin [456] characterized pancyclic and vertex-pancyclic extended semicomplete digraphs. Clearly no extended semicomplete digraph of the form $D = \vec{C}_2[\overline{K}_{n_1}, \overline{K}_{n_2}]$ with at least 3 vertices is pancyclic since all cycles are of even length. Hence we must assume that there are at least 3 partite sets in order to get a pancyclic extended semicomplete digraph. It is also easy to see that the (unique) strong 3-partite extended semicomplete digraph on 4 vertices is not pancyclic (since it has no 4-cycle). These observations and the following theorem completely characterize pancyclic and vertex-pancyclic extended semicomplete digraphs.

Theorem 8.1.7 *[456] Let D be a hamiltonian extended semicomplete digraph of order $n \geq 5$ with k partite sets ($k \geq 3$). Then*

(a) D is pancyclic if and only if D is not triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs, such that either $|V_0| = |V_1| = |V_2|$ or no $D\langle V_i\rangle$ $(i = 0, 1, 2)$ contains a path of length 2.

(b) D is vertex-pancyclic if and only if it is pancyclic and either $k > 3$ or $k = 3$ and D contains two cycles Z, Z' of length 2 such that $Z \cup Z'$ has vertices in the three partite sets. □

It is not difficult to see that Theorem 8.1.7 extends Theorem 1.5.1, since no semicomplete digraph on $n \geq 5$ vertices satisfies any of the exceptions from (a) and (b).

The next two lemmas by Bang-Jensen and Huang [103] concern cycles in triangular digraphs. They are used in the proof of Theorem 8.1.10 which characterizes pancyclic and vertex-pancyclic quasi-transitive digraphs.

Lemma 8.1.8 *[103] Suppose that D is a triangular digraph with a partition V_0, V_1, V_2 and suppose that D is hamiltonian. If $D\langle V_1\rangle$ contains an arc xy and $D\langle V_2\rangle$ contains an arc uv, then every vertex of $V_0 \cup \{x, y, u, v\}$ is on cycles of lengths $3, 4, \ldots, n$.*

Proof: Let C be a hamiltonian cycle of D. We construct an extended semicomplete digraph D' from D in the following way. For each of $i = 0, 1, 2$, first path-contract[1] each maximal subpath of C which is contained in $D\langle V_i\rangle$ and then delete the remaining arcs of $D\langle V_i\rangle$. It is clear that D' is a subdigraph of D, and in this process, C is changed to a hamiltonian cycle C' of D'. Hence D' is also triangular with a partition V_0', V_1', V_2' such that $|V_0'| = |V_1'| = |V_2'| = r$, for some r (the last fact follows from the existence of a hamiltonian cycle in D'). Then each vertex of D is on a cycle of length k with $3r \leq k \leq |V(D)|$ (to see this, just use suitable pieces of the r subpaths of C in each V_i).

Now we may assume that $r \geq 2$ and we show that each vertex of $V_0 \cup \{x, y, u, v\}$ is on a cycle of length k with $3 \leq k \leq 3r-1$. To see this, we modify D' to another digraph D'' as follows. If x and y are in distinct maximal subpaths P_x, P_y of C in $D\langle V_1\rangle$, then we add (in D') an arc from the vertex to which P_x was contracted to the vertex to which P_y was contracted. If x and y are in the same maximal subpath P of C in $D\langle V_1\rangle$, then we add (in D') an arc from the vertex to which P was contracted to an arbitrary other vertex of V_1'. For the vertices u and v we make a similar modification. Hence we obtain a digraph D'' which is isomorphic to a subdigraph of D. The digraph D'' is also triangular with a partition V_0'', V_1'', V_2'' such that $|V_0''| = |V_1''| = |V_2''| = r$. Moreover $D''\langle V_1''\rangle$ contains an arc $x'y'$ and $D''\langle V_2''\rangle$ contains an arc $u'v'$. It is clear now that each vertex of $V_0'' \cup \{x', y', u', v'\}$ is on a cycle of length k where $3 \leq k \leq 3r - 1$. Using the same structure as for these cycles we can see that in D each vertex of $V_0 \cup \{x, y, u, v\}$ is on a cycle of length k with $3 \leq k \leq 3r - 1$. □

[1] Recall the definition of path-contraction from Section 1.3.

Lemma 8.1.9 *[103] Suppose that D is a triangular digraph with a partition V_0, V_1, V_2 and D has a hamiltonian cycle C. If $D\langle V_0\rangle$ contains an arc of C and a path P of length 2, then every vertex of $V_1 \cup V_2 \cup V(P)$ is on cycles of lengths $3, 4, \ldots, n$.*

Proof: Exercise 8.5. □

It is easy to check that a strong quasi-transitive digraph on 4 vertices is pancyclic if and only if it is a semicomplete digraph. For $n \geq 5$ we have the following characterization due to Bang-Jensen and Huang:

Theorem 8.1.10 *[103] Let $D = (V, A)$ be a hamiltonian quasi-transitive digraph on $n \geq 5$ vertices.*

(a) D is pancyclic if and only if it is not triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs, such that either $|V_0| = |V_1| = |V_2|$, or no $D\langle V_i\rangle$ $(i = 0, 1, 2)$ contains a path of length 2.

(b) D is not vertex-pancyclic if and only if D is not pancyclic or D is triangular with a partition V_0, V_1, V_2 such that one of the following occurs:

(b1) $|V_1| = |V_2|$, both $D\langle V_1\rangle$ and $D\langle V_2\rangle$ have no arcs, and there exists a vertex $x \in V_0$ such that x is not contained in any path of length 2 in $D\langle V_0\rangle$ (in which case x is not contained in a cycle of length 5).

(b2) one of $D\langle V_1\rangle$ and $D\langle V_2\rangle$ has no arcs and the other contains no path of length 2, and there exists a vertex $x \in V_0$ such that x is not contained in any path of length 1 in $D\langle V_0\rangle$ (in which case x is not contained in a cycle of length 5).

Proof: To see the necessity of the condition in (a), suppose that D is triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs. If $|V_0| = |V_1| = |V_2|$, then D contains no cycle of length $n - 1$. If no $D\langle V_i\rangle$ $(i = 0, 1, 2)$ contains a directed path of length 2, then D contains no cycle of length 5.

Now we prove the sufficiency of the condition in (a). According to Theorem 2.7.5, there exists a semicomplete digraph T on k vertices for some $k \geq 3$ such that D is obtained from T by substituting a quasi-transitive digraph H_v for each vertex $v \in V(T)$ (here H_v is non-strong if it has more than one vertex). Let C be a hamiltonian cycle of D. We construct an extended semicomplete digraph D' from D in the following way: for each $H_v, v \in V(T)$, first path-contract each maximal subpath of C which is contained in H_v and then delete the remaining arcs of H_v. In this process C is changed to a hamiltonian cycle C' of D'.

Suppose D is not pancyclic. Then it is easy to see that D' is not pancyclic. By Theorem 8.1.7, D' is triangular with a partition V_0', V_1', V_2'. Let $V_i \subset V$ be obtained from V_i', $i = 0, 1, 2$, by substituting back all vertices on contracted subpaths of C. Then D is triangular with partition V_0, V_1, V_2. Moreover each $D\langle V_i\rangle$ is covered by r disjoint subpaths of C for some r.

By Lemma 8.1.8, two of V_0, V_1, V_2, say V_1 and V_2, induce subdigraphs with no arcs in D. If $|V_0| = |V_1| = |V_2|$ we have the first exception in (a). Hence we may assume that $|V_0| > |V_1| = |V_2|$. Then $D\langle V_0\rangle$ contains an arc of C. From Lemma 8.1.9, we see that $D\langle V_0\rangle$ contains no path of length 2. This completes the proof of (a).

The proof of (b) is left to the reader as Exercise 8.6. □

8.1.3 Pancyclic and Vertex-Pancyclic Locally Semicomplete Digraphs

We saw in the last subsection how the structure theorem for quasi-transitive digraphs (i.e., Theorem 2.7.5) was helpful in finding a characterization for (vertex-)pancyclic quasi-transitive digraphs. Now we show that the structure theorem for locally semicomplete digraphs (Theorem 2.10.15) is also very useful for finding a characterization of those locally semicomplete digraphs which are (vertex-)pancyclic. Our first goal (Lemma 8.1.14) is a characterization of those round decomposable locally semicomplete digraphs which are (vertex-)pancyclic.

Lemma 8.1.11 *Let R be a strong round local tournament and let C be a shortest cycle of R and suppose C has $k \geq 3$ vertices. Then for every round labelling $v_0, v_1, \ldots, v_{n-1}$ of R such that $v_0 \in V(C)$ there exist indices $0 < a_1 < a_2 < \ldots < a_{k-1} < n$ so that $C = v_0 v_{a_1} v_{a_2} \ldots v_{a_{k-1}} v_0$.*

Proof: Let C be a shortest cycle and let $\mathcal{L} = v_0, v_1, \ldots, v_{n-1}$ be a round labelling of R so that $v_0 \in V(C)$. If the claim is not true, then there exists a number $2 \leq l < k-1$ so that $C = v_0 v_{a_1} v_{a_2} \ldots v_{a_{k-1}} v_0$, where $0 < a_1 < \ldots < a_{l-1}$ and $a_l < a_{l-1}$. Now the fact that $\mathcal{L}$ is a round labelling of R implies that $v_{l-1} \rightarrow v_0$, contradicting the fact that C is a shortest cycle. □

Recall that the girth $g(D)$ of a digraph is the length of a shortest cycle in $D = (V, A)$. For a vertex $v \in V$ we let $g_v(D)$ denote the length of a shortest cycle in D that contains v. The next lemma shows that every round local tournament R is $g(R)$-pancyclic.

Lemma 8.1.12 *A strong round local tournament digraph R on r vertices has cycles of length $k, k+1, ..., r$, where $k = g(R)$.*

Proof: By Lemma 8.1.11 we may assume that R contains a cycle of the form $v_{i_1} v_{i_2} ... v_{i_k} v_{i_1}$, where $0 = i_1 < i_2 < ... < i_k < r$. Because D is strong, v_{i_m} dominates all the vertices $v_{i_m+1}, \ldots, v_{i_{m+1}}$ for $m = 1, 2, ..., k$. Now it is easy to see that D has cycles of lengths $k, k+1, ..., r$ through the vertices $v_{i_1}, v_{i_2}, ..., v_{i_k}$. □

There is also a very nice structure on cycles through a given vertex in a round local tournament digraph. We leave the proof as Exercise 8.7.

Lemma 8.1.13 *If a strong round locally tournament digraph with r vertices has a cycle of length k through a vertex v, then it has cycles of all lengths $k, k+1, \ldots, r$ through v.* □

Lemma 8.1.14 *[80] Let D be a strongly connected round decomposable locally semicomplete digraph with round decomposition $D = R[S_1, \ldots, S_p]$. Let $V(R) = \{r_1, r_2, \ldots, r_p\}$, where r_i is the vertex of R corresponding to S_i. Then*

(1) D is pancyclic if and only if either the girth of R is 3 or $g(R) \leq \max_{1 \leq i \leq p} |V(S_i)| + 1$.
(2) D is vertex-pancyclic if and only if, for each $i = 1, ..., p$, either $g_{r_i}(R) = 3$ or $g_{r_i}(R) \leq |V(S_i)| + 1$.

Proof: As each S_i is semicomplete, it has a hamiltonian path P_i. Furthermore, since R is a strong locally semicomplete digraph, it is hamiltonian by Theorem 6.3.1. Thus, starting from a p-cycle with one vertex from each S_i, we can get cycles of all lengths $p+1, p+2, \ldots, n$, by taking appropriate pieces of hamiltonian paths $P_1, P_2, \ldots, P_p$ in $S_1, \ldots, S_p$. Thus, if $g(R) = 3$, then D is pancyclic by Lemma 8.1.12. If $g(R) \leq \max_{1 \leq i \leq r} |V(S_i)| + 1$, then D is pancyclic by Lemma 8.1.12 and the fact that (by Moon's theorem) every S_i has cycles of lengths $3, 4, \ldots, |V(S_i)|$. If $g(R) > 3$ and, for every $i = 1, ..., r$, $g(R) > |V(S_i)| + 1$, then D is not pancyclic since it has no $(g(R)-1)$-cycle. The second part of the lemma can be proved analogously by first proving that for each $i = 1, 2, \ldots, p$, every vertex in S_i is on cycles of all lengths $g_{r_i}(R), g_{r_i}(R)+1, \ldots, n$ (using Lemma 8.1.13) and then applying Theorem 1.5.1. □

The main part of the characterization of (vertex-)pancyclic locally semicomplete digraphs is to prove the following lemma (recall Theorem 2.10.15).

Lemma 8.1.15 *[80] Let D be a strong locally semicomplete digraph on n vertices which is not round decomposable. Then D is vertex-pancyclic.*

Proof: If D is semicomplete, then the claim follows from Moon's theorem. So we assume that D is not semicomplete. Thus, D has the structure described in Lemma 2.10.14.

Let S be a minimal separating set of D such that $D - S$ is not semicomplete and let $D_1, D_2, ..., D_p$ be the acyclic ordering of the strong components of $D - S$. Since the subdigraph $D\langle S \rangle$ is semicomplete, it has a unique acyclic ordering $D_{p+1}, ..., D_{p+q}$ with $q \geq 1$ of its strong components. Recalling Lemma 2.10.14(a), the semicomplete decomposition of $D - S$ contains exactly three components D'_1, D'_2, D'_3. Recall that the index of the initial component of D'_2 is λ_2. From Theorem 2.10.8 and Lemma 2.10.12, we see that $D'_2 \Rightarrow D'_1 \Rightarrow S \Rightarrow D_1$ and there is no arc between D'_1 and D'_3.

We first consider the spanning subdigraph D^* of D which is obtained by deleting all the arcs between S and D'_2. By Lemma 2.10.14, D^* is a round

decomposable locally semicompletedigraph and $D^* = R^*[D_1, D_2, \ldots, D_{p+q}]$, where R^* is the round locally semicomplete digraph obtained from D^* by contracting each D_i to one vertex (or, equivalently, R^* is the digraph obtained by keeping an arbitrary vertex from each D_i and deleting the rest). It can be checked easily that $g_v(R^*) \leq 5$ for every $v \in V(R^*)$. Thus D^* is vertex 5-pancyclic by the remark in the proof of Lemma 8.1.14 (in the case when $n = 4$, D is easily seen to be vertex-pancyclic so we may assume $n \geq 5$). Thus, it remains to show that every vertex of D lies on a 3-cycle and a 4-cycle.

We define

$$t = \max\{\, i \mid N^+(S) \cap V(D_i) \neq \emptyset, \lambda_2 \leq i < p\},$$

$$A = V(D_{\lambda_2}) \cup \ldots \cup V(D_t),$$

$$t' = \min\{\, j \mid N^+(D_j) \cap V(D'_2) \neq \emptyset, p+1 \leq j \leq p+q\}$$

$$\text{and } B = V(D_{t'}) \cup \ldots \cup V(D_{p+q}).$$

It follows from Proposition 2.10.16 that $B \mapsto D'_3 \mapsto A$.

Since we have $S \mapsto D_1 \mapsto D_{\lambda_2} \mapsto D'_1 \mapsto S$, every vertex of S is in a 4-cycle and since we have $B \mapsto D'_3 \mapsto A \mapsto D'_1 \mapsto S$, each vertex of $V(D'_3) \cup A \cup V(D'_1)$ is contained in a 4-cycle.

By the definition of t' and A, there is an arc sa from $D_{t'}$ to A. It follows from Lemma 2.10.14(b) that there is an arc $a's'$ from A to B. Let $v \in V(D'_1)$ and $w \in V(D'_3)$ be arbitrarily chosen. Then $savs$ and $s'wa's'$ are 3-cycles.

Suppose D'_2 contains a vertex x that is not in A, then $A \mapsto x$. We also have $x, s' \in N^+(a')$ and this implies that $x \to s'$. From this we get that $x \mapsto D_{t'}$, in particular, $x \to s$. Hence $xsax$ is a 3-cycle and $xvsax$ is a 4-cycle. Thus, there only remains to show that every vertex of $S \cup A$ is contained in a 3-cycle.

Let u be a vertex of S and let D_ℓ be the strong component containing u. If D_ℓ has at least three vertices, then u lies on a 3-cycle by Theorem 1.5.1. So we assume $|V(D_\ell)| \leq 2$. If $\ell < t'$, then u and a' are adjacent because D_ℓ dominates the vertex s' of B. If $\ell \geq t'$, then either $u = s$ or $s \to u$ (if $V(D_\ell) = \{s, u\}$, then usu is a 2-cycle) and hence u, a are adjacent. Therefore, in any case, u is adjacent to one of $\{a, a'\}$. Assume without loss of generality that a and u are adjacent. If $u \to a$, then $uavu$ is a 3-cycle. If $a \to u$, then $uwau$ is a 3-cycle because of $D'_3 \to A$. Hence, every vertex of S has the desired property.

Finally, we note that $S' = N^+(D'_3)$ is a subset of $V(D'_2)$ and it is also a minimal separating set of D. Furthermore, $D - S'$ is not semicomplete. From the proof above, every vertex of S' is also in a 3-cycle. So the proof of the theorem is completed by the fact that $A \subseteq S'$. □

Combining Lemmas 8.1.14 and 8.1.15 we have the following characterization of pancyclic and vertex-pancyclic locally semicomplete digraphs due to Bang-Jensen, Guo, Gutin and Volkmann:

Theorem 8.1.16 *[80] A strong locally semicomplete digraph D is pancyclic if and only if it is not of the form $D = R[S_1, \ldots, S_p]$, where R is a round local tournament digraph on p vertices with $g(R) > \max\{2, |V(S_1)|, \ldots, |V(S_p)|\}+1$. D is vertex-pancyclic if and only if D is not of the form $D = R[S_1, \ldots, S_p]$, where R is a round local tournament digraph with $g_{r_i}(R) > \max\{2, |V(S_i)|\}+1$ for some $i \in \{1, \ldots, p\}$, where r_i is the vertex of R corresponding to S_i.* □

8.1.4 Further Pancyclicity Results

To characterize pancyclic locally in-semicomplete digraphs seems a much harder problem than that of characterizing pancyclic locally semicomplete digraphs. Tewes [848] studied this problem and obtained several partial results of which we will state a few below.

Theorem 8.1.17 *[848] Let D be a locally in-tournament digraph on n vertices and let $3 \leq k \leq n$ be an integer such that $\delta^-(D) > \frac{3n}{2(k+1)} - \frac{1}{2}$. Furthermore, let D be strong if $k \geq 2\delta^-(D)+2$. Then D has a cycle of length k. For $k \geq \sqrt{n+1}$ this bound is sharp.* □

For further results on pancyclic and vertex-pancyclic locally in-tournament digraphs, see [849, 850].

Let the function $f(k)$ be defined as follows for fixed n:

$$f(k) = \begin{cases} \frac{n+1}{k} + \frac{k-1}{2} & \text{if } k \text{ is even} \\ \frac{n+2}{k} + \frac{k-5}{2} & \text{if } k \text{ is odd.} \end{cases}$$

Theorem 8.1.18 *[848] Let D be a strongly connected locally in-tournament digraph on n vertices such that $\delta^-(D) > f(k)$ for some integer $3 \leq k \leq \sqrt{n+1}$. Then D has cycles of all lengths $k, k+1, \ldots, n$.* □

Since every regular tournament is strong (Exercise 8.4) it is also pancyclic by Moon's theorem. Note that by Theorem 6.6.22, every regular multipartite tournament is hamiltonian. This motivated Volkmann to make the following conjecture.

Conjecture 8.1.19 *[890] Every regular p-partite tournament with $p \geq 4$ is pancyclic.*

Note that in the 3-partite tournament $D = \vec{C}_3[\overline{K}_k, \overline{K}_k, \overline{K}_k]$ all cycles have length some multiple of 3. Hence the condition $p \geq 4$ above is necessary. For $p \geq 5$ Conjecture 8.1.19 follows from the next result due to Yeo (for an outline of Yeo's proof see Volkmann [890]).

Theorem 8.1.20 *[918] Every regular multipartite tournament with at least 5 partite sets is vertex-pancyclic.* □

Yeo [920] also proved that all regular 4-partite tournaments with at least 13918 vertices are vertex-pancyclic.

Volkmann [893] raised the following conjecture for regular 3-partite tournaments:

Conjecture 8.1.21 *Every regular semicomplete 3-partite digraph D contains cycles of length $3, 6, \ldots, |V(D)|$.*

The following results support this conjecture: by Theorem 6.6.22, D is hamiltonian.

There are also many results on sufficient conditions in terms of the number of arcs for a digraph to contain a cycle of length precisely k. We refer the reader to the survey of Bermond and Thomassen [152] for a number of references to such results.

Recall that for a given directed pseudograph $D = (V, A)$, the line digraph $L(D)$ of D has vertex set A and $a \to a'$ is an arc in $L(D)$ precisely when the head of a equals the tail of a' in D (note that a loop in D gives rise to a loop in $L(D)$). Let $D = (V, A)$ be a directed pseudograph; D is **pancircular** if it contains a closed trail of length q for every $q \in \{3, 4, ..., |A|\}$. Due to a natural bijection between the set of closed trails in D and the set of cycles in $L(D)$, we obtain the following:

Proposition 8.1.22 *$L(D)$ is pancyclic if and only if D is pancircular.* □

Imori, Matsumoto and Yamada [551], who introduced the notion of pancircularity, proved the following theorem.

Theorem 8.1.23 *Let D be a regular and pancircular directed pseudograph. Then, $L(D)$ is also regular and pancircular.* □

This theorem was used in [551] to show that de Bruijn digraphs are pancyclic and pancircular.

Theorem 8.1.24 *[551] Every de Bruijn digraph $D_B(d, t)$ is pancyclic and pancircular.*

Proof: de Bruijn digraphs $D_B(d, t)$ were introduced for $d \geq 2$ and $t \geq 1$. Let $D_B(d, 0)$ be the directed pseudograph consisting of a singular vertex and d loops. Clearly, $D_B(d, 1) = L(D_B(d, 0))$. Since

$$D_B(d, t + 1) = L(D_B(d, t)) \tag{8.2}$$

for $t \geq 1$ by Lemma 2.5.1, we conclude that (8.2) holds for all $t \geq 0$. We prove the theorem by induction on $t \geq 0$. Clearly, $D_B(d, 0)$ is pancyclic and pancircular. Assume that $D_B(d, t)$ is pancyclic and pancircular. By Theorem 8.1.23, $L(D_B(d, t))$ is pancircular. By Proposition 8.1.22, $L(D_B(d, t))$ is pancyclic. By (8.2), $D_B(d, t + 1) = L(D_B(d, t))$. Thus, $D_B(d, t + 1)$ is pancyclic and pancircular. □

8.1.5 Cycle Extendability in Digraphs

The following definitions are due to Hendry [517]. A non-hamiltonian cycle C in a digraph D is **extendable** if there is some cycle C' with $V(C') = V(C) \cup \{y\}$ for some vertex $y \in V - V(C)$. A digraph D which has at least one cycle is **cycle extendable** if every non-hamiltonian cycle of D is extendable. Clearly a cycle extendable digraph is pancyclic if and only if it contains a 3-cycle and vertex-pancyclic if and only if every vertex is in a 3-cycle.

The following is an easy consequence of the proof of Theorem 1.5.1:

Theorem 8.1.25 *[704] A strong tournament $T = (V, A)$ is cycle extendable unless V can be partitioned into sets U, W, Z such that $W \mapsto U \mapsto Z$ and $T\langle U \rangle$ is strong.* □

Hendry [517] studied cycle extendability in digraphs with many arcs and obtained the next two results.

Theorem 8.1.26 *[517] Every strong digraph on n vertices and at least $n^2 - 3n + 5$ arcs is cycle extendable.* □

Hendry showed that digraphs may have very large in- and out-degree and still not be cycle extendable. This contrasts to the situation for undirected graphs. Hendry has shown in [518, Corollary 8] that, apart from certain exceptions, every graph satisfying Dirac's condition for hamiltonicity ($d(x) \geq n/2$ for every vertex [266]) is also cycle extendable (with the obvious analogous definition of cycle extendability for undirected graphs). The main result of [517] is the following.

Theorem 8.1.27 *[517] Let D be a digraph on $n \geq 7$ vertices such that $\delta^0(D) \geq \frac{2n-3}{3}$. Then D is cycle extendable unless $n = 3r$ for some r and D contains F_n as a spanning subdigraph and D is a spanning subdigraph of G_n. See Figure 8.1 for the definition of F_n, G_n.* □

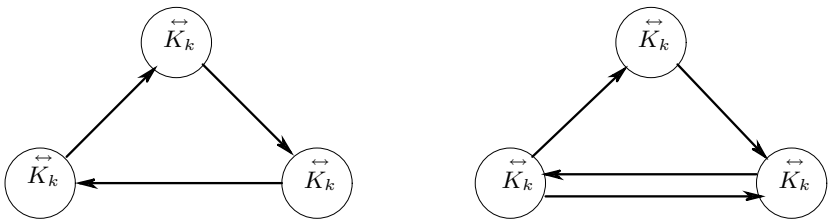


Figure 8.1 The digraphs F_n and G_n. All arcs indicate complete domination in the direction shown.

A cycle C in a digraph D is **1-maximal** if D has no cycle C' such that $C - a$ is a subpath of C' for some arc a of C and $|V(C')| > |V(C)|$. Each of the next two results generalizes Corollary 1.5.2.

Lemma 8.1.28 *Let D be a strong digraph and let C be a 1-maximal cycle in D. Then no vertex of $D - V(C)$ is adjacent to all vertices in C.*

Proof: Exercise 8.12. □

Lemma 8.1.29 *Given a strong digraph D and a vertex $v \in V(D)$ we can find in polynomial time either a hamiltonian cycle of D or a 1-maximal cycle containing v.*

Proof: Exercise 8.13. □

8.1.6 Arc-Pancyclicity

A digraph D of order n is **arc-k-cyclic** for some $k \in \{3, 4, \ldots, n\}$ if each arc of D is contained in a cycle of length k. A digraph $D = (V, A)$ is **arc-pancyclic** if it is arc-k-cyclic for every $k = 3, 4, \ldots, n$. Demanding that a digraph is arc-pancyclic is a very strong requirement, since in particular every arc must be in a hamiltonian cycle. Hence it is not surprising that most results on arc-pancyclic digraphs are for tournaments and generalizations of tournaments. However, Moon proved that almost all tournaments are arc-3-cyclic [704], so for tournaments this is not such a hard requirement, in particular in the light of Theorem 8.1.30 below.

Tian, Wu and Zhang characterized all tournaments that are arc-3-cyclic but not arc-pancyclic. See Figure 8.2 for the definition of the classes $\mathcal{D}_6, \mathcal{D}_8$.

Theorem 8.1.30 *[873] An arc-3-cyclic tournament is arc-pancyclic unless it belongs to one of the families $\mathcal{D}_6, \mathcal{D}_8$ (in which case the arc yx does not belong to a hamiltonian cycle).* □

It is not difficult to derive the following two corollaries from this result:

Corollary 8.1.31 *[873] Every arc-3-cyclic tournament has at most one arc which is not in cycles of all lengths $3, 4, \ldots, n$.*

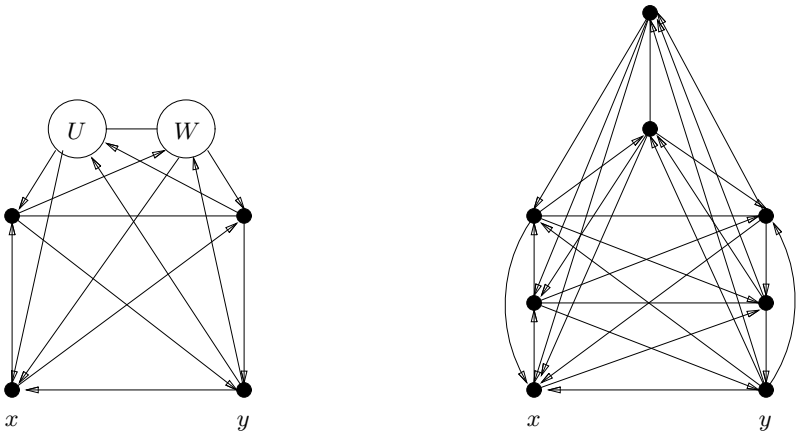
Proof: Exercise 8.14. □

Corollary 8.1.32 *[913] A tournament is arc-pancyclic if and only if it is arc-3-cyclic and arc-n-cyclic.*

Proof: Exercise 8.15. □

The following result due to Alspach is also an easy corollary:

Corollary 8.1.33 *[34] Every regular tournament is arc-pancyclic.* □



$\mathcal{D}_6$ $\mathcal{D}_8$

Figure 8.2 The two families of non-arc-pancyclic arc-3-cyclic tournaments. Each of the sets U and W induces an arc-3-cyclic tournament. All edges that are not already oriented may be oriented arbitrarily, but all arcs between U and W have the same direction.

Finally, observe that since each tournament in the infinite family $\mathcal{D}_6$ is 2-strong and the arc yx is not in any hamiltonian cycle we obtain the following result due to Thomassen:

Theorem 8.1.34 *[856] There exist infinitely many 2-strong tournaments containing an arc which is not in any hamiltonian cycle.* □

In [432, 434] Guo studied arc-pancyclic locally tournament digraphs and obtained several results which generalize those above. In particular he made the important observation that one can in fact get a more general result by studying paths from x to y for all such pairs where the arc xy is not present rather than just those for which the arc yx is present (which is the case for tournaments of course).

Theorem 8.1.35 *[434] Let D be an arc-3-cyclic local tournament and let x, y be distinct vertices such that there is no arc from x to y. Then D contains an (x, y)-path of length k for every k such that $2 \leq k \leq n-1$ unless D is isomorphic to one of the local tournaments T_8^1, T_8^2 (from Section 7.2) or D belongs to one of the families $\mathcal{D}_6$ or $\mathcal{D}_8$, possibly with the arc from y to x missing.* □

The proofs of Theorems 8.1.30 and 8.1.35 are very technical and consist of a long case analysis. Hence it makes no sense to give any of these proofs

here. However, we will finish the section with a proof of the following partial result which Guo used in his proof of Theorem 8.1.35.

Theorem 8.1.36 *[434] Let D be a connected, arc-3-cyclic local tournament which is not 2-strong. Then D is isomorphic to $\vec{C}_3[T_1, T_2, \{s\}]$ where T_i is an arc-3-cyclic tournament for $i = 1, 2$ and s is a vertex. Furthermore, D is arc-pancyclic.*

Proof: First observe that D is strongly connected since it is connected and arc-3-cyclic. Since D is not 2-strong, it has a separating vertex s. Let $T_1, T_2, \ldots, T_k$ denote the acyclic ordering of the strong components of $D - s$. If there is an arc xs from $V(T_1)$ to s, then no arc from x to $V(T_2)$ can be in a 3-cycle. Hence we must have $s \mapsto V(T_1)$ and similarly $V(T_k) \mapsto s$. Since D is arc-3-cyclic, each of T_1, T_k must be an arc-3-cyclic tournament.

If $k \geq 3$, then for every vertex $u \in V(T_2)$, either no arc from $V(T_1)$ to u or no arc from u to $V(T_3)$ can be in a 3-cycle, contradicting our assumption. Thus we must have $k = 2$ and we have proved that $D = \vec{C}_3[T_1, T_2, \{s\}]$.

It remains to prove that D is arc-pancyclic. Since T_1 and T_2 have hamiltonian paths, it is easy to see that each arc which does not belong to either T_1 or T_2 is on cycles of all possible lengths. So we just have to consider arcs inside T_1, T_2. If $|V(T_1)| = |V(T_2)| = 1$, there is nothing more to prove. So suppose without loss of generality that $|V(T_1)| \geq 3$. Let $u_1u_2 \ldots u_ru_1$, $r \geq 3$, be a hamiltonian cycle of T_1. Let u_iu_j be an arbitrary arc of T_1. If $T_1 - u_i$ is strong, then $T_1 - u_i$ has a hamiltonian cycle and hence T_1 has a hamiltonian path starting with the arc u_iu_j. Using this and a hamiltonian path in T_2 we can easily obtain cycles of all lengths $3, 4, \ldots, n$ through u_iu_j in D. Suppose now that $T_1 - u_i$ is not strong. Then $T_1 - u_i$ satisfies the assumption of the theorem, so by induction it has the same structure as D and u_j must belong to the initial component of $T_1 - u_i$. Hence again we find a hamiltonian path starting with the arc u_iu_j in T_1 and finish as above.

Similarly, if $|V(T_2)| \geq 3$, the same proof as above can be applied to every arc of T_2. Thus we have shown that D is arc-pancyclic. □

The following natural and interesting problem remains open.

Problem 8.1.37 *Characterize arc-pancyclic semicomplete digraphs.*

A partial result on the problem was obtained by Darrah, Liu and Zhang [249].

8.2 Colour Coding: Efficient Algorithms for Paths and Cycles

While it is $\mathcal{NP}$-complete to decide whether a digraph D of order n has a path or cycle with n vertices, it is not trivial to see for what functions $l_p(n)$ and

$l_c(n)$, one can verify in polynomial time whether D contains a path (cycle, respectively) of length $l_p(n)$ ($l_c(n)$, respectively). In particular, Papadimitriou and Yannakakis [743] conjectured that one can determine in polynomial time the existence of a path of length $p_l(n) = \Theta(\log n)$. Alon, Yuster and Zwick [31, 32] resolved this conjecture in the affirmative. They also proved that one can check whether a digraph of order n has a $\vec{C}_k$ in polynomial time as long as $k = O(\log n)$. In this section we will briefly consider certain elegant ideas behind the algorithms designed in [31, 32]. Further developments on the topic can be found in [33] and in the references therein. Various algorithmic aspects on enumeration of short cycles are also discussed there. For other applications of colour coding, see, e.g., Liu, Lu, Chen and Sze [649].

We start with a simple technical result on the expectation of a geometric random variable. This result can be found in many books on probability theory; we include its short proof for the sake of completeness. We use Prob(E) to denote the probability of the event E.

Lemma 8.2.1 *Let $0 < p \leq 1$ and let $x_1, x_2, \ldots$ be a sequence of random boolean variables such that $x_j = 1$ with probability p for each $j \geq 1$. A random variable ν is defined as follows: for $j \geq 1$, $\nu = j$ if and only if $x_j = 1$ and $x_1 = x_2 = \ldots = x_{j-1} = 0$. Then, the expectation of ν is $1/p$.*

Proof: The expectation of ν equals

$$\sum_{i=1}^{\infty} i \cdot \text{Prob}(\nu = i) = \sum_{i=1}^{\infty} \text{Prob}(\nu \geq i) = \sum_{i=1}^{\infty} (1-p)^{i-1} = 1/p.$$

□

To design algorithms verifying the existence of paths and cycles, Alon, Yuster and Zwick [31, 32] introduced two methods: the random acyclic subdigraph method and the colour-coding method. We consider first the random acyclic subdigraph method and then the method of colour-coding. In the rest of this section, we will follow [32].

Let $D = (V, A)$ be a digraph with $V = \{u_1, u_2, \ldots, u_n\}$. Let $M = [m_{ij}]$ be the adjacency matrix of D, i.e., $m_{ij} = 1$ if $u_i \rightarrow u_j$ and $m_{ij} = 0$, otherwise. It is well known (see Exercise 3.20) that the (i, j)th entry of the kth power of M is non-zero if and only if there is a (u_i, u_j)-walk of length k. However, many of the (u_i, u_j)-walks of length k can be with repeated vertices (and even arcs). Thus, one naturally asks how we can get rid of walks that are not paths or cycles. One such method is the **random acyclic subdigraph method**: we choose randomly a permutation π on $[n]$ and construct the corresponding acyclic spanning subdigraph H of D by taking the following arcs: $u_{\pi(i)}u_{\pi(j)} \in A(H)$ if and only if $u_{\pi(i)}u_{\pi(j)} \in A$ and $\pi(i) < \pi(j)$. Clearly, every walk of H is a path in D (no vertices can be repeated as H is acyclic). On the other hand, every path P with k arcs in D has a $1/(k+1)!$ chance to be a path in H as well (Exercise 8.18).

Let $O(n^\omega)$ be the complexity of boolean matrix multiplication (i.e., of the multiplication of two boolean $n \times n$ matrices). Due to Coppersmith and Winograd [230], $\omega < 2.376$. Using random acyclic subdigraphs, one can prove the following:

Theorem 8.2.2 *[31, 32] Let $D = (V, A)$ be a digraph that contains a path (a cycle, respectively) of length k. A path (a cycle, respectively) of length k in D can be found in expected time $O((k+1)!\cdot m)$ ($O(k! \log k \cdot n^\omega)$, respectively).*

Proof: To find a path of length k in D one can apply the following algorithm. Choose randomly a permutation π of $[n]$ and construct the corresponding acyclic spanning subdigraph H of D as described above. Using the $O(m)$ algorithm of Subsection 3.3.2, find a longest path P in H. If the length of P is less than k, then repeat the above procedure. Otherwise return a subpath of P whose length is k.

Since D contains $\vec{P}_{k+1}$, H has a path of length at least k with probability at least $1/(k+1)!$. Hence, by Lemma 8.2.1, the expected number of iterations in the above algorithm is at most $(k+1)!$. Thus, the expected running time is $O((k+1)!m)$ as claimed.

To find a cycle of length k in D one can apply the following algorithm. Choose randomly a permutation π on $[n]$ and construct the corresponding acyclic spanning subdigraph H of D as above. By computing (in time $O(n^\omega \cdot \log k)$, see Exercise 3.21) the $(k-1)$th power of the adjacency matrix of H, we find all pairs of vertices which are end-vertices of $(k-1)$-paths in H (see Exercise 8.19). If the terminal vertex of one of the paths dominates the initial vertex of the path in D, we construct the corresponding k-cycle and stop. If no k-cycle is found, we repeat the above procedure.

Clearly, the expected number of iterations in the above algorithm is at most $k!$. This implies the expected running time of $O(k! \log k \cdot n^\omega)$. □

Now we turn our attention to a more powerful approach, the **colour-coding method**. Let $c : V \to [k]$ be a colouring of the vertices of D. A path P in D is **colourful** if no pair of vertices of P are of the same colour.

Lemma 8.2.3 *Let $D = (V, A)$ be a digraph and let $c : V \to [k+1]$ be a colouring of the vertices of D. A colourful path of length k in D, if one exists, can be found in time $2^{O(k)} \cdot m$.*

Proof: Add to D a new vertex s of colour 0 that dominates all vertices of D and is dominated by no vertex. As a result, we obtain a digraph D', which has a $(k+1)$-path starting at s if and only if D contains a path of length k. To find a path of length $k+1$ in D' starting at s we use dynamic programming. Suppose that we have already found for each vertex $v \in V$ the possible sets of colours on colourful (s, v)-paths of length i as well as the corresponding paths (just one path for every possible set). We also call such sets **colourful**. Observe that for every v we have at most $\binom{k+1}{i}$ colourful sets and (s, v)-paths,

respectively. We inspect every colourful set C that belongs to the collection of v and every arc vu. Let $P(C)$ be the corresponding colourful path. If $c(u) \notin C$, then we add $C \cup c(u)$ ($P(C)u$, respectively) to the collection of colourful sets (paths, respectively) of u of cardinality (length, respectively) $i+1$. Clearly, D' contains a colourful $(k+1)$-path with respect to the colouring c if and only if the collection of colourful paths of length $k+1$ for some vertex is not empty. The number of operations of this algorithm is at most

$$O\left(\sum_{i=0}^{k+1} i\binom{k+1}{i}m\right) = O((k+1)2^{k+1}m).$$

□

The next lemma follows from Lemma 8.2.3 and is left as Exercise 8.21.

Lemma 8.2.4 *Let $D=(V,A)$ be a digraph and let $c: V \to [k]$ be a colouring of the vertices of D. For all ordered pairs x, y of distinct vertices colourful (x,y)-paths of length $k-1$ in D, if they exist, can be found in total time $2^{O(k)} \cdot nm$.* □

Actually, for dense digraphs the complexity of this lemma can be improved to $2^{O(k)} \cdot n^{\omega}$ [32]. Clearly, Lemma 8.2.4 implies a $2^{O(k)} \cdot nm$ algorithm to find a k-cycle in D.

If P is a path of order k in D whose vertices are randomly coloured from a set of k colours, then P has a chance of $k!/k^k > e^{-k}$ to become colourful. Thus, by Lemma 8.2.1, the expected number of times to randomly generate k-colouring to detect P is at most $\lfloor e^k \rfloor$. This fact and Lemmas 8.2.3 and 8.2.4 imply the following:

Theorem 8.2.5 (Alon, Yuster and Zwick) *[31, 32] If a digraph D has a path of length k (k-cycle, respectively), then a path of length k (k-cycle, respectively) can be found in $2^{O(k)} \cdot m$ ($2^{O(k)} \cdot nm$, respectively) expected time.* □

The algorithms mentioned in this theorem are quite simple, but unfortunately not deterministic. Fortunately, one can de-randomize these algorithms to obtain deterministic algorithms with time complexity still linear in m. Observe that for a path P of order k in $D=(V,A)$ many k-colourings of V are equally good or bad depending on P being colourful or not. This means that we do not need to consider all k^n k-colourings of V to detect a path of order k in D; a subset S of colourings such that every path of order k is colourful for at least one colouring of S is sufficient. In other words, we wish that for every k-set W of vertices there is a colouring from S that assigns vertices of W different colours.

This is captured in the notion of a *k-perfect family of hash functions* from $[n]$ to $[k]$. Schmidt and Siegel [795] following Fredman, Komlós and Szemerédi

[359] gave an explicit construction of a k-perfect family from $[n]$ to $[k]$ in which each function is specified by $b = O(k) + 2\log_2 \log_2 n$ bits. Thus, the size of the family is $2^b = 2^{O(k)} \log_2^2 n$. The value of each of these functions on each specified element of $[n]$ can be computed in $O(1)$ time. Using this family, the algorithms of Theorem 8.2.5 can be de-randomized to obtain deterministic algorithms running in time $O(2^{O(k)} \cdot m \log^2 n)$ and, respectively, $O(2^{O(k)} \cdot mn \log^2 n)$. Alon, Yuster and Zwick [31, 32] pointed out how to decrease each of the above complexities by the multiplicative factor of $\log n$. They also showed how to de-randomize some versions of algorithms mentioned in Theorem 8.2.2. This implies that the following two parameterized problems are fixed parameter tractable: given an input digraph D and a parameter k, check whether D has a path (cycle, respectively) with at least k vertices (for an introduction to fixed parameter tractability, see Section 18.4).

8.3 Cycles of Length k Modulo p

The linear-time algorithms for computing the period of a digraph described in Section 17.8 show that the problem to verify whether all cycles of a digraph are of length 0 modulo p for some p is polynomial time solvable. This problem has the natural 'existence' analogue: given a (fixed) integer $p \geq 2$, verify whether a digraph D has a cycle of length equal 0 modulo p. In this section, we consider this and the more general problem of the existence of cycles of lengths equal k modulo p. In Subsection 8.3.1, we study the complexity results on these problems; Subsection 8.3.2 is devoted to some sufficient conditions for the existence of cycles of lengths equal k modulo p.

8.3.1 Complexity of the Existence of Cycles of Length k Modulo p Problems

We start our consideration from the following problem. Given a (fixed) integer $p \geq 2$, verify whether a digraph D has a cycle of length equal 0 modulo p. The case of $p = 2$ of this problem is called the Even Cycle problem. The even cycle problem has numerous applications (see e.g. Robertson, Seymour and Thomas [786] and Thomassen [868] and the reference to further literature therein) and is related to several problems on permanents of matrices, so-called Pfaffian orientations of graphs, colouring of hypergraphs, etc. The complexity of the even cycle problem was an open problem for quite some time: Thomassen [869] proved that the even cycle problem is polynomial time solvable for planar digraphs and Galluccio and Loebl [390] extended this result to digraphs, whose underlying undirected graphs do not contain subgraphs contractible to either K_5 or $K_{3,3}$. Finally, independently McCuaig, and Robertson, Seymour and Thomas (see [786]) found highly non-trivial proofs of the following result:

Theorem 8.3.1 *The even cycle problem is polynomial time solvable.* □

We are not aware of any paper determining the complexity of the problem to check whether a digraph has a cycle of length equal 0 modulo p for fixed $p > 2$.

Problem 8.3.2 *Is there a polynomial algorithm to check whether a digraph has a cycle of length equal 0 modulo p for fixed $p > 2$?*

The last problem can be naturally generalized to the problem to verify whether a digraph D has a cycle of length equal k modulo p for fixed k, p such that $0 \leq k < p,\ p \geq 2$. We have considered the case of $k = 0$; the case of $k > 0$ was studied by Arkin, Papadimitriou and Yannakakis [49], who proved the following theorem (observe that the case of $k = 1$ and $p = 2$ is polynomial time solvable since one can check whether a digraph is bipartite in polynomial time):

Theorem 8.3.3 *Let k, p be a pair of fixed integers such that $p > 2$ and $k \in [p-1]$. The problem to verify whether a digraph D has a cycle of length k modulo p is $\mathcal{NP}$-complete.*

Proof: Let D be a digraph and let $k \geq 2$. Choose k arbitrary arcs $a_1, a_2, \ldots, a_k$ in D and replace every arc xy in $A(D) - \{a_1, a_2, \ldots, a_k\}$ by an (x, y)-path of length p, whose intermediate vertices do not belong to D (and the intermediate vertices of all such paths are distinct). Clearly, the obtained digraph D' has a cycle of length equal k modulo p if and only if D has a cycle through all arcs $a_1, a_2, \ldots, a_k$. For a fixed $k \geq 2$, the problem of the existence of a cycle through k given arcs in a digraph is $\mathcal{NP}$-complete (see Proposition 10.1.2 and Theorem 10.2.1); hence this theorem is proved for $k \geq 2$. For $k = 1$, we choose a pair of arcs a, b, replace a by a path of length 2, b by a path of length $p-1$, and every $c \in A(D) - \{a, b\}$ by a path of length p such that all internal vertices of the paths are distinct and distinct from the vertices of D. Clearly, the obtained digraph D' has a cycle of length equal 1 modulo p if and only if D has a cycle through a and b; the last problem is $\mathcal{NP}$-complete as we remarked above. □

Because of this theorem, the following result of Galluccio and Loebl [389] is of certain interest:

Theorem 8.3.4 *Let k, p be a pair of fixed integers such that $p \geq 2$ and $0 \leq k < p$. There is a polynomial algorithm to verify whether a planar digraph D has a cycle of length k modulo p.* □

8.3.2 Sufficient Conditions for the Existence of Cycles of Length k Modulo p

A digraph $D = (V, A)$ is called **even** if, for every $B \subseteq A$, the subdivision of all arcs in B results in a digraph with an even cycle. A ***k*-weak-double-cycle** is a digraph which is defined recursively as follows (see Figure 8.3):

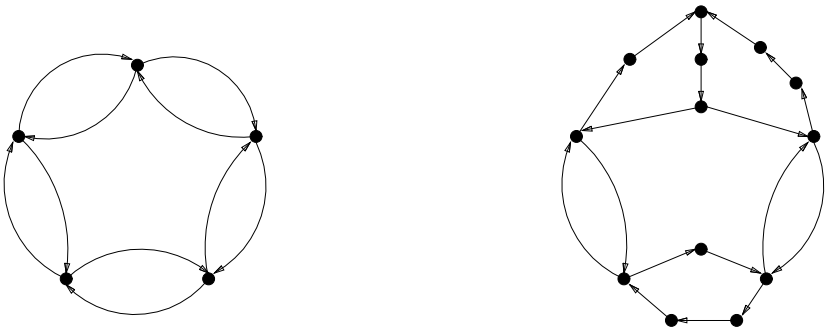


Figure 8.3 The 5-double-cycle and a 5-weak-double-cycle.

1. The complete biorientation $\overleftrightarrow{C}_k$ of a k-cycle is a k-weak-double-cycle.
2. If H is a k-weak-double-cycle and D is obtained from H by subdividing an arc or splitting a vertex [2], then D is a k-weak-double-cycle.

It is easy to see that for odd k a k-weak-double-cycle is even because it has an odd number of cycles and every arc is in an even number of distinct cycles (see Exercise 8.24). The following result is much more difficult to prove.

Theorem 8.3.5 (Seymour and Thomassen) *[813] A digraph is even if and only if it contains a k-weak-double-cycle for some odd k.* □

Galluccio and Loebl [391] have extended this result. They call a digraph $D = (V, A)$ **(k, p)-odd** if, for every $B \subseteq A$, the subdivision of all arcs in B results in a digraph with a cycle of length different from k modulo p.

Theorem 8.3.6 *[391] A digraph is (k, p)-odd if and only if it contains a q-weak-double-cycle, with $(q-2)k \not\equiv 0 \pmod{p}$.* □

Using Theorem 8.3.5 and other results, Thomassen [868] proved the following very interesting theorem:

[2] The operations of subdividing an arc and splitting a vertex are introduced at the end of Section 1.3.

Theorem 8.3.7 (Thomassen's even cycle theorem) *If D is a strong digraph with $\delta^0(D) \geq 3$, then D is even.* □

Koh [605] constructed an infinite family of digraphs D with $\delta^0(D) \geq 2$ and with no even cycle. Thomassen [860] strengthened this result by exhibiting, for every $k \geq 2$, a digraph D_k with $\delta^0(D_k) \geq k$ and with no even cycle. This implies that the strong connectivity requirement in Theorem 8.3.7 is necessary. Theorem 8.3.7 implies that every 3-strong digraph has an even cycle. Thomassen [863] pointed out that there exists a 2-strong digraph of order 7 that has no even cycle, namely, the digraph in Figure 8.4.

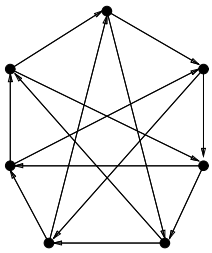


Figure 8.4 A 2-strong digraph with no even cycle.

Thomassen [860] constructed infinitely many 2-strong digraphs that are not even. However, the following question is still open:

Problem 8.3.8 *[863] Are there infinitely many 2-strong digraphs with no even cycle?*

Theorem 8.3.7 was extended by Galluccio and Loebl [391], who proved that every strong digraph D with $\delta^0(D) \geq 3$ contains a cycle of length different from k modulo p, for every $1 \leq k < p,\ p \geq 3$.

Although we do not provide a proof of Theorem 8.3.7, we will prove Theorem 8.3.11 which implies a result weaker than Theorem 8.3.7, i.e.m Corollary 8.3.12, but its assertion is not only on even cycles but also on cycles of length 0 modulo $q(\geq 2)$. To prove Theorem 8.3.11, we need two lemmas; the first lemma is the famous Lovász local lemma (cf. Alon and Spencer [28] or McDiarmid [690]). For an event E, $\overline{E}$ means that E does not hold.

Lemma 8.3.9 (Lovász local lemma) *Let $E_1, \ldots, E_n$ be events in an arbitrary probability space. Suppose that each event E_i is mutually independent of all other events except for at most d events, and that* $\mathrm{Prob}(E_i) \leq p$ *for every $i \in [n]$. If $ep(d+1) \leq 1$, where e is the basis of the natural logarithm, then* $\mathrm{Prob}(\cap_{i=1}^{n} \overline{E}_i) > 0$. □

Lemma 8.3.10 *[26] Let D be a digraph and let $q \geq 2$ be an integer. Suppose that every vertex x of D is assigned a colour $c^*(x)$, an integer in $[q-1]$, such that for every $u \in V(D)$ there exists an out-neighbour v with $c^*(v) \equiv c^*(u)+1$ (mod q), then D contains a cycle of length 0 (mod q).*

Proof: Clearly, choosing an arbitrary vertex u_0 in $V(D)$, we can find a sequence $u_0, u_1, \ldots$ of vertices such that $u_i u_{i+1} \in A(D)$ and $c^*(u_{i+1}) \equiv c^*(u_i) + 1 \pmod q$ for every $i \geq 0$. Let s be the least integer such that $u_j = u_s$ for some $j < s$. It remains to observe that the cycle $u_j u_{j+1} \ldots u_s$ is of length 0 (mod q). □

The following result is due to Alon and Linial:

Theorem 8.3.11 *[26] For a digraph $D = (V, A)$, if*

$$e(\Delta^-(D)\delta^+(D) + 1)(1 - 1/q)^{\delta^+(D)} < 1 \tag{8.3}$$

or if

$$e(\Delta^+(D)\delta^-(D) + 1)(1 - 1/q)^{\delta^-(D)} < 1, \tag{8.4}$$

then D contains a cycle of length 0 (mod q).

Proof: Since (8.4) transforms into (8.3) by replacing D by its converse, it suffices to prove that (8.3) implies that D has a cycle of length 0 modulo q.

For every vertex u, delete $d^+(u) - \delta^+(D)$ arcs with tail u and consider the resulting digraph $D' = (V, A')$. Assign to every vertex u of D' a colour $c(u)$, an integer in $\{0, 1, \ldots, q-1\}$, independently according to a uniform distribution. For each $u \in V$, let E_u denote the event that there is no $v \in V$ with $uv \in A'$ and $c(v) \equiv c(u)+1 \pmod q$. Clearly, $\text{Prob}(E_u) = (1-1/q)^{\delta^+(D)}$. It is not difficult to verify that each event E_u is mutually independent of all the events E_v except for those satisfying

$$N^+(u) \cap (v \cup N^+(v)) \neq \emptyset.$$

The number of such v's is at most $\Delta^-(D)\delta^+(D)$ and hence, by our assumption (8.3) and Lemma 8.3.9, $\text{Prob}(\cap_{u \in V} \overline{E}_u) > 0$. This means that there is a colouring c^* such that for every $u \in V$ there exists a $v \in V$ with $uv \in A'$ and $c^*(v) \equiv c^*(u) + 1 \pmod q$. Now it follows from Lemma 8.3.10 that D has a cycle of length 0 modulo q. □

The easy proof of the following corollary is left as Exercise 8.29.

Corollary 8.3.12 *Every k-regular digraph D with $k \geq 8$ contains an even cycle.* □

We have seen above that no constant k can guarantee that a digraph of out-degree at least k contains an even cycle. This leads to the following natural question (raised by Erdős, see [860]): what is the smallest integer $h(n)$

such that every digraph of order n and minimum out-degree $h(n)$ contains an even cycle? In order to prove an upper bound for $h(n)$ we need a result on hypergraph colouring. The following lemma is due to Beck [129][3]: Recall that a hypergraph $H = (V, \mathcal{E})$ is 2-colourable if there is a function $f : V \to \{0, 1\}$ such that, for every edge $E \in \mathcal{E}$, there exist a pair of vertices $x, y \in E$ such that $f(x) \neq f(y)$.

Lemma 8.3.13 *There exists an absolute constant d such that every m-uniform hypergraph with at most $\lfloor dm^{1/3}2^m \rfloor$ edges is 2-colourable.* □

Lemma 8.3.14 *[26] For every $n \geq 2$,*

$$h(n) \leq \log_2 n - \frac{1}{3} \log_2 \log_2 n + O(1).$$

Proof: Let $m \geq 2$ be an integer and let d be a constant satisfying Lemma 8.3.13. Suppose that

$$n = \lfloor dm^{1/3}2^m \rfloor \tag{8.5}$$

and let $D = (V, A)$ be a digraph of order n and $\delta^+(D) \geq m - 1$. Let H be the hypergraph on the set of vertices V, whose n edges are the sets $N^+[u] = N^+(u) \cup u$. Since every edge of H is of cardinality at least m, Lemma 8.3.13 implies that H is 2-colourable. This means that there exists a vertex colouring $c^* : V \to \{0, 1\}$ such that for every $u \in V$ there is $v \in N^+(v)$ with $c^*(v) \equiv c^*(u) + 1 \pmod 2$. Hence, by Lemma 8.3.10, D has an even cycle. Solving for m from (8.5) we obtain that

$$h(n) \leq m - 1 \leq \log_2 n - \frac{1}{3} \log_2 \log_2 n + O(1).$$

□

Clearly, if a digraph D contains cycles of length k and $k + 1$ for some k, then D has an even cycle. Deciding the existence of such cycles of consecutive length in a strong digraph is $\mathcal{NP}$-complete (see Exercise 8.33). Furthermore, it is easy to construct digraphs of arbitrary high vertex-strong connectivity with no such cycles (Exercise 8.34). It would be interesting to find non-trivial degree conditions (weaker than conditions implying pancyclicity, such as those in Section 8.1) which guarantee that a non-bipartite digraph has two cycles of consecutive lengths. See also Exercise 1.31 for another type of sufficient condition for the existence of two cycles of consecutive lengths.

8.4 Girth

Recall that the girth $g(D)$ of a digraph D is the length of a shortest cycle in D. The girth is an important parameter of a digraph and has been studied in a number of papers especially with respect to its extreme values.

[3] Radhakrishnan and Srinivasan [758] improved the bound of this lemma to $0.7 \cdot 2^m \sqrt{m/\ln m}$. Hence, the bound of Lemma 8.3.14 can be slightly improved.

Theorem 6.4.10 claims that if the minimum degree of every vertex in a strong digraph D is large enough, then the length of a longest cycle in D is large as well. Caccetta and Häggkvist [186] conjectured a somewhat similar result for girth (with obvious replacement of upper bound to a lower bound):

Conjecture 8.4.1 (Caccetta and Häggkvist) *[186] Every digraph of minimum out-degree k and order n has a cycle of length at most $\lceil n/k \rceil$.*

This conjecture is trivially true for $k = 1$; it was proved for $k = 2$ by Caccetta and Häggkvist [186], for $k = 3$ by Hamidoune [496] and for $k = 4$ and 5 by Hoang and Reed [530]. Hamidoune [495] proved that the conjecture is true for digraphs with transitive automorphism group. It follows from the next result that there is only a finite number, if any, of counterexamples to Conjecture 8.4.1. In particular, Conjecture 8.4.1 is proved for the case $n \geq 2k^2 - 3k + 1$.

Theorem 8.4.2 (Shen) *[815] For every digraph of order n and minimum out-degree k, $g(D) \leq \max\{\lceil n/k \rceil, 2k - 2\}$.* □

For an arbitrary integer $k \geq 1$, we have the following:

Theorem 8.4.3 (Chvátal and Szemerédi) *[223] There is a constant c such that every digraph of minimum out-degree $k \geq 1$ and order n contains a cycle of length at most $\lceil n/k \rceil + c$. Moreover, $c \leq 2500$.* □

Refinements of the proof in [223] were used by Nishimura [729] to show that $c \leq 304$ and by Shen [816] to prove that $c \leq 73$. For relatively small values of n/k, the following result of Chvátal and Szemerédi [223] is of interest.

Theorem 8.4.4 *Every digraph of minimum out-degree k and order n has a cycle of length at most $\lceil 2n/(k+1) \rceil$.*

Proof: By induction on $n \geq 2$. For $n = 2$ or 3 and $k \geq 1$, the digraph in question has either a 2-cycle or a 3-cycle and hence the claim holds. Let D be a digraph of order $n \geq 4$ and minimum out-degree $k \geq 1$. Since the size of D is at least kn, D contains a vertex v of in-degree at least k. If D has a 2-cycle, we are done. So, assume that D is an oriented graph. Let D' be the digraph obtained from D by deleting the vertices of $N^-[v] = N^-(v) \cup \{v\}$ and adding the new arc xy for every ordered pair x, y such that $xy \notin A(D)$, $y \in N^+(v)$ and x dominates an in-neighbour of v. Clearly, D' is of order at most $n-k-1$ and minimum out-degree at least k. By the induction hypothesis, D' contains a cycle C of length at most $2(n-k-1)/(k+1)$. Replacing each of the new arcs xy in C by the path $xuvy$, we obtain a closed walk C^* in D. If C has precisely s new arcs, then v appears on C^* exactly s times, and so C^* is the union of at least s cycles (see Exercise 1.5), whose total length is at most $2(n-k-1)/(k+1) + 2s$. Clearly, the shortest of these cycles has length at most $2n/(k+1)$. □

Searching for new approaches to the Caccetta-Häggkvist conjecture, Hoang and Reed [530] came up with the following conjecture that implies the Caccetta-Häggkvist conjecture (Exercise 8.35).

Conjecture 8.4.5 *Every digraph D of minimum out-degree k contains a sequence $C_1, C_2, \ldots, C_k$ of cycles such that $\cup_{i=1}^{j-1} C_i$ and C_j have at most one vertex in common.*

In the case of $k = 2$, the last conjecture was proved by Thomassen [862]. The conjecture was verified for tournaments by Havet, Thomassé and Yeo [510].

Theorem 8.4.6 *[862] Every digraph D of minimum out-degree 2 contains a pair of cycles with precisely one vertex in common.*

Proof: By induction on n, the order of D. If $n = 3$, the claim trivially holds, so assume that $n \geq 4$. Since the minimum out-degree in the terminal strong component of D is at least 2, we may assume that D is strong. Moreover, since $\delta^+(D) \geq 2$, D has a vertex x such that $D - x$ is strong (see Exercise 8.22). If $D\langle N^-(x)\rangle$ contains a cycle C, then the required pair of cycles consists of C and a cycle formed by a shortest path P from x to C and the arc from the terminal vertex of P to x. So, we may assume that $D\langle N^-(x)\rangle$ is acyclic, and, thus, $D\langle N^-(x)\rangle$ has a vertex y of in-degree 0.

If we delete all arcs with tail y and identify x and y, we obtain the digraph D' of order $n - 1$ and minimum out-degree at least 2. By the induction hypothesis, D' has a pair of cycles with precisely a vertex in common; these cycles correspond to cycles C_1 and C_2 in D. We may assume that C_1 and C_2 have yx in common for otherwise they have precisely a vertex in common. Since $D - x$ is strong, y is in a cycle C_3 of $D - x$. It is not difficult to see that $C_1 \cup C_2 \cup C_3$ contains a pair of cycles having precisely y in common. Indeed, if C_3 has only y in common with C_1 or C_2, then there is nothing to prove. If C_3 intersects with $C_1 \cup C_2$ at a vertex distinct from y, then let z be such a vertex with $C_3[y, z]$ being as short as possible (meaning that $C_3[y, z]$ has only y and z in common with $V(C_1) \cup V(C_2)$). Choose i such that z is in C_i, where $i = 1$ or 2. Then C_{3-i} and $C_i[z, y]C_3[y, z]$ is the required pair of cycles. □

The **density** of a digraph D is the ratio of its size and order (i.e., m/n). Clearly, high density of a strong digraph D guarantees that $g(D)$ is small. Thomassen (see [148]) asked to determine the least number $m(n, k)$ such that every strong digraph of order n and size at least $m(n, k)$ contains a cycle of length at most k. Bermond, Germa, Heydemann and Sotteau [148] solved this problem by proving the following:

Theorem 8.4.7 *Let D be a strong digraph of order n and let $k \geq 2$. Then*

$$|A(D)| \geq \frac{n^2 + (3 - 2k)n + k^2 - k}{2}$$

implies that $g(D) \leq k$. □

This theorem is best possible since there exist strong digraphs of order n and size $(n^2 + (3-2k)n + k^2 - k)/2 - 1$ with shortest cycle of length $k+1$ (Exercise 8.36).

In many questions on properties of (di)graphs, one may ask whether all (di)graphs satisfying a certain property must have cycles of length at most a constant. Perhaps the most famous such question is the problem regarding the chromatic number of an undirected graph: given $k \geq 3$ and $g \geq 3$, is there an undirected graph of chromatic number k and of girth at least g? This problem was resolved in the affirmative by Erdős [296] using probabilistic argument (a simplification of the original proof is given by Alon and Spencer [28]). Clearly, many digraphs of large vertex-strong connectivity are quite dense and, thus, of small girth. However, it is not difficult to construct digraphs of large vertex-strong connectivity and large girth. The 'vertex-strong connectivity' and 'girth' parts of the next result were proved by Ayoub and Frisch [54] (see Exercise 5.21) and Liu and Zhou [647] (see Exercise 8.37), respectively.

Proposition 8.4.8 *If $n = gs$, $g \geq 2$, then there exists an s-regular round digraph of order n which is s-strong and has girth g.* □

8.5 Short Cycles in Semicomplete Multipartite Digraphs

As we mentioned in Chapter 6 the hamiltonian cycle problem is $\mathcal{NP}$-complete for arbitrary digraphs and polynomial time solvable for certain families of digraphs including semicomplete multipartite digraphs. In this section we consider the existence of 'short' cycles in semicomplete multipartite digraphs. By **short cycles** in a semicomplete p-partite digraph we mean cycles of length at most p.

The cycle structure of semicomplete bipartite digraphs is quite well understood due to Theorem 6.6.4 and Exercises 8.16, 8.17. The cycle structure of semicomplete p-partite digraphs, $p \geq 3$, is less investigated especially for cycles of length more than p. In this section, we will consider results on cycles of length at most p. Most of the results on short cycles in semicomplete multipartite digraphs were actually obtained for multipartite tournaments. Therefore, we state them for multipartite tournaments. However, all of them can be immediately extended to semicomplete multipartite digraphs due to the following theorem of Volkmann.

Theorem 8.5.1 *[890] Let D be a strong semicomplete p-partite digraph of order n, $p, n \geq 2$, with a cycle C of length at least 3. Then D contains a strong orientation containing the cycle C, if and only if $D \neq \overleftrightarrow{K}_{1,n-1}$.* □

Interestingly enough the analogue of this theorem does not hold for longest paths, see Exercise 8.38 (some relaxation of the analogue still holds, see Exercise 8.39). It is often more convenient to work with the following easy corollary of this theorem.

Corollary 8.5.2 *[890] Every strong semicomplete p-partite digraph, $p \geq 3$, contains a spanning strong oriented subgraph.* □

One of the most interesting results on the topic is the following theorem of Guo and Volkmann.

Theorem 8.5.3 *[441] Let D be a strong p-partite tournament, $p \geq 3$, with partite sets $V_1, \ldots, V_p$. For each $i \in [p]$, there exists a vertex $v \in V_i$ belonging to an s-cycle of D for every $s \in \{3, 4, \ldots, p\}$.*

Proof: It suffices to prove that V_1 has a vertex v which is on an s-cycle of D for every $s \in \{3, 4, \ldots, p\}$. We proceed by induction on s.

We will first show that D has a 3-cycle through a vertex in V_1. Let $C = v_1v_2 \ldots v_kv_1$ be a shortest cycle through a vertex, say v_1, in V_1. Suppose that $k \geq 4$. By the minimality of k, $v_3 \in V_1$, since otherwise $v_3 \rightarrow v_1$ implying the 3-cycle $v_1v_2v_3v_1$ through a vertex in V_1, a contradiction. This means that $v_4 \notin V_1$; without loss of generality assume that $v_4 \in V_2$. Since $k \geq 4$ is minimal and $v_3 \in V_1$, we conclude that $v_4 \rightarrow v_1$, i.e., $k = 4$, and $v_2 \in V_2$. If there is a vertex $x \in V - (V_1 \cup V_2)$ which dominates a vertex of C and is dominated by a vertex in C, then there exists $i \in \{1, 2, 3, 4\}$ such that $v_{i+1} \rightarrow x \rightarrow v_i$ (indices modulo 4), which implies that there is a 3-cycle through v_1 or v_3, a contradiction.

This means that the set $V(D) - (V_1 \cup V_2)$ can be partitioned into sets S_1, S_2 such that $S_2 \rightarrow V(C) \rightarrow S_1$. Assume without loss of generality that $S_1 \neq \emptyset$. Since D is strong there is a path from S_1 to C. Let $P = x_1x_2 \ldots x_q$ be a shortest such path. Clearly, $q \geq 3$. If P has no vertex in S_2, then one of the vertices x_2, x_3 belongs to V_1 and the other to V_2 ($V - (S_1 \cup S_2) \subset V_1 \cup V_2$). By the minimality of P, $x_3 \rightarrow x_1$ implying that $x_1x_2x_3x_1$ is a 3-cycle containing a vertex in V_1, a contradiction. Therefore, P has a vertex in S_2. By the minimality of P and $S_2 \rightarrow C$, it follows that $x_{q-1} \in S_2$. If $q = 3$, then $v_1x_1x_2v_1$ is a 3-cycle, a contradiction. So, assume that $q \geq 4$. Since x_{q-2} cannot be in $S_1 \cup S_2$, $x_{q-2} \in V_1 \cup V_2$. If $x_{q-2} \in V_1$, we have $v_2 \rightarrow x_{q-2}$ implying that $x_{q-2}x_{q-1}v_2x_{q-2}$ is a 3-cycle, a contradiction. Finally, if $x_{q-2} \in V_2$, then $v_1x_{q-2}x_{q-1}v_1$ is a 3-cycle, a contradiction. We have shown that D has a 3-cycle containing a vertex in V_1.

Suppose now that $3 \leq s < p$ and some vertex u_1 of V_1 is contained in a k-cycle for every $k = 3, 4, \ldots, s$. Assume, on the other hand, that

$$\text{no vertex of } V_1 \text{ is in a } k\text{-cycle for any } k = 3, 4, \ldots, s, s+1. \tag{8.6}$$

Let $u_1u_2 \ldots u_su_1$ be an s-cycle of D and let S be the union of partite sets of D not represented in C. We claim that there is no vertex in S, which

dominates a vertex in C and is dominated by a vertex in C. Indeed, if such a vertex existed, one could insert it into C, a contradiction with (8.6). This means that S can be partitioned into sets S_1, S_2 such that $S_2 \rightarrow C \rightarrow S_1$. Assume without loss of generality that $S_1 \neq \emptyset$. Since D is strong there is a path from S_1 to C. Let $P = y_1 y_2 \dots y_q$ be a shortest such path. Clearly, $q \geq 3$.

Assume that P has a vertex of S_2. Clearly, $y_{q-1} \in S_2$ and no other vertex of P is in S_2. If $y_{q-2} \notin V_1$, then $y_{q-2}y_{q-1}C[u_3, u_1]y_{q-2}$ is an $(s+1)$-path containing u_1, a contradiction with (8.6). Hence, $y_{q-2} \in V_1$ and $u_2 \rightarrow y_{q-2}$. Now we see that $u_2 y_{q-2} y_{q-1} P[u_4, u_2]$ (or $u_1 u_2 y_{q-2} y_{q-1} u_1$, if $s = 3$) is an $(s+1)$-cycle containing u_1, a contradiction with (8.6). Thus, we conclude that P has no vertex of S_2.

Assume that P contains a vertex y_l of V_1. Let l be chosen such that $\{y_1, y_2, \dots, y_{l-1}\} \cap V_1 = \emptyset$. Assume that $q \leq s$. Due to the facts that every vertex of C dominates y_1, for every $k = 3, 4, \dots, s+1$, and $y_l \rightarrow \{y_1, y_2, ..., y_{l-2}\}$, there is a k-cycle C_k containing parts of C and P; C_k includes $y_l \in V_1$, a contradiction with (8.6). Therefore, $q \geq s+1$. Assume that $l \leq s+1$. Since $y_i \rightarrow y_1$, for every $i = 3, 4, \dots, s+1$, we obtain that $P[y_1, y_i]y_1$ is an i-cycle containing y_l, a contradiction with (8.6). Thus, we conclude that $l \geq s+2$. In the cycle $C' = P[y_1, y_l]y_1$, the vertex y_l dominates every vertex. Hence, for every $i = 3, 4, \dots, s+1$ we can construct an i-cycle using part of the vertices of C' including y_l, a contradiction with (8.6).

Thus, P has no vertex in V_1. Hence, u_1 dominates every vertex in P. If $q \geq k+1$, then $u_1 P[y_{q-k}, y_q]C[u_{k+1}, u_1]$ would be an $(s+1)$-cycle containing u_1, a contradiction with (8.6). Therefore, $q \leq k$. Since every vertex of C dominates y_1, $PC[u_{k+1}, u_{k-q+1}]y_1$ is an $(s+1)$-cycle containing u_1, a contradiction with (8.6).

Thus, the assumption (8.6) has resulted in a contradiction. This proves the theorem. □

This theorem generalizes several other results on multipartite tournaments and (ordinary) tournaments. Three of them are Moon's theorem on vertex pancyclic tournaments, Theorem 1.5.1, and the following extension of Theorem 1.5.1 by Gutin.

Corollary 8.5.4 *[453] Let D be a strong p-partite tournament, $p \geq 3$, such that one partite set of D consists of a single vertex v. Then for each $k \in \{3, 4, \dots, p\}$, D contains a k-cycle through v.* □

By Theorem 8.5.1, Corollary 8.5.4 can be extended to semicomplete p-partite digraphs, $p \geq 3$. Theorem 8.5.3 generalizes the following assertion, due to Bondy, which was actually the first non-trivial result on cycles in multipartite tournaments. Again, Corollary 8.5.5 can be extended to semicomplete p-partite digraphs, $p \geq 3$.

Corollary 8.5.5 *[167] A strong p-partite tournament contains an s-cycle for every $s \in \{3, 4, \dots, p\}$.* □

The assertion of this corollary is the best possible in the sense that for every $p \geq 3$ there exists a strong p-partite tournament with no cycle of length more than p. The following example is due to Bondy [167]. Let H be a p-partite tournament with partite sets $V_1 = \{v\}, V_2, \ldots, V_p$ such that $|V_i| \geq 2$ for each $2 \leq i \leq p$. If $V_2 \rightarrow v \rightarrow \cup_{j=3}^{p} V_j$ and $V_j \rightarrow V_i$ for $2 \leq i < j \leq p$, then H is strong but does not have a k-cycle for every $k > p$.

Another interesting generalization of Moon's theorem is due to Goddard and Oellermann.

Theorem 8.5.6 *[413] Every vertex of a strong p-partite tournament D belongs to a cycle that contains vertices from exactly t partite sets of D for each $t \in \{3, 4, \ldots, p\}$.* □

It is left as Exercise 8.40 to show that Theorem 8.5.3 is the best possible in the following sense: for every $p \geq 3$ there exists a strong p-partite tournament T such that some vertex v of T is not contained in a k-cycle for some $3 \leq k \leq p$. If one wishes to consider only cycles through a given vertex of a multipartite tournament, one perhaps should sacrifice the exactness. This is illustrated by the following result due to Guo, Pinkernell and Volkmann.

Theorem 8.5.7 *[438] If D is a strong p-partite tournament and v an arbitrary vertex of D, then v belongs to either a k-cycle or a $(k+1)$-cycle for every $k \in \{3, 4, \ldots, p\}$.* □

For regular multipartite tournaments Guo and Kwak proved the following much stronger result. Observe that the partite sets of a regular multipartite tournament are of the same cardinality.

Theorem 8.5.8 *[437] Let D be a regular p-partite tournament. If the cardinality of the partite sets of D is odd, then every arc of D is on a cycle that contains vertices from exactly k partite sets for each $k \in \{3, 4, \ldots, p\}$.* □

This theorem generalizes the corresponding result by Alspach [34] on regular tournaments. The next theorem is another generalization of Alspach's theorem.

Theorem 8.5.9 *[436] Let D be a regular p-partite tournament. If every arc of D is contained in a 3-cycle, then every arc of D is on a k-cycle for each $k \in \{3, 4, \ldots, p\}$.* □

Gutin and Rafiey [470] characterized strong p-partite tournaments in which a longest cycle is of length p and, thus, settled a problem in [888]. This characterization implies an $O(pn^3)$-time algorithm for checking whether the length of a longest cycle of a p-partite tournament on n vertices is p. Gutin, Rafiey and Yeo [471] characterized strong p-partite tournaments, which are not tournaments, that have a unique p-cycle. (Tournaments with unique Hamilton cycle were characterized by Douglas [273].) The characterization allowed the authors of [471] to enumerate such non-isomorphic p-partite tournaments for $p \geq 5$.

8.6 Exercises

8.1. **Non-pancyclic digraphs satisfying Meyniel's condition**. Prove that if $m > (n+1)/2$, then the digraph $D_{n,m}$ described after Theorem 8.1.3 satisfies Meyniel's condition for hamiltonicity but has no m-cycle.

8.2. **Pancyclic digraphs satisfying Woodall's condition for hamiltonicity**. Prove that if D satisfies the condition in Corollary 6.4.6, then either D is pancyclic, or n is even and $D = \overleftrightarrow{K}_{\frac{n}{2},\frac{n}{2}}$. Hint: use Theorem 8.1.3.

8.3. Prove the following result due to Overbeck-Larisch [736]. If a digraph $D = (V, A)$ satisfies $d(x) + d(y) \geq 2n + 1$ for every pair of non-adjacent vertices $x, y \in V$, then D is pancyclic. Hint: use Theorem 8.1.3.

8.4. (−) Prove that every regular tournament is strong.

8.5. (+) Prove Lemma 8.1.9. Hint: use a similar approach as that taken in the proof of Lemma 8.1.8.

8.6. (+) **Vertex-pancyclic quasi-transitive digraphs.** Prove part (b) of Theorem 8.1.10. Hint: use a similar approach as taken in the proof of (a) to reduce the problem to one for extended semicomplete digraphs and then apply Theorem 8.1.7.

8.7. Prove Lemma 8.1.13. Hint: consider a shortest cycle through v (which by the assumption has length at most k).

8.8. [517] Prove the following: let $C = v_1v_2 \ldots v_kv_1$ be a non-extendable cycle in a digraph $D = (V, A)$ on n vertices where $2 \leq k \leq n-1$ and let $u \in V - V(C)$. Then
(a) for every $1 \leq i \leq k$, D contains at most one of the arcs v_iu and uv_{i+1},
(b) $|(u, V(C))| + |(V(C), u)| \leq k$,
(c) for every $1 \leq i \leq k$, $|(v_i, V - V(C))| + |(V - V(C), v_{i+1})| \leq n - k$, and
(d) if $v_{i-1}u, uv_{i+1} \in A$, then for $1 \leq h \leq i-2$ or $i+1 \leq h \leq k$, D contains at most one of the arcs v_hv_i and v_iv_{h+1} and hence $|(v_i, V(C) - v_i)| + |(V(C) - v_i, v_i)| \leq k$.

8.9. **Cycle extendable regular tournaments.** Characterize these.

8.10. **Cycle extendable locally semicomplete digraphs.** Characterize cycle extendable locally semicomplete digraphs.

8.11. (+) **Weakly cycle extendable digraphs.** Call a digraph D **weakly cycle extendable** if every cycle C which is not a longest cycle of D is contained in some larger cycle C', i.e., $V(C) \subset V(C')$. For each of the following classes characterize weakly cycle extendable digraphs:
- Extended semicomplete digraphs
- Path-mergeable digraphs
- In-semicomplete digraphs

8.12. Prove Lemma 8.1.28.

8.13. Prove Lemma 8.1.29.

8.14. Prove Corollary 8.1.31.

8.15. Prove Corollary 8.1.32.

8.16. **(+)** A bipartite digraph $D = (V, A)$ on an even number n of vertices is **even (vertex-)pancyclic** if it has cycles of all lengths $4, 6, 8, \ldots, n$ (through every vertex $v \in V$). Prove the following theorem due to Zhang [929]:

Theorem 8.6.1 *A bipartite tournament D is even vertex-pancyclic if and only if D is hamiltonian and is not isomorphic to $\vec{C}_4[\overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}]$.*

8.17. Extend Theorem 8.6.1 to semicomplete bipartite digraphs (Gutin [456]).

8.18. **(−)** Let $1 \leq k \leq n$ be integers. Let $a_1, a_2, \ldots, a_k$ be a sequence of objects and let c be a colouring that assigns one of the colours $\{1, 2, \ldots, n\}$ to every object such that no colour is assigned to two objects. Prove that the probability of the event $c(a_1) < c(a_2) < \ldots < c(a_k)$ equals $1/k!$.

8.19. **(−)** Let M be an $n \times n$ matrix and let k be a natural number. Describe an algorithm that finds the kth power of M using only $O(\log k)$ multiplications of two $n \times n$ matrices.

8.20. Prove the first equality in the proof of Lemma 8.2.1.

8.21. Prove Lemma 8.2.4 using Lemma 8.2.3.

8.22. Let D be a strong digraph of minimum out-degree 2. Prove that D contains a vertex x such that $D - x$ is strong. Hint: consider D', a maximal strong proper subdigraph of D. Prove that D' contains all vertices of D but one.

8.23. **(−)** Prove that a digraph D is even if and only if, for every assignment of weights 0 and 1 to its arcs, D contains a cycle of even weight.

8.24. Let D be a k-weak-double-cycle for some odd k. Prove that D has an odd number of cycles and that every arc is in an even number of cycles. Hint: use the recursive definition of a k-weak-double-cycle.

8.25. Let D be a k-weak-double-cycle for some odd k. Prove that D has an even cycle. Hint: assume that all cycles in D are odd and use Exercise 8.24 to obtain a contradiction.

8.26. Prove that given an arc e in a digraph D it is $\mathcal{NP}$-complete to decide whether D has an odd cycle through e (even cycle through e, respectively) (Thomassen [860]).

8.27. **Digraphs for which all cycles have the same parity.** Show that there is a polynomial algorithm to decide if the length of all cycles of a given digraph have the same parity.

8.28. **(−)** Give a short direct proof that the problem to verify whether a digraph D has cycle of length 0 modulo p, where both D and p form an input, is $\mathcal{NP}$-complete.

8.29. Prove Corollary 8.3.12.

8.30. **(−)** Prove the following generalization of Lemma 8.3.10. Let $D = (V, A, w)$ be a weighted digraph and let $k \geq 2$ be an integer. If there is a vertex colouring $c^* : V \to \{0, 1, \ldots, k-1\}$ of D such that for every $u \in V$ there is a $v \in N^+(u)$ with $c^*(v) \equiv c^*(u) + w(u, v) \pmod{k}$, then D has a cycle of weight 0 (mod k) (Alon and Linial [26]).

8.31. **Cycles modulo k in weighted digraphs.** Using the result of the previous exercise and the method of proof of Theorem 8.3.11 prove the following generalization of Theorem 8.3.11: Let $D = (V, A, w)$ be a weighted digraph and let $k \geq 2$ be an integer. If either (8.3) or (8.4) holds, then D contains a cycle of weight 0 (mod k) (Alon and Linial [26]).

8.32. Prove that a 3-weak-double-cycle is (k, p)-odd for every pair k, p such that $1 \leq k < p,\ p \geq 3$ (Galluccio and Loebl [391]).

8.33. Prove that it is $\mathcal{NP}$-complete to decide whether a strong digraph has two cycles whose lengths differ by one. Hint: reduce the hamiltonian cycle problem to this problem.

8.34. Construct for every k an infinite family of k-strong digraphs such that no digraph in the family has two cycles whose lengths differ by one.

8.35. Prove that Conjecture 8.4.5 implies Conjecture 8.4.1.

8.36. For every $k \geq 2$, construct strong digraphs on n vertices such that the number of arcs is $\frac{n^2+(3-2k)n+k^2-k}{2} - 1$ and the shortest cycle has length $k + 1$.

8.37. (−) Prove that if $n = gs$, then the s-regular round digraph of order n is of girth g.

8.38. (+) For $p \geq 3$, construct an infinite family $\mathcal{F}_p$ of strong semicomplete p-partite digraphs such that every digraph D in $\mathcal{F}_p$ contains a hamiltonian path, yet, a longest path of any strong orientation of D has $n - 2$ vertices, where n is the order of D (Gutin, Tewes and Yeo [473]).

8.39. (++) Prove the following theorem. Let D be a strong semicomplete multipartite digraph of order n such that $D \neq \overleftrightarrow{K}_{n-1,1}$ and let l be the length of a longest path in D. Then D contains a strong spanning oriented subgraph with a path of length at least $l - 2$ (Gutin, Tewes and Yeo [473]).

8.40. For every $p \geq 3$ construct a strong p-partite tournament T such that some vertex v of T is not contained in a k-cycle for some $3 \leq k \leq p$.

9. Branchings

Recall that an in-tree (out-tree) is a oriented tree in which every vertex except one, called the **root**, has out-degree (in-degree) one. This chapter deals with spanning in- and out-trees in directed (multi)graphs. Some papers use the name arborescence for spanning in- and out-trees but we will use the name branching instead as this is also widely used and was used in the original paper by Edmonds on arc-disjoint out-branchings (see Section 9.3 below). Thus an **out-branching (in-branching)** of D is a spanning out-tree (in-tree) in D.

Branchings play an important role in the theory of directed graphs and have several important applications, which is why we have devoted a separate chapter to branchings. We start with Tutte's Matrix Tree theorem which gives a formula for the number of out-branchings with a given root in a directed multigraph. Then we discuss the minimum cost branching problem, a directed analogue of the minimum spanning tree problem. After this we consider arc-disjoint branchings and out-branchings with bounds on the out-degrees. Then we move on to arc-disjoint in- and out-branchings and out-trees with extreme numbers of leaves (many or few). Finally we give some results on the source location problem and discuss a number of miscellaneous topics.

We will use the following notation: B_s^+, B_s^-, B^+ and B^- denote respectively an out-branching rooted at s, an in-branching rooted at s, an out-branching with no root specified and an in-branching with no root specified. Similarly T_s^+, T_s^-, T^+, T^- denote respectively an out-tree rooted at s, an in-tree rooted at s, an out-tree with no root specified and an in-tree with no root specified.

9.1 Tutte's Matrix Tree Theorem

Counting spanning trees in graphs is a fundamental problem in graph theory and dates back to Cayley's famous formula [193] from 1889 stating that the number of spanning trees in K_n is n^{n-2}. Actually, this formula follows from the so-called Matrix Tree Theorem for graphs (Corollary 9.1.3) which is implicit in the work of Kirchoff from 1847 [596]. In 1948 Tutte [876] proved a generalization of this theorem to directed multigraphs (Theorem 9.1.2). He proved that the number of out-branchings rooted at the same vertex r in a

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 339
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_9,

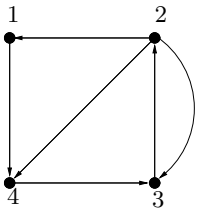
digraph D can be found efficiently by calculating the determinant of a certain matrix derived from D. The purpose of this section is to derive this result.

Recall that for a directed multigraph D we denote by $\mu_D(i,j)$ the number of arcs from i to j in D.

The so-called **Kirchoff matrix** $K = K(D)$ of a directed multigraph D on n vertices is defined as follows, where we assume that the vertices are numbered $1, 2, \ldots, n$:

$$K_{ij} = \begin{cases} d^-(i) & \text{if } i = j \text{ and} \\ -\mu_D(i,j) & \text{if } i \neq j. \end{cases} \tag{9.1}$$

Notice that the sum of the entries of any column of K is zero by the definition above. Figure 9.1 shows a digraph and its Kirchoff matrix.



$$K = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & -1 & -1 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Figure 9.1 A digraph D and its Kirchoff matrix $K = K(D)$.

For $i \in [n]$ we denote by $K_{\bar{i}}(D)$ the matrix obtained from $K(D)$ by deleting the ith row and the ith column.

Lemma 9.1.1 *Let H be a directed multigraph with $\Delta^-(H) = 1$ and let i be a vertex of H. Then H has at most one out-branching rooted at i. Moreover, $\det(K_{\bar{i}}(H)) \in \{0,1\}$ and $\det(K_{\bar{i}}(H)) = 1$ if and only if H has an out-branching rooted in i.*

Proof: Suppose first that H contains an out-branching B_i^+ rooted at i. Then every vertex except i has its only incoming arc included in B_i^+, showing that H has no other out-branching rooted at i. Since application of the same permutation to the rows and the columns to a matrix does not change the value of its determinant, we may assume that $i = 1$ and that the other vertices are labelled according to a breadth-first order from vertex 1. With respect to this ordering $K_{\bar{i}}(H)$ is an upper triangular matrix, all of whose diagonal elements are one, implying that $\det(K_{\bar{i}}(H)) = 1$.

Now suppose that H has no out-branching rooted at i. If some vertex $j \neq i$ has $d^-(j) = 0$, then the jth column of $K_{\bar{i}}$ consists of zeros only, implying that $\det(K_{\bar{i}}(H)) = 0$. So we may assume that $d^-(j) = 1$ for every $j \neq i$. Now the fact that there is no out-branching from i implies that H has a cycle C which does not contain vertex i. It is easy to show that the columns corresponding to the vertices of C are linearly dependent and thus $\det(K_{\bar{i}}(H)) = 0$. □

Theorem 9.1.2 (Tutte's Matrix Tree Theorem) *[876] For every directed multigraph D with Kirchoff matrix $K(D)$ defined as in (9.1), the number of out-branchings rooted at vertex i equals* $\det(K_{\bar{i}}(D))$.

Proof: Recall the following basic identity for the determinant of a matrix consisting of columns $c_1, c_2, \ldots, c_n$ each on n elements:

$$\det(c_1, \ldots, (c_i + c_i'), \ldots, c_n) = \det(c_1, \ldots, c_i, \ldots, c_n) + \det(c_1, \ldots, c_i', \ldots, c_n).$$

Since the only entries in $K(D)$ which are greater than zero occur in the diagonal elements and each column sums to zero, we may decompose $K(D)$ into $s = \Pi_{i=1}^{n} K_{ii}$ $n \times n$ matrices $\{K(H_i)\}_{i\in[s]}$ by writing each column i as the sum of K_{ii} columns which correspond in a one-to-one fashion to the arcs entering vertex i (see Figure 9.2).

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Figure 9.2 The decomposition of $K(D)$ into its four components. Here D is the digraph in Figure 9.1.

We may assume w.l.o.g. that $i = 1$. We will now show how to express $\det(K_{\bar{1}}(D))$ in terms of the simpler matrices $K(H_i)$, $i \in [s]$ above. Let $\hat{D}$ be obtained from D by deleting all arcs into vertex 1. Then we have

$$\det(K_{\bar{1}}(\hat{D})) = \sum_{j_2=1}^{d^-(2)} \det(K_{\bar{1}}(\hat{D}_{j_2})),$$

where $\hat{D}_{j_2}$ is obtained from $\hat{D}$ by deleting all arcs entering vertex 2 apart from one (the j_2th arc). Continuing this way we get

$$\det(K_{\bar{1}}(\hat{D})) = \sum_{j_2=1}^{d^-(2)} \cdots \sum_{j_n=1}^{d^-(n)} \det(K_{\bar{1}}(\hat{D}_{j_2 \ldots j_n})).$$

Now the theorem follows from Lemma 9.1.1 applied to each term above and the fact that $K_{\bar{1}}(\hat{D}) = K_{\bar{1}}(D)$. □

Now we can prove Kirchoff's famous formula for the number of spanning trees in an undirected graph G.

Corollary 9.1.3 (Kirchoff's Matrix Tree Theorem) *[596] The number of spanning trees in an undirected graph G on n vertices is equal to any one of the numbers* $\det(K(\overleftrightarrow{G_i}))$, $i \in [n]$.

Proof: Fix an arbitrary vertex r in G and observe that every spanning tree T in G corresponds to a unique out-branching B_r^+ in $\overleftrightarrow{G}$. Now the claim follows from Theorem 9.1.2. □

Since the determinant of a matrix can be calculated efficiently we obtain the following (see, e.g., [405, page 53]).

Corollary 9.1.4 *There is an $\mathcal{O}(n^3)$ algorithm for finding the number of out-branchings rooted at a given of a directed multigraph on n vertices.* □

If we actually wanted to list all out-branchings in a digraph D, we clearly have to spend time at least proportional to the number of such branchings in D. In [582] Kapoor and Ramesh give an $\mathcal{O}(Nn + n^3)$ algorithm for listing all out-branchings in a directed multigraph on n vertices and N out-branchings. The algorithm is based on generating one out-branching from another by a series of arc swaps.

9.2 Optimum Branchings

Given a directed multigraph $D = (V, A)$ a special vertex s and a non-negative cost function w on the arcs. What is the minimum cost (measured as the sums of the arc costs) of an out-branching B_s^+ rooted at s in D? This problem, which is a natural generalization of the minimum spanning tree problem for undirected graphs (Exercise 9.6), is called the MINIMUM COST BRANCHING PROBLEM. The problem arises naturally in applications where one is seeking a minimum cost subnetwork which allows communication from a given source to all other vertices in the network (see the discussion at the end of the section).

It is easy to find a minimum spanning tree in an undirected graph. The greedy approach works as follows: order the edges according to their weights in increasing order $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$. Start from $T = \emptyset$ and go through $\mathcal{E}$ while always adding the next edge to T if it can be added without creating a cycle. This is the so-called Kruskal algorithm (see, e.g., [232]). It is not difficult to construct examples which show that using a similar greedy approach to find a minimum cost out-branching in a directed multigraph may be incorrect (Exercise 9.2).

The minimum cost branching problem was first shown to be polynomially solvable by Edmonds [283]. Later Fulkerson [366] gave a two-phase greedy algorithm which solves the problem very elegantly. The fastest algorithm for the problem is due to Tarjan [844]. Tarjan's algorithm solves the problem in time

$O(m \log n)$, that is, with the same time complexity as Kruskal's algorithm for undirected graphs [231]. The purpose of this section is to describe two different algorithms for finding minimum cost out-branching in a weighted directed multigraph. First we show how to solve the problem using matroids and then we give a simple direct algorithm based on Edmonds' original algorithm.

9.2.1 Matroid Intersection Formulation

To illustrate the generality of matroids, let us show how to formulate the minimum cost branching problem as a weighted matroid intersection problem. We refer to Section 18.8 for relevant definitions on matroids.

Let $D = (V, A)$ be a directed multigraph and let $r \in V$ be a vertex which can reach all other vertices by directed paths. We define $M_1 = (A, \mathcal{I}_1)$ and $M_2 = (A, \mathcal{I}_2)$ as follows (here $\mathcal{I}_1, \mathcal{I}_2 \subseteq 2^A$):

- $A' \in \mathcal{I}_1$ if and only if no two arcs in A' have a common head and no arc has head r,
- $A'' \in \mathcal{I}_2$ if and only if $UG(D\langle A''\rangle)$ has no cycle.

It follows from the definition of M_2 that M_2 is the circuit matroid of $UG(D)$ (see Section 18.8). It is easy to show that M_1 satisfies the axioms (I1)-(I3) and hence is a matroid. In particular, all maximal members of $\mathcal{I}_1$ have the same size $n-1$ (by our assumption, every vertex in $V - r$ has at least one in-neighbour) and thus the rank of M_1 is $n-1$.

Since r can reach all other vertices, $UG(D)$ is connected and hence the rank of M_2 is also $n-1$. We claim that every common base of M_1 and M_2 is an out-branching with root r. This follows easily from the definition of an out-branching and the fact that any common base corresponds to a spanning tree in $UG(D)$, since M_2 has rank $n-1$.

Thus we can find an out-branching with root r by applying the algorithm for matroid intersection of Theorem 18.8.11 to the pair M_1, M_2. Of course such an out-branching can be found much easier by using, e.g., DFS starting from r. However, the point is that using the algorithm for weighted matroid intersection, we can find a minimum cost out-branching B_r^+ in D. It is easy to see that the required oracles for testing independence in M_1 and M_2 can be implemented very efficiently (Exercise 9.3). In fact (and much more importantly in the light of the existence of other and more efficient algorithms for minimum cost branchings), using matroid intersection algorithms we can even find a minimum cost subdigraph which has k out-branchings with a specified root s in a directed multigraph with non-negative weights on the arcs (Exercise 9.4). Furthermore, in Exercise 9.5, the reader is asked to show that one can also solve the following problem, using matroid intersection: Given directed multigraphs $D = (V, A)$ and $D' = (V, A')$ on the same vertices, a cost function c on A', a natural number k and a vertex $s \in V$. Find a minimum cost set of arcs $A^* \subseteq A'$ such that the directed multigraph $D^* = (V, A \cup A^*)$ has k arc-disjoint out-branchings rooted at s.

Clearly, the minimum cost branching problem corresponds to the case when $A = \emptyset$. Hence, using matroid intersection formulations, one can in fact solve problems which are much more general than the minimum cost branching problem.

9.2.2 A Simple Algorithm for Finding a Minimum Cost Out-Branching

Below we will often call a minimum cost out-branching an **optimum** out-branching. Let $D = (V, A)$ be a directed multigraph with a designated root $r \in V$ and c a non-negative cost vector on A. Denote by y_v, $v \in V - r$, the minimum cost of an arc entering v. The following easy observation is the key to the algorithm below.

Lemma 9.2.1 *Let c' be the cost function on A defined by $c'(uv) = c(uv) - y_v$. Then B_r^+ is an optimum branching with respect to c if and only if it is optimum with respect to c'.*

Proof: Since every vertex except r has precisely one arc entering it in any out-branching, $c(B_r^+) = c'(B_r^+) + \sum_{v \in V-r} y_v$ holds for an arbitrary out-branching B_r^+ and the claim follows. □

For a given directed multigraph D and weight function c, let F^* be a subdigraph of D obtained by taking a minimum cost arc entering each vertex except r, that is, $d^-_{F^*}(v) = 1$ for $v \neq r$. Note that the cost of F^* is zero with respect to c' and hence the following holds, by Lemma 9.2.1.

Lemma 9.2.2 *If F^* is an out-branching, then it is optimum.* □

The following result is due to Karp.

Lemma 9.2.3 *[584] There exists an optimum out-branching with root r which contains all but one arc of every cycle C in F^*.*

Proof: Let B_r^+ be an optimum out-branching which contains the maximum number of arcs from F^*. If F^* is itself a branching, then, by Lemma 9.2.2, we have $B_r^+ = F^*$, so assume that C is a cycle in F^* and suppose $A(C) - A(B_r^+) = \{u_1v_1, u_2v_2, \ldots, u_kv_k\}$ has at least 2 arcs and occurring in that order on C. Consider an arbitrary vertex v_i, $i \in [k]$, and denote by $a(v_i)$ the arc entering v_i in B_r^+. By the choice of B_r^+, $H_i = B_r^+ + u_iv_i - a(v_i)$ is not an out-branching. This implies that H_i contains a cycle which consists of the arc u_iv_i and a path P_i which starts in v_i and ends in u_i. Consider the last arc xy of P_i which does not belong to C. As H_i contains all the arcs of $C[v_{i-1}, u_i]$ and every vertex of C has in-degree one in H_i it follows that $y = v_{i-1}$ (indices are taken modulo k). Thus we have shown that H_i and hence B_r^+ contains a (v_i, v_{i-1})-path. However, this holds for every $i \in [k]$ and so B_r^+ contains a

directed cycle[1], a contradiction. Hence we have shown that B_r^+ contains all but one arc of C. □

When we contract the cycle C below to get the weighted directed multigraph D/C, the arcs incident to v_C inherit the costs from the original arcs between C and $V - C$.

Lemma 9.2.4 *If C is a cycle in F^* and W_r^+ is an optimum out-branching in D/C (the directed multigraph obtained by contracting C to a vertex v_C), then we can obtain an optimum branching B_r^+ in D by replacing v_C by C minus one arc.*

Proof: Let xv_C be the unique arc of W_r^+ entering v_C and let $y \in V(C)$ be chosen so that $xy \in A$ and has the same cost as xv_C in D/C. Clearly we can extend W_r^+ to an out-branching B_r^+ of D by blowing up C again and deleting the unique arc of C which enters y (arcs leaving v_C in W_r^+ are replaced by corresponding arcs starting in vertices from C). By Lemma 9.2.3, there exists an optimum out-branching $\hat{B}_r^+$ containing all but one arc of C and contracting C will transform $\hat{B}_r^+$ into an out-branching $\tilde{B}_r^+$ in D/C. Now, by Lemma 9.2.1, it follows from the fact that W_r^+ is an optimum out-branching in D/C and C has cost zero w.r.t. c' that $c'(\hat{B}_r^+) \geq c'(B_r^+)$, implying that B_r^+ is an optimum out-branching. □

Theorem 9.2.5 *[283] There is a polynomial algorithm for the minimum cost out-branching problem.*

Proof: We may assume that the root r can reach every other vertex, as otherwise no branching exists. The algorithm is very simple. First construct F^* and search for a cycle in it. If F^* is acyclic, it is the desired branching. If F^* contains a cycle C, let $D' = D/C$ and solve the problem recursively in D'. Finally convert the optimum out-branching in D' to an optimum out-branching of D as described in the proof of Lemma 9.2.4. This algorithm can easily be implemented as an $O(n(n+m))$ algorithm. □

9.3 Arc-Disjoint Branchings

This section is devoted to a very important result due to Edmonds [285]. The result can be viewed as just a fairly simple generalization of Menger's theorem. However, as will be clear from the next subsections, it has many important consequences.

[1] Note that when $k = 1$ we do not get the contradiction since P_1 is simply $C[v_1, u_1]$.

Theorem 9.3.1 (Edmonds' branching theorem) *[285] A directed multigraph $D = (V, A)$ with a special vertex z has k arc-disjoint out-branchings rooted at z if and only[2] if*

$$d^-(X) \geq k \ \ \forall \ \emptyset \neq X \subseteq V - z. \tag{9.2}$$

Proof: We give a short proof due to Lovász [654]. The necessity is clear, so we concentrate on sufficiency. The idea is to grow an out-tree T_z^+ from z in such a way that the following condition is satisfied:

$$d^-_{D-A(T_z^+)}(U) \geq k-1 \quad \text{for all } \emptyset \neq U \subseteq V - z. \tag{9.3}$$

If we can keep on growing T_z^+ until it becomes spanning while always preserving (9.3), then the theorem follows by induction on k. To show that we can do this, it suffices to prove that we can add one more arc at a time to T_z^+ until it is spanning. Let us call a set $X \subseteq V - z$ **problematic** if $d^-_{D-A(T_z^+)}(X) = k-1$. It follows from the submodularity of $d^-_{D-A(T_z^+)}$ (recall Corollary 5.1.2) that if X, Y are problematic and $X \cap Y \neq \emptyset$, then so are $X \cap Y, X \cup Y$. Observe also that if X is problematic, then $X \cap V(T_z^+) \neq \emptyset$, because X has in-degree at least k in D. If all problematic sets are contained in $V(T_z^+)$, then let $W = V$. Otherwise let W be an inclusions-wise minimal problematic set which is not contained in $V(T_z^+)$.

We claim that there exists an arc uv in D such that $u \in V(T_z^+) \cap W$ and $v \in W - V(T_z^+)$. Indeed if this was not the case, then $W \neq V$ and every arc that enters $W - V(T_z^+)$ also enters W. Hence we would have

$$d^-_D(W - V(T_z^+)) = d^-_{D-A(T_z^+)}(W - V(T_z^+)) \leq d^-_{D-A(T_z^+)}(W) \leq k-1, \tag{9.4}$$

contradicting the assumption of the theorem.

The arc uv cannot enter a problematic set X, since that would contradict the definition of W (recall that $u \in W$). Hence we can add the arc uv to T_z^+ without violating (9.3) and now the claim follows by induction. □

Corollary 9.3.2 *There exists a polynomial algorithm for finding k arc-disjoint out-branchings from a given root s in a directed multigraph which satisfies (9.2).*

Proof: The proof above can be turned into a polynomial algorithm which, given a directed multigraph $D = (V, A)$ a vertex $z \in V$ and a natural number k, either finds k arc-disjoint out-branchings from k, or a set $X \subseteq V - z$ with out-degree less than k (Exercise 9.7). □

[2] By Menger's theorem (Theorem 5.4.1), (9.2) is equivalent to the existence of k arc-disjoint paths from z to every other vertex of D.

The following possible generalization naturally emerges. In addition to z, we are given a subset $W \subseteq V - z$ so that $d^-(X) \geq k$ for every subset $X \subseteq V - z, X \cap W \neq \emptyset$ (by Menger's theorem this is equivalent to saying that there are k arc-disjoint (z,t)-paths for every $t \in W$). Is it true that there are k arc-disjoint out-trees rooted at z so that each contains every element of W? The answer is yes if $W = V - z$ (by Edmonds' theorem) or if $|W| = 1$ (by Menger's theorem). However, Lovász [652] found the example in Figure 9.3 which shows that such a statement is not true in general. This example can be generalized to directed multigraphs with arbitrarily many vertices (Exercise 9.10).

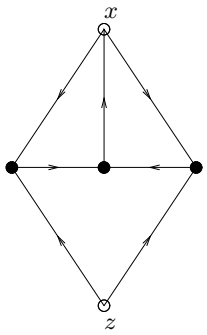


Figure 9.3 A digraph with $\lambda(z,t) \geq 2$, $t \in W$ which has no two arc-disjoint out-trees rooted at z and both containing every element of T. Here W consists of the three black vertices ([652, Figure 1]).

Observe that in Figure 9.3 $d^-(x) = 1 < 2 = d^+(x)$ holds for the only vertex x not in T and recall that the desired number of arc-disjoint out-trees above was two. Bang-Jensen, Frank and Jackson proved that if $\lambda(z,x) \geq k$ holds for those vertices $x \in V(D)$ for which $d^+(x) > d^-(x)$ (that is, the value of k is restricted by the local arc-connectivities from z to these vertices), then a generalization is indeed possible.

Theorem 9.3.3 *[78] Let $D = (V,A)$ be a directed multigraph with a special vertex z and let $T' := \{x \in V - z : d^-(x) < d^+(x)\}$. If $\lambda(z,x) \geq k (\geq 1)$ for every $x \in T'$, then there is a family $\mathcal{F}$ of k arc-disjoint out-trees rooted at z so that every vertex $x \in V$ belongs to at least $r(x) := \min(k, \lambda(z,x))$ members of $\mathcal{F}$.* □

Clearly, if $\lambda(z,x) \geq k$ holds for every $x \in V$ in Theorem 9.3.3, then we are back at Edmonds' theorem. Another special case is also worth mentioning. Call a directed multigraph $D = (V,A)$ with root z a **preflow directed multigraph** if $d^-(x) \geq d^+(x)$ holds for every $x \in V - z$. (The name arises from the max-flow algorithms of Karzanov [586] and Goldberg and Tarjan

[415], see also the definition of a preflow in Chapter 4.) The following corollary of Theorem 9.3.3 may be considered as a generalization of Theorem 4.3.1.

Corollary 9.3.4 *[78] In a preflow directed multigraph $D = (V, A)$ for any integer $k(\geq 1)$ there is a family $\mathcal{F}$ of k arc-disjoint out-trees with root z so that every vertex x belongs to $\min(k, \lambda(z, x; D))$ members of $\mathcal{F}$. In particular, if $k := \max(\lambda_D(z, x) : x \in V - z)$, then every x belongs to $\lambda_D(z, x)$ members of $\mathcal{F}$.* □

Aharoni and Thomassen have shown that Edmonds' branching theorem cannot be generalized to infinite directed multigraphs [9].

9.4 Implications of Edmonds' Branching Theorem

Below we give a number of consequences of Theorem 9.3.1 (for yet another consequence see Theorem 10.7.3). The first result, due to Even, may be viewed as a generalization of Menger's theorem for global arc-strong connectivity.

Corollary 9.4.1 *[306, Theorem 6.10] Let $D = (V, A)$ be a k-arc-strong directed multigraph and let x, y be arbitrary distinct vertices of V. Then for every $0 \leq r \leq k$ there exist paths $P_1, P_2, \ldots, P_k$ in D which are arc-disjoint and such that the first r paths are (x, y)-paths and the last $k - r$ paths are (y, x)-paths.*

Proof: Let $[D, x, y]$ be as described above. Add a new vertex s and join it to x by r parallel arcs of the form sx and to y by $k - r$ parallel arcs of the form sy. Let D' denote the new directed multigraph. We claim that D' satisfies (9.2). To see this let $X \subseteq V$ be arbitrary. If $X \neq V$, then we have $d^-_{D'}(X) \geq d^-_D(X) \geq k$, since D is k-arc-strong. If $X = V$, we have $d^-_{D'}(V) = d^+_{D'}(s) = k$. It follows from Theorem 9.3.1 that D' contains k arc-disjoint out-branchings all rooted at s. By the construction of D', when we restrict to D, these branchings must consist of r out-branchings rooted at x and $k - r$ out-branchings rooted at y. Take the r (x, y)-paths from those rooted at x and the $k - r$ (y, x)-paths from those rooted at y and we obtain the desired paths. □

The next result, due to Nash-Williams, gives a sufficient condition for the existence of k edge-disjoint spanning trees in an undirected graph. This condition is the best possible in terms of the edge-connectivity (see the remark after Theorem 9.4.3) and hence we see that for an undirected graph we may need twice the obvious edge-connectivity requirement to guarantee k edge-disjoint trees. This contrasts with the case for directed graphs where k-arc-strong connectivity suffices by Edmonds' theorem.

Theorem 9.4.2 *[717] Every $2k$-edge-connected undirected graph contains k edge-disjoint spanning trees.*

Proof: Let $G = (V, E)$ be a $2k$-edge-connected undirected graph. By Nash-Williams' orientation theorem (Theorem 11.5.3), G has a k-arc-strong orientation $D = (V, A)$. Let $z \in V$ be arbitrary and note that $d^-(X) \geq k$ holds for each subset $X \subseteq V - z$ of vertices. Hence by Theorem 9.3.1, D contains k arc-disjoint out-branchings rooted at z. Suppressing the orientation of all arcs on the branchings we obtain k edge-disjoint trees in $G = UG(D)$. □

The following characterization, due to Tutte, of undirected graphs which have k edge-disjoint spanning trees can also be derived from Edmonds' branching theorem and Theorem 11.7.6 (see Exercise 9.19). See also Exercise 11.59 for a simpler orientation result which still implies Theorem 9.4.3.

Theorem 9.4.3 *[879] An undirected graph $G = (V, E)$ has k edge-disjoint spanning trees if and only if*

$$\sum_{1 \leq i < j \leq p} e(V_i, V_j) \geq k(p-1), \tag{9.5}$$

holds for every partition $V_1, V_2, \ldots, V_p$ of V. Here $e(V_i, V_j)$ denotes the number of edges with one end in V_i and the other in V_j. □

It is easy to derive Theorem 9.4.2 from Theorem 9.4.3. Furthermore, we can use Theorem 9.4.3 to show that the condition in Theorem 9.4.2 is best possible in terms of the edge-connectivity. Let G_k be the graph obtained from the complete graph on $2k+2$ vertices by removing the edges of a hamiltonian cycle. Then it is easy to show that G_k is $(2k-1)$-edge-connected and using Theorem 9.4.3 on the partition corresponding to one vertex per set in the partition we can see that G_k has no k edge-disjoint spanning trees (in fact this partition has precisely one arc less than the required number). In order to get an example with arbitrarily many vertices and no k edge-disjoint trees for each k we let H be an arbitrary $2k$-edge-connected graph and let H_k be the graph consisting of $2k+2$ copies $H_1, H_2, \ldots, H_{2k+2}$ of H and with one edge between H_i and H_j just if the corresponding vertices v_i, v_j are adjacent in G_k (where we have assumed that the vertices of G_k are labelled $v_1, v_2, \ldots, v_{2k+2}$ and H_i corresponds to v_i for $i \in [2k+2]$). It is not difficult to prove that H_k is $(2k-1)$-edge-connected and the partition corresponding to the $2k+2$ copies of H shows that H_k has no k edge-disjoint spanning trees. Note also that G_k above is $(2k-1)$-edge-connected and $(2k-1)$-regular. Furthermore, a simple counting argument shows that all except finitely many $(2k-1)$-edge-connected and $(2k-1)$-regular graphs have no k edge-disjoint spanning trees (simply because they do not have enough edges).

In some applications (e.g., when a number of tasks have to be distributed to different units who can cover part of the jobs or demands) one is interested in covering all edges (arcs) of an undirected (a directed) graph by forests (in- or out-trees).

Theorem 9.4.4 *[718] Let $G = (V, E)$ be an undirected graph. Then E can be covered by k forests if and only if*

$$|E(G\langle X\rangle)| \leq k(|X|-1) \qquad \text{for all } X \subseteq V. \tag{9.6}$$

Proof: Since no forest can use more than $|X|-1$ edges with both ends inside any set X, we see that the condition (9.6) is necessary. To prove sufficiency we use Theorem 9.3.1 and the following result which follows easily from Theorem 11.7.3:

Proposition 9.4.5 *A graph $H = (V, E)$ has an orientation $D = (V, A)$ such that $d^-_D(v) \leq k$ for every vertex $v \in V$ if and only if*

$$|E(G\langle X\rangle)| \leq k|X| \qquad \text{for all } X \subseteq V. \qquad \square$$

Suppose now that $G = (V, E)$ satisfies (9.6). By Proposition 9.4.5, G has an orientation D such that $d^-_D(v) \leq k$ for every vertex $v \in V$. Add a new vertex s to D and add $k - d^-_D(v)$ arcs from s to v for each $v \in V$. Denote the new directed multigraph by D'. We claim that

$$d^-_{D'}(X) \geq k \qquad \text{for all } X \subseteq V. \tag{9.7}$$

This follows from the fact that for every $X \subseteq V$ we have

$$\begin{aligned} d^-_{D'}(X) &= \sum_{v \in X} d^-_D(v) - |E(G\langle X\rangle)| \\ &= k|X| - |E(G\langle X\rangle)| \\ &\geq k|X| - k(|X|-1) = k. \end{aligned}$$

By Theorem 9.3.1, D' has k arc-disjoint out-branchings rooted at s. These branchings must use all arcs of D since every vertex of V has in-degree one in each of these branchings and we have only added $k - d^-_D(v)$ arcs from s to v. Now delete the vertex s from each of the branchings and suppress the orientations of all arcs. The resulting k forests cover E. $\square$

The last part of the proof above also implies the sufficiency part of the following theorem. The necessity of (9.8) follows from the fact that no vertex of an out-branching has in-degree bigger than one. The necessity of (9.9) is seen as in the proof above.

Theorem 9.4.6 *[336] The arc set of a directed multigraph $D = (V, A)$ can be covered by k out-trees if and only if*

$$d^-(v) \leq k \qquad \text{for all } v \in V \text{ and} \tag{9.8}$$

$$|A(D\langle X\rangle)| \leq k(|X|-1) \qquad \text{for all } X \subseteq V. \tag{9.9}$$

$\square$

9.5 Out-Branchings with Degree Bounds

Finding a spanning tree T with restrictions on the maximum degree of T in a graph is a well-known problem which has many practical applications, e.g., in communications, design of reliable networks, etc. Such problems have been studied extensively in both the mathematical and the computer science literature (see, e.g., the references in [117]). If we wish to find a spanning tree where the maximum degree is at most some given integer, then the problem is NP-hard[3], but Fürer and Raghavachari [369] showed that, in polynomial time, one can find in a given connected graph G a spanning tree of G whose maximum degree is at most $t(G)+1$, where $t(G)$ is the least number k such that G has a spanning tree T with $\Delta(T)=k$. Czumaj and Strothmann [236] showed that if the input graph G is k-connected and has maximum degree at most $k(r-2)+2$, then one can find, in polynomial time, a spanning tree T of G such that $\Delta(T)\leq r$, where $\Delta(T)$ denotes the maximum degree of any vertex in T. They also showed that for the special case of 2-connected graphs one can obtain a stronger result: Every 2-connected graph G contains a spanning tree T with the property that

$$d_T(v)\leq \frac{d_G(v)+3}{2} \text{ for every vertex } v. \tag{9.10}$$

Bang-Jensen, Thomassé and Yeo [117] studied analogous problems for out-branchings in directed multigraphs. They made the following conjecture.

Conjecture 9.5.1 *[117] Let D be a k-arc-strong directed multigraph. For every vertex $s\in V$, there exists an out-branching B_s^+ such that $d^+_{B_s^+}(x)\leq \frac{d^+_D(x)}{k}+1$ for all vertices x of D.*

This would be best possible as there are k-arc-strong k-regular directed multigraphs with no hamiltonian path (Exercise 9.11). Now let G be a $2k$-edge-connected graph. By Exercise 11.32, G has a k-arc-strong balanced[4] orientation D. Hence, Conjecture 9.5.1 would imply the existence of a spanning out-branching B_s^+ whose underlying tree T satisfies $d_T(x)\leq \frac{d_G(x)+1}{2k}+2$ for all vertices x. Hence if $\Delta(G)\leq 2k(r-2)+2$, Conjecture 9.5.1 would imply that G contains a spanning tree with maximum degree at most $r+\frac{3}{2k}$ and hence would strengthen the result of Czumaj and Strothmann (by showing that we may replace k-connectivity by k-edge-connectivity when k is even).

By Theorem 9.3.1, every 2-arc-strong directed multigraph contains, in particular, the union of two arc-disjoint out-branchings rooted at any given vertex s. So the case $k=2$ of Conjecture 9.5.1 follows from the next theorem.

Theorem 9.5.2 *[117] Let D be a directed multigraph which is the union of two arc-disjoint out-branchings rooted at s. There exists an out-branching B_s^+*

[3] The hamiltonian path problem easily reduces to this problem.

[4] Recall that this means that for every vertex v we have $|d^+_D(v)-d^-_D(v)|\leq 1$.

rooted at s such that $d^+_{B^+_s}(x) \leq \frac{d^+_D(x)}{2} + 1$ for all vertices x of D. Furthermore, such an out-branching can be found in polynomial time.

Proof: Let T^+_s be an out-tree rooted at s and let $D' = D\langle V(T^+_s)\rangle$. We say that T^+_s is **good** if for all vertices x of T^+_s we have

$d^+_{T^+_s}(x) \leq \frac{d^+_{D'}(x)}{2} + 1$ when $d^+_{D'}(x) = d^+_D(x)$ and $d^+_{T^+_s}(x) \leq \frac{d^+_{D'}(x)+1}{2}$ otherwise.

Clearly the out-tree consisting of just the vertex s is good and if one can find a good spanning out-tree, the proof is completed. It suffices then to prove that every non-spanning good out-tree T^+_s is strictly contained in a good out-tree.

Call a vertex x of T^+_s an **out-vertex** if it has an out-neighbour in D which belongs to $V - V(T^+_s)$. It is clear that T^+_s has at least one out-vertex. Suppose one vertex x of T^+_s has precisely one out-neighbour y in D which belongs to $V - V(T^+_s)$ (i.e., $d^+_{D'}(x) = d^+_D(x) - 1$). Then taking $\hat{T}^+_s = T^+_s \cup xy$ and letting D'' be the subdigraph induced by $V(\hat{T}^+_s)$ in D we have

$$\begin{aligned} d^+_{\hat{T}^+_s}(x) = d^+_{T^+_s}(x) + 1 &\leq \frac{d^+_{D'}(x)+1}{2} + 1 \\ &= \frac{d^+_D(x)}{2} + 1 \\ &= \frac{d^+_{D''}(x)}{2} + 1, \end{aligned}$$

implying that $\hat{T}^+_s$ is good. Hence we may assume that every out-vertex $x \in T^+_s$ has at least two out-neighbours belonging to $V - V(T^+_s)$ in D.

Start now from any out-vertex x_1 and denote by y_1 one of its out-neighbours in $V - V(T^+_s)$. As D contains two arc-disjoint (s, y_1)-paths, there exists a path P_1 starting at some vertex x_2 of T^+_s and ending at y_1, with all internal vertices outside of T^+_s and which does not use the arc x_1y_1. Since x_2 has at least two out-neighbours in $V - V(T^+_s)$ it is the origin of an arc $e = x_2y_2$, where $y_2 \notin T^+_s$ and e is not the first arc of P_1. Applying the same argument as above we see that y_2 is the end of a path P_2 starting at some vertex x_3 of T^+_s, with all internal vertices in $V - V(T^+_s)$, and which does not use the arc e. We continue this construction, and let k be the largest integer such that $V(P_1), \ldots, V(P_{k-1})$ are pairwise disjoint. We denote by a the first repeated vertex (which belongs to P_k. Here, by the first repeated vertex, we mean the first vertex among $V(P_1) \cup \ldots \cup V(P_{k-1})$ that we encounter by moving backwards on P_k starting in y_k). If $a = x_1$, we consider the out-tree $A^* = T^+_s \cup P_1 \cup \ldots \cup P_k[a, y_k]$ and let D^* be the subdigraph induced by the vertices of A^*. We claim that A^* is good. To see this, it suffices to observe the following: every new vertex that we add to T^+_s has out-degree one and

for each $i \in [k]$ the out-degree of x_i is one larger in A^* than in T_s^+, while the out-degree of x_i in D^* is at least two larger than in D'.

If $a \neq x_1$, there exists a unique P_i, $i < k$, such that $a \in P_i$. Again it is easy to see that the out-tree $T_s^+ \cup P_i[x_{i+1}, a] \cup P_{i+1} \cup \ldots \cup P_{k-1} \cup P_k[a, y_k]$ is good. The complexity claim follows from the constructive proof above. □

Using induction, with Theorem 9.5.2 as the base case, one can prove the following result which provides some support for Conjecture 9.5.1. The proof is left to the reader as Exercise 9.12.

Theorem 9.5.3 *[117] If a vertex s has k arc-disjoint paths to every other vertex in D, then D has an out-branching B_s^+ rooted at s such that $d^+_{B_s^+}(v) \leq \frac{d^+_D(v)}{2^r} + r$, where $r = \lfloor \log_2 k \rfloor$. Furthermore, such an out-branching can be found in polynomial time.* □

For acyclic directed multigraphs Bang-Jensen, Thomassé and Yeo gave a complete characterization for the existence of an out-branching satisfying given (not necessarily uniform) restrictions on the out-degree of each vertex. For a set of vertices X in D we denote by X^- the set of vertices with at least one arc to a vertex in X. Thus $X^- = \bigcup_{x \in X} N^-(x)$.

Theorem 9.5.4 *[117] Let $D = (V, A)$ be an acyclic directed multigraph and let $f : V \to \mathbb{Z}_0$. Suppose that D has precisely one vertex s of in-degree zero. Then D has an out-branching B_s^+ rooted at s satisfying*

$$d^+_{B_s^+}(v) \leq f(v) \text{ for all } v \in V \tag{9.11}$$

if and only if

$$\sum_{x \in X^-} f(x) \geq |X| \text{ for all } X \subset V - s. \tag{9.12}$$

Furthermore, there exists a polynomial algorithm which given an acyclic directed multigraph and a non-negative integer assignment to its vertices, either finds an out-branching satisfying (9.11) or a set X of vertices violating (9.12). □

In the case of tournaments we can get much more structure on an out-branching as illustrated by the following theorem due to Lu [658].

Theorem 9.5.5 *[658] Let T be a tournament and let v be a vertex of maximum out-degree. Then T contains an out-branching B_v^+ such that*

- *every vertex in B_v^+ except v has out-degree at most two;*
- *the distance from v to any other vertex in B_v^+ is at most 2.*

Proof: We give a proof due to Bondy [166]. Let $Y = N^+(v)$, $X = V - Y - v$ and form the undirected bipartite graph B with bipartition classes X, Y and an edge between $x \in X$ and $y \in Y$ for each $yx \in A(T)$. Now it is easy to see that T has the desired out-branching if and only if we can find a subgraph of B in which every vertex of X has degree one and no vertex in Y has degree more than two. By Hall's theorem (Theorem 4.11.3) such a subgraph exists if and only if every subset $S \subseteq X$ has at least $\frac{1}{2}|S|$ neighbours in Y. (Indeed, apply Hall's theorem to the graph B' that we obtain by substituting two independent vertices for each $y \in Y$.).

Let $S \subset X$ be arbitrary. Since T is a tournament, some $s \in S$ dominates at least $\frac{1}{2}(|S| - 1)$ vertices in S. Since $sv \in A(T)$ and v has maximum out-degree it follows that s dominates at most $|Y| - \frac{1}{2}(|S|+1)$ vertices of Y. Thus at least $\frac{1}{2}(|S| + 1)$ vertices of Y dominate s and hence S has at least $\frac{1}{2}|S|$ neighbours in Y in B. □

9.6 Arc-Disjoint In- and Out-Branchings

We saw in Section 9.3 that the problem of deciding the existence of k arc-disjoint out-branchings all with the same root could be solved efficiently and in Section 9.4 we saw that many problems can be reformulated and solved using an algorithm for the k arc-disjoint out-branchings problem. In this section we consider the following much harder problem called the ARC-DISJOINT IN- AND OUT-BRANCHING PROBLEM: Given a digraph D and vertices u, v (not necessarily distinct). Decide whether D has a pair of arc-disjoint branchings B_u^+, B_v^- such that B_u^+ is an out-branching rooted at u and B_v^- is an in-branching rooted at v.

Theorem 9.6.1 *[68] The arc-disjoint in- and out-branching problem is $\mathcal{NP}$-complete for arbitrary digraphs.*

Proof: We give a proof due to Thomassen (see [68]). The problem belongs to $\mathcal{NP}$, since if the desired branchings exist, then such a pair forms a certificate that the given instance is a 'yes' instance. We show how to reduce the arc-disjoint 2-path problem to the arc-disjoint in- and out-branching problem in polynomial time.

Let $[D, x_1, x_2, y_1, y_2]$ be an instance of the arc-disjoint 2-path problem. Construct a new digraph D' by adding four new vertices x_1', x_2', y_1', y_2' and the following arcs (see Figure 9.4):
$\{x_1'x_1, x_2'x_2, y_1y_1', y_2y_2', x_2'x_1', y_1'x_1', y_2'y_1', y_2'x_2', y_2'x_1'\} \cup \{vx_1' : v \in V(D) - x_1\} \cup \{y_2'v : v \in V(D) - y_2\}$.

The reader can easily verify that there exist arc-disjoint branchings $B^+_{x_2'}$, $B^-_{y_1'}$ in D' if and only if D contains a pair of arc-disjoint (x_1, y_1)-, (x_2, y_2)-paths. Since we can construct D' in polynomial time from D, it follows that the arc-disjoint in- and out-branching problem is $\mathcal{NP}$-complete. □

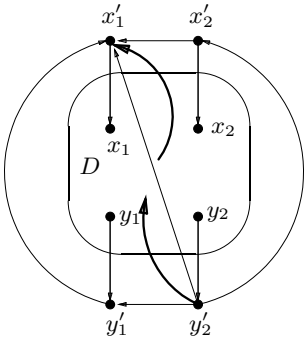


Figure 9.4 The construction of D' in the proof of Theorem 9.6.1. The bold arcs indicate that all the arcs have that direction, except the arcs $x'_1x_1, y_2y'_2$.

For arbitrary digraphs it is easy to reduce the arc-disjoint in- and out-branching problem for the case when $u \neq v$ to the case when $u = v$ by a polynomial reduction (Exercise 9.13). Hence the problem remains $\mathcal{NP}$-complete when we ask for an out-branching and an in-branching that are arc-disjoint and have the same root. However, Bang-Jensen and Huang showed that if the vertex that is to be the root is adjacent to all other vertices in the digraph and is not in any 2-cycle, then the problem becomes polynomially solvable.

Theorem 9.6.2 *[103] Let $D = (V, A)$ be a strongly connected digraph and v a vertex of D such that v is not on any 2-cycle and $V(D) = \{v\} \cup N^-(v) \cup N^+(v)$. Let $\mathcal{A} = \{U_1, U_2, \ldots, U_k\}$ ($\mathcal{B} = \{W_1, W_2, \ldots, W_r\}$) denote the set of terminal (initial) components in $D\langle N^+(v)\rangle$ ($D\langle N^-(v)\rangle$). Then D contains a pair of arc-disjoint branchings B^+_v, B^-_v such that B^+_v is an out-branching rooted at v and B^-_v is an in-branching rooted at v if and only if there exist disjoint arc sets $E_{\mathcal{A}}, E_{\mathcal{B}} \subset A$ such that all arcs in $E_{\mathcal{A}} \cup E_{\mathcal{B}}$ go from $N^+(v)$ to $N^-(v)$ and every $U_i \in \mathcal{A}$ ($W_j \in \mathcal{B}$) is incident with an arc from $E_{\mathcal{A}}$ ($E_{\mathcal{B}}$). Furthermore, there exists a polynomial algorithm to find the desired branchings, or demonstrate the non-existence of such branchings.*

Proof: We prove the characterization and refer the reader to [103] and Exercise 9.14 for the algorithmic part.

First we note that if the branchings exist, then the arc sets $E_{\mathcal{A}}$ and $E_{\mathcal{B}}$ exist. Indeed, if B^+_v, B^-_v are such branchings, then there must be an arc from B^-_v (B^+_v) leaving (entering) every terminal (initial) component of $D\langle N^+(v)\rangle$ ($D\langle N^-(v)\rangle$) and since v is not on any 2-cycle, all these arcs go from $N^+(v)$ to $N^-(v)$.

Suppose that there exist sets $E_{\mathcal{A}}$ and $E_{\mathcal{B}}$ as above. Every vertex $x \in N^+(v)$ has a path to one of the terminal components in $\mathcal{A}$ and every vertex in $N^-(v)$ can be reached by a path from one of the initial components in $\mathcal{B}$. Hence, we can choose a family of vertex-disjoint trees $T^-_1, T^-_2, \ldots, T^-_k$,

$T_1^+, T_2^+, \ldots, T_r^+$ such that T_i^- (T_j^+) is an in-tree (out-tree) rooted at a vertex in U_i (W_j) and $\bigcup_{i=1}^k V(T_i^-) = N^+(v)$, $\bigcup_{j=1}^r V(T_j^+) = N^-(v)$. Let B_v^+ be the out-branching induced by the arcs $\{vw : w \in N^+(v)\} \cup E_{\mathcal{B}} \cup \bigcup_{j=1}^r A(T_j^+)$ and B_v^- be the in-branching induced by the arcs $\{uv : u \in N^-(v)\} \cup E_{\mathcal{A}} \cup \bigcup_{i=1}^k A(T_i^-)$. Then B_v^+ and B_v^- are the desired branchings. □

The following is an easy corollary of Theorem 9.6.2.

Corollary 9.6.3 *[68] A tournament $D = (V, A)$ has arc-disjoint branchings B_v^+, B_v^- rooted at a specified vertex $v \in V$ if and only if D is strong and for every arc $a \in A$ the digraph $D - a$ contains either an out-branching or an in-branching with root v.* □

There is a small inconsistency in the statement (and the proof) of Theorem 9.6.2 in [103] as it was not mentioned that v is not on a 2-cycle and the statement (the part involving the ends to the arcs in $E_{\mathcal{A}}$, $E_{\mathcal{B}}$) becomes slightly different when v is contained in a 2-cycle. However, as the reader is asked to prove in Exercise 9.14, one can still describe a nice characterization and prove that it can be checked in polynomial time whether the desired branchings exist and to find such branchings if they exist.

When v is adjacent to all other vertices, one can prove the following, using Theorem 9.6.2 and the extension in Exercise 9.14 (see Exercise 9.15).

Theorem 9.6.4 *Let D be a 2-arc-strong digraph with a vertex v that is adjacent to all other vertices of D. Then D has arc-disjoint in- and out-branchings rooted at v.* □

Since the discussion above takes care of the semicomplete case, a possible next step is to consider the following problem posed by Bang-Jensen.

Problem 9.6.5 *[89] Characterize those locally semicomplete digraphs D that have arc-disjoint branchings B_v^+, B_v^- for a given vertex $v \in V(D)$.*

When $u \neq v$, the arc-disjoint in- and out-branching problem becomes much harder even for semicomplete digraphs. Bang-Jensen [68] found a complete characterization for the case of tournaments. This characterization, which is quite complicated, implies the tournament case of the following theorem by Bang-Jensen and Yeo.

Theorem 9.6.6 *[120] Every 2-arc-strong semicomplete digraph $T = (V, A)$ contains arc-disjoint in- and out-branchings B_r^-, B_s^+ for every choice of vertices $r, s \in V$.*

Proof: This follows easily from Theorem 13.10.3 since it is easy to show that the semicomplete digraph S_4 which is the unique exception to that theorem has arc-disjoint in- and out-branchings B_u^-, B_v^+ for every choice of $u, v \in V(S_4)$. □

Theorem 9.6.7 *[68] There is a polynomial algorithm for checking whether a given tournament with specified distinct vertices u, v has arc-disjoint branchings B_u^+, B_v^- and finding such branchings if they exist.* □

This algorithm uses the polynomial algorithms from Corollary 10.7.22 and Theorem 10.7.23.

The following conjecture, due to Bang-Jensen, was verified by Bang-Jensen and Huang [103] for the special case when D is quasi-transitive and $u = v$.

Conjecture 9.6.8 *[89] The arc-disjoint in- and out-branching problem is polynomially solvable for locally semicomplete digraphs and quasi-transitive digraphs.*

Thomassen conjectured that there is some sufficient condition, in terms of arc-strong connectivity, for the existence of arc-disjoint in- and out-branchings rooted at the same vertex in a digraph (see also Conjecture 13.10.14).

Conjecture 9.6.9 *[866] There exists a natural number N such that every digraph D which is N-arc-strong has arc-disjoint branchings B_v^+, B_v^- for every choice of $v \in V(D)$.*

Verifying a conjecture from [89] (see also [91, Conjecture 9.9.12]) Bang-Jensen and Yeo proved that for tournaments the following much stronger property holds. Note that if Conjecture 13.10.2 is true, then we may replace $74k$ by $2k$.

Theorem 9.6.10 *[120] Let T be a $74k$-arc-strong tournament. Then T has $2k$ arc-disjoint branchings $B_{v,1}^+, \ldots, B_{v,k}^+, B_{v,1}^-, \ldots, B_{v,k}^-$ such that $B_{v,1}^+, \ldots, B_{v,k}^+$ are out-branchings rooted at v and $B_{v,1}^-, \ldots, B_{v,k}^-$ are in-branchings rooted at v, for every vertex $v \in V(T)$.* □

Conjecture 9.6.11 *[120] Theorem 9.6.10 holds also if we replace $74k$ by $2k$.*

Now let us turn to the case of acyclic directed multigraphs. It is easy to see that if an acyclic directed multigraph D has arc-disjoint branchings B_s^+, B_t^-, where B_s^+ is an out-branching rooted at s and B_t^- is an in-branching rooted at t, then s (t) must be the unique source (sink) in D. The following corollary of Theorem 9.5.4 characterizes when an acyclic directed multigraph contains such a pair of arc-disjoint branchings. Recall the definition of X^- from the end of Section 9.5.

Corollary 9.6.12 *[117] Let D be an acyclic directed multigraph such that there is exactly one sink s of and exactly one source t in D. Then D contains*

arc-disjoint branchings B_s^+ and B_t^- where the first is an out-branching rooted at s and the second is an in-branching rooted at t if and only if we have

$$\sum_{x \in X^-} (d^+(x) - 1) \geq |X| \text{ for all } X \subset V - s. \tag{9.13}$$

Furthermore, there exists a polynomial algorithm which given D, s, t either finds the desired branchings or set X of vertices violating (9.13).

Proof: As remarked above, an acyclic directed multigraph H has an in-branching rooted at a vertex z if and only if z is the unique vertex of out-degree zero in H. Now we see that D has the desired branchings if and only if D has an out-branching rooted at s which satisfies (9.11) with respect to $f(v) = d^+(v) - 1$ for $v \neq t$ and $f(t) = 0$. By Theorem 9.5.4 this is equivalent to requiring that (9.13) must hold. The complexity claim follows from the last part of Theorem 9.5.4. □

9.7 Out-Branchings with Extremal Number of Leaves

A vertex x of an out-branching B is called a **leaf** if $d_B^+(x) = 0$. For an out-branching B, let $L(B)$ denote the set of leaves of B. For a digraph D containing an out-branching, let $\ell_{\min}(D)$ and $\ell_{\max}(D)$ denote the minimum and maximum number of leaves in an out-branching of D. If D has no out-branching, we will write $\ell_{\min}(D) = 0$ or $\ell_{\max}(D) = 0$; recall that, by Proposition 1.7.1, a connected digraph D contains an out-branching if and only if D has only one initial strong component.

The problem of finding an out-branching with extremal number of leaves is of interest in applications, e.g., the problem of finding an out-branching with minimum number of leaves was considered in the US patent [256] by Demers and Downing, where its application to the area of database systems was described.

For general digraphs, the problems of finding an out-branching with minimum/maximum number of leaves are $\mathcal{NP}$-hard: a digraph has an out-branching with just one leaf if and only if it is traceable and by taking the complete biorientation of an undirected graph, we can reduce the $\mathcal{NP}$-hard problem of finding a spanning tree with maximum number of leaves in a connected undirected graph [393] to the problem of finding an out-branching with the maximum number of leaves in a digraph.

Note that restricted to acyclic digraphs the problems of finding an out-branching with minimum and maximum number of leaves are of different complexity (provided $\mathcal{P} \neq \mathcal{NP}$): while the former is polynomial time solvable (see Subsection 9.7.1), the latter is $\mathcal{NP}$-hard (see Exercise 9.21).

9.7.1 Minimum Leaf Out-Branchings

In this subsection, we give upper bounds on $\ell_{\min}(D)$ for general and strong digraphs D and a polynomial algorithm for computing $\ell_{\min}(D)$ for acyclic digraphs D.

Recall that for a digraph D, $\alpha(D)$ denotes the independence number of $UG(D)$. Las Vergnas proved the following upper bound on $\ell_{\min}(D)$ for general digraphs.

Theorem 9.7.1 (Las Vergnas' theorem) *[635] For every digraph D, we have $\ell_{\min}(D) \leq \alpha(D)$.* □

We will prove the following proposition which immediately implies the theorem.

Proposition 9.7.2 *Let B be an out-branching of D with more than $\alpha(D)$ leaves. Then D contains an out-branching B' such that $L(B')$ is a proper subset of $L(B)$.*

Proof: We will prove this claim by induction on the number n of vertices in D. For $n \leq 2$ the result holds; thus, we may assume that $n \geq 3$ and consider an out-branching B of D with $|L(B)| > \alpha(D)$. Clearly, D has an arc xy such that x, y are leaves of B. If the in-neighbor p of y in B is of out-degree at least 2, then $L(B') \subset L(B)$, where $B' = B + xy - py$. So, we may assume that $d^+_B(p) = 1$. Observe $\alpha(D - y) \leq \alpha(D) < |L(B)| = |L(B - y)|$. Hence by the induction hypothesis, $D - y$ has an out-branching B'' such that $L(B'') \subset L(B - y)$. Notice that $L(B - y) = L(B) \cup \{p\} \setminus \{y\}$. If $p \in L(B'')$, then observe that $L(B'' + py) \subset L(B)$. Otherwise, $L(B'' + xy) \subseteq L(B) \setminus \{x\} \subset L(B)$. □

It is easy to show that Las Vergnas' theorem implies the Gallai-Millgram theorem (see Section 13.5). Using Las Vergnas' theorem, we can easily show the following result which is equivalent to another important theorem, Dilworth's theorem (see Section 13.5).

Theorem 9.7.3 *If D if a transitive acyclic digraph with a unique source s, then $\ell_{\min}(D) = \alpha(D)$.*

Proof: By Las Vergnas' theorem, D contains an out-branching B with $k \leq \alpha(D)$ leaves. Observe that B is rooted at s and the vertices of every path in B starting at s and terminating at a leaf induce a clique in $UG(D)$. Thus, the vertices of $UG(D)$ can be covered by k cliques and, hence, $\alpha(UG(D)) \leq k$. We conclude that $\ell_{\min}(D) = \alpha(D)$. □

Las Vergnas proved another upper bound on $\ell_{\min}(D)$.

Theorem 9.7.4 *[635] Let D be a digraph on n vertices such that any two distinct non-adjacent vertices have degree sum at least $2n - 2h - 1$, where $1 \leq h \leq n - 1$. Then $\ell_{\min}(D) \leq h$.* □

Settling a conjecture of Las Vergnas [635], Thomassé [852] proved the following:

Theorem 9.7.5 *If D is a strong digraph, then $\ell_{\min}(D) \leq \max\{\alpha(D)-1, 1\}$.* □

Demers and Downing [256] suggested a heuristic approach for finding, in an acyclic digraph, an out-branching with minimum number of leaves. However, no argument or assertion has been made to provide the validity of their approach and to investigate its computational complexity. Using another approach, Gutin, Razgon and Kim [472] showed that a minimum leaf out-branching in an acyclic digraph can be found in polynomial time.

The following algorithm MINLEAF introduced in [472] returns an out-branching with minimum number of leaves in an acyclic digraph[5]. It is not difficult to prove that MINLEAF is correct and of running time $O(m + n^{1.5}\sqrt{m/\log n})$ (Exercise 9.24).

MINLEAF
Input: An acyclic digraph D with vertex set V.
Output: A minimum leaf out-branching T of D if $\ell_{\min}(D) > 0$ and 'NO', otherwise.

1. Find a source r in D. If there is another source in D, return "no out-branching". Let $V' = \{v' : v \in V\}$.
2. Construct a bipartite graph $B = B(D)$ of D with partite sets $V, V' - r'$ and an edge xy' for each arc $xy \in A(D)$.
3. Find a maximum matching M in B.
4. $M^* := M$. For all $y' \in V'$ not covered by M, set $M^* := M^* \cup \{$an arbitrary edge incident with $y'\}$.
5. $A(T) := \emptyset$. For all $xy' \in M^*$, set $A(T) := A(T) \cup \{xy\}$.
6. Return T.

Figure 9.5 illustrates MINLEAF. There $M = \{rx', xy', zt'\}$ and $T = D - zy$.

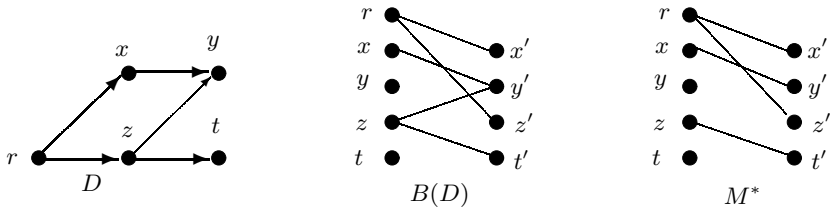


Figure 9.5 Illustration for MINLEAF.

[5] Observe that an acyclic digraph D has an out-branching if and only if it is connected and has exactly one source.

By Lemma 2.13.8, acyclic digraphs are the digraphs of directed tree-width (DAG-width, directed path-width) 0. Dankelmann, Gutin and Kim proved the following:

Theorem 9.7.6 *[241] The problem of finding an out-branching with the minimum number of leaves is $\mathcal{NP}$-hard even for digraphs of directed tree-width (DAG-width, directed path-width, respectively) 1.* □

This theorem is in sharp contrast with the Hamilton path part of Theorem 6.1.4.

9.7.2 Maximum Leaf Out-Branchings

Bonsma and Dorn [173] showed that the problem of checking whether a digraph has an out-branching with at least k leaves is fixed-parameter tractable. Lower bounds on the maximum number of leaves in an out-branching of a digraph were investigated by Alon, Fomin, Gutin, Krivelevich and Saurabh [22, 21] (there is extensive literature on the maximum number of leaves in a spanning tree of a connected undirected graph [22]).

For a digraph D let $\ell^t_{\max}(D)$, denote the maximum possible number of leaves in an out-tree of D. Clearly $\ell_{\max}(\overleftrightarrow{G}) = \ell^t_{\max}(\overleftrightarrow{G})$ if G is a connected undirected graph, as any maximum leaf tree can be extended to a maximum leaf spanning tree with the same number of leaves.

Notice that $\ell^t_{\max}(D) \geq \ell_{\max}(D)$ for every digraph D. Let $\mathcal{L}$ be the family of digraphs D for which either $\ell_{\max}(D) = 0$ or $\ell^t_{\max}(D) = \ell_{\max}(D)$. It is easy to see that $\mathcal{L}$ contains all strong and acyclic digraphs.

The following assertion, whose proof is left to the reader as Exercise 9.16, shows that $\mathcal{L}$ includes a large number of digraphs including all strong digraphs, acyclic digraphs, semicomplete multipartite digraphs and quasi-transitive digraphs.

Proposition 9.7.7 *[22] Suppose that a digraph D satisfies the following property: for every pair R and Q of distinct strong components of D, if there is an arc from R to Q, then each vertex of Q has an in-neighbor in R. Then $D \in \mathcal{L}$.* □

Let $P = u_1u_2 \dots u_q$ be a directed path in a digraph D. An arc u_iu_j of D is a **forward** (**backward**) **arc for** P if $i \leq j-2$ ($j < i$, respectively). Every backward arc of the type $v_{i+1}v_i$ is called **double**.

The following assertion is a slight refinement of a result by Alon, Fomin, Gutin, Krivelevich and Saurabh [22]. For a better bound, see [21].

Lemma 9.7.8 *Let D be an oriented graph of order n containing an out-branching and with $d^-(x) = 2$ for all $x \in V(D)$. If D has no out-tree with k leaves, then $n \leq k^5$.*

Proof: Assume that D has no out-tree with k leaves. Consider an out-branching B^+ of D with p leaves so that this is the maximum number of leaves over all out-branchings in D. By the assumption $p < k$.

First observe that if $Q = v_1v_2 \dots v_s$ is an arbitrary path in B^+ from the root to a leaf and v_iv_j is a forward arc, then, by the maximality of p, B^+ must branch at v_{j-1}, that is, $d^+_{B^+}(v_{j-1}) \geq 2$. Since B^+ has at most $k-1$ leaves and no two forward arcs end in the same vertex, this implies that Q has at most $(k-2)$ forward arcs.

Now fix a path $P = u_1u_2 \dots u_q$ from the root to a leaf in B^+ which has $q \geq n/p$ vertices. When we delete all vertices of P from B^+ we obtain a collection of out-trees covering $V(D) - V(P)$. It is easy to show by induction on the number of leaves that B^+ can be decomposed into a collection $\mathcal{P} = \{P_1, P_2, \dots, P_p\}$ of vertex-disjoint directed paths covering all vertices of D so that $P = P_1$.

Let $P' \in \mathcal{P} \setminus \{P\}$ be arbitrary. There are at most $k-1$ vertices on P with in-neighbors on P' since otherwise we could choose a set X of at least k vertices on P for which there were in-neighbors on P'. The vertices of X would be leaves of an out-tree formed by the vertices $V(P') \cup X$. Thus, there are $m \leq (k-1)(p-1) \leq (k-1)(k-2)$ vertices of P with in-neighbors outside P and at least $q - (k-2)(k-1)$ vertices of P have both in-neighbors on P.

Let f be the number of forward arcs for P. By the argument above $f \leq k-2$. Let uv be an arc of $A(D) \setminus A(P)$ such that $v \in V(P)$. There are three possibilities: (i) $u \notin V(P)$, (ii) $u \in V(P)$ and uv is forward for P, (iii) $u \in V(P)$ and uv is backward for P. By the inequalities above for m and f, we conclude that there are at most $k(k-2)$ vertices on P which are not terminal vertices (i.e., heads) of a backward arc. Consider a path $R = v_0v_1 \dots v_r$ formed by backward arcs. Observe that the arcs $\{v_iv_{i+1} : 0 \leq i \leq r-1\} \cup \{v_jv_j^+ : 1 \leq j \leq r\}$ form an out-tree with r leaves, where v_j^+ is the successor of v_j on P. Thus, there is no path of backward arcs of length more than $k-1$.

If the in-degree of u_1 in $D[V(P)]$ is 2, remove one of the backward arcs terminating at u_1. Observe that now the backward arcs for P form a vertex-disjoint collection of out-trees with roots at vertices that are not terminal vertices of backward arcs. Therefore, the number of out-trees in the collection is at most $k(k-2)$. Observe that each out-tree in the collection has at most $k-1$ leaves and thus its arcs can be decomposed into at most $k-1$ paths, each of length at most $k-1$. Hence, the original total number of backward arcs for P is at most $k(k-2)(k-1)^2+1$ (where the last one comes from the possible extra arc into u_1). On the other hand, it is at least $q-k(k-2)$ as every vertex on P is the head of an arc not in $A(P)$. Thus, $q-k(k-2) \leq k(k-2)(k-1)^2+1$. Combining this inequality with $q \geq n/(k-1)$, we conclude that $n \leq k^5$. □

Theorem 9.7.9 *[22] Let D be a digraph in $\mathcal{L}$ with $\ell_{\max}(D) > 0$.*

(a) If D is an oriented graph with $\delta^-(D) \geq 2$, then $\ell_{\max}(D) \geq n^{1/5} - 1$.
(b) If D is a digraph with $\delta^-(D) \geq 3$, then $\ell_{\max}(D) \geq n^{1/5} - 1$.

Proof: Let B^+ be an out-branching of D. By deleting some arcs from $A(D) \setminus A(B^+)$, if necessary, we may assume that $d^-(x) = 2$ for every $x \in V(D)$. Now the inequality $\ell_{\max}(D) \geq n^{1/5} - 1$ follows from Lemma 9.7.8 and the definition of $\mathcal{L}$.

Let B^+ be an out-branching of D. Let P be the path formed in the proof of Lemma 9.7.8. (Note that $A(P) \subseteq A(B^+)$.) Delete every double arc of P, in case there are any, and delete some more arcs from $A(D) \setminus A(B^+)$, if needed, to ensure that the in-degree of each vertex of D becomes 2. It is not difficult to see that the proof of Lemma 9.7.8 remains valid for the new digraph D. Now the inequality $\ell_{\max}(D) \geq n^{1/5} - 1$ follows from Lemma 9.7.8 and the definition of $\mathcal{L}$. □

It is not difficult to give examples showing that the restrictions on the minimum in-degrees in Theorem 9.7.9 are optimal. Indeed, any directed cycle C is a strong oriented graph with all in-degrees 1 for which $\ell_{\max}(C) = 1$ and the complete biorientation of any cycle is a strong digraph D with all in-degrees equal to 2 and $\ell_{\max}(D) = 2$.

For some subfamilies of $\mathcal{L}$, one can obtain better bounds on $\ell_{\max}(D)$. An example is the class of multipartite tournaments. Recall from Section 3.7 that every multipartite tournament D with at most one source has an out-branching B^+ such that the distance from the root of B^+ to any vertex is at most 4. This implies that $\ell_{\max}(D) \geq \frac{n-1}{4}$. Also for a tournament D of order n, it is easy to prove that $\ell_{\max}(D) \geq n - \log_2 n$ (Exercise 9.25). This bound is essentially tight, i.e., we cannot replace the right-hand side by $n - \log_2 n + \Omega(\log_2 \log_2 n)$ as shown by random tournaments; see [29, pages 3-4] for more details.

Solving an open problem from [21], Bonsma and Dorn proved the following result.

Theorem 9.7.10 *[174] Let D be a digraph on n vertices with at least one out-branching. If $\delta^-(D) \geq 3$ or if D is an oriented graph and $\delta^-(D) \geq 2$, then $\ell_{max}(D) \geq \frac{1}{4}\sqrt{n}$.* □

Results in [21] imply that this bound is best possible modulo the coefficient, i.e., $\ell_{max}(D) = \Theta(\sqrt{n})$.

9.8 The Source Location Problem

Definition 9.8.1 *Let $D = (V, A)$ be a directed multigraph and let k, l be non-negative integers. A subset $S \subseteq V$ is a (k, l)-**source** for D if S has k arc-disjoint paths to every vertex $v \in V - S$ and every vertex $v \in V - S$ has l arc-disjoint paths to S.*

By Theorem 9.3.1, S is a (k, l)-source if and only if the digraph we obtain by contracting S to a new vertex s has k arc-disjoint out-branchings rooted at

s and l arc-disjoint in-branchings rooted at s. It follows from this remark and Menger's theorem that if $d^-(X) < k$ or $d^+(X) < l$, then every (k,l)-source S must intersect X. Note that, trivially, $V(D)$ is a (k,l)-source. The SOURCE LOCATION PROBLEM is as follows: Given $[D,k,l]$; find a (k,l)-source S of D of minimum size.

Call a set $X \subset V$ (k,l)**-bad** if $d^-(X) < k$ or $d^+(X) < l$ and no proper subset of X has this property. Clearly, S is a (k,l)-source if and only if it intersects all (k,l)-bad sets. Now we can translate the source location problem into a problem of finding transversals of hypergraphs as follows. Given D,k and l let $\mathcal{H}_{(k,l)}(D) = (V,\mathcal{E})$ be the hypergraph with vertex set V and $\mathcal{E}$ contains the hyperedge X for each (k,l)-bad set X. A **transversal** of $\mathcal{H}_{(k,l)}(D)$ is a set of vertices containing at least one vertex from each edge. Hence the source location problem is equivalent to finding a minimum size transversal of $\mathcal{H}_{(k,l)}(D)$.

Clearly the minimum size of a (k,l)-source is at least as big as the maximum number of disjoint (k,l)-bad sets. Ito, Makino, Arata, Honami, Itatsu and Fujishige proved that the other direction also holds.

Lemma 9.8.2 *[553] For every digraph D and non-negative numbers k,l the size of a minimum (k,l)-source of D equals the maximum number of disjoint (k,l)-bad sets.* □

A hypergraph $\mathcal{H} = (V,\mathcal{E})$ is a **subtree hypergraph** if there exists a tree T on V such that each hyperedge of $\mathcal{E}$ induces a subtree of T.

In [523] van den Heuvel and Johnson proved the following result, implying that a minimum size transversal of a subtree hypergraph can be found in polynomial time, provided that an oracle for deciding, in polynomial time, whether a subset is a transversal or not is given[6].

Theorem 9.8.3 *[523] Let $\mathcal{H} = (V,\mathcal{E})$ be a subtree hypergraph on n vertices. If it is possible to check whether or not a subset $S \subseteq V$ is a transversal in time $g(n)$, then it is possible to find a minimum size transversal of $\mathcal{H}$ in time $O(n^3 g(n))$.* □

Proposition 9.8.4 *[553] For every digraph and non-negative integers k,l the hypergraph $\mathcal{H}_{(k,l)}(D)$ is a subtree hypergraph.* □

The next lemma follows easily from our remark just after the definition of a (k,l)-source.

Lemma 9.8.5 *[523] Using flows one can check, in polynomial time, whether a given set of vertices is a (k,l)-source.* □

Thus combining Proposition 9.8.4, Lemma 9.8.5 and Theorem 9.8.3 we obtain the following.

[6] Without this assumption the problem is NP-hard [523].

Theorem 9.8.6 *[523] The source location problem is solvable in polynomial time.* □

Using an entirely different approach, Bárász, Becker and Frank [124] found another polynomial algorithm for the source location problem. In [524] van den Heuvel and Johnson studied a problem closely related to the source location problem.

9.9 Miscellaneous Topics

9.9.1 Edge-Disjoint Mixed Branchings

We saw in the proof of Theorem 9.4.2 that we could use Edmonds' branching theorem to prove that every $2k$-edge-connected graph has k edge-disjoint spanning trees. However, that proof does not imply an algorithm to check whether a given undirected graph has k edge-disjoint spanning trees. In fact, this problem is more complicated for undirected graphs than the problem of finding k arc-disjoint out-branchings from a given root in a directed multigraph where the proof of Edmonds' branching theorem provides the answer. For undirected graphs the characterization, given in Theorem 9.4.3, is much more complicated and does not imply a polynomial algorithm for the problem. Such an algorithm can be obtained from a formulation of the problem as a matroid partition problem (see Exercise 18.27). See also the remark at the end of the subsection.

A **mixed multigraph** is the same as a mixed graph, except that we allow parallel arcs and parallel edges as well as arcs that are parallel to edges. We say that two subgraphs of a mixed multigraph are **edge-disjoint** if they do not share any arcs or edges (they may contain different copies of an arc/edge, but not the same).

Definition 9.9.1 *Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s. A mixed out-branching B_s^+ with root s is a spanning tree in the underlying undirected multigraph G of M with the property that there is a path from s to every other vertex v in B_s^+.*

One reason why mixed out-branchings are of interest in relation to undirected graphs can be seen from the following easy lemma (which in particular covers the case when no arc of M is directed).

Lemma 9.9.2 *Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s called root. There are k edge-disjoint mixed out-branchings rooted at s if and only if there exists an orientation D of M with k edge-disjoint out-branchings at s.*

Proof: Exercise 9.17. □

The following characterization, due to Frank, generalizes Theorems 9.4.3 and 9.3.1.

Theorem 9.9.3 *[336] Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s. There are k edge-disjoint mixed out-branchings rooted at s, if and only if the following holds for all subpartitions $\mathcal{F} = \{V_1, V_2, \ldots, V_t\}$ of $V - s$:*

$$a_{\mathcal{F}} \geq kt, \tag{9.14}$$

where $a_{\mathcal{F}}$ denotes the number of edges, oriented or not, which enter some V_i. □

One can use submodular flows to decide in polynomial time whether a given undirected graph G has k edge-disjoint spanning trees. By Lemma 9.9.2, all we need to check is whether there is some orientation of G which has k arc-disjoint out-branchings from a given vertex. Thus, given G we form an arbitrary orientation D of G and then follow the approach in Exercise 11.67. It is not hard to see that, with a slight modification, the same approach can be used to determine the existence of k edge-disjoint mixed branchings from a given root in a mixed graph (Exercise 9.18).

9.9.2 The Minimum Covering Out-Tree Problem

It is easy to decide whether a digraph D has some out-tree rooted at a prescribed vertex s which covers (that is, contains the vertices of) a certain specified subset $X \subseteq V(D)$ (Exercise 9.26). This makes it natural to consider the following problem which we call the MINIMUM COVERING OUT-TREE PROBLEM. Given a digraph $D = (V, A)$ with a non-negative integer-valued weight function w on the arcs, some vertex $s \in V$ and a subset $X \subseteq V$. What is the minimum cost of an out-tree T_s^+ rooted in s such that $X \subseteq V(T_s^+)$?

Theorem 9.9.4 *The minimum covering out-tree problem is $\mathcal{NP}$-hard even when $w \equiv 1$.*

Proof: The GRAPH STEINER PROBLEM is as follows (this is a special case[7], but already this is $\mathcal{NP}$-complete, see Exercise 9.27). Given an undirected graph $G = (V, E)$ and a subset $X \subset V$, find a subtree of G which contains all vertices of X and as few other vertices as possible. We show how to reduce the graph Steiner problem to the special case $w \equiv 1$ of the minimum covering out-tree problem in polynomial time. Let $[G, X]$ be an instance of the graph Steiner problem and construct an instance $[D, X, s]$ of the minimum covering

[7] A more well-known version is the so-called STEINER TREE PROBLEM for graphs. Here one is given an undirected graph $G = (V, E)$ with non-negative cost on the edges and a subset $S \subseteq V$ and the goal is to find a minimum cost tree containing all the vertices of S.

out-tree problem by letting D be the complete biorientation of G, taking s as some vertex from X and using the same X. Every tree T which covers X in G corresponds in the obvious way to an out-tree T_s^+ in D which covers X and vice versa. This completes the construction which can obviously be performed in polynomial time. Since the graph Steiner problem is $\mathcal{NP}$-hard [585], we conclude that so is the minimum covering out-tree problem. □

It follows from results by Frank in [349] that if all arcs whose head does not belong to X have cost zero, then the problem can be solved in polynomial time. In fact, the model in [349] shows that even the generalization where one is seeking k arc-disjoint out-trees with a common root all of which cover a prescribed subset X can be solved in polynomial time, provided the cost of all arcs whose head do not belong to X is zero.

9.9.3 Minimum Cost Arc-Disjoint Branchings with Bandwidth Constraints

Consider the following typical problem in network communications. A collection of k distinct messages is to be transferred from a source node s to all other nodes in the network. Each message is transferred from the source to the recipients via an out-branching in the network. A subset of r of these branchings may overlap in a link (arc) a if the corresponding link (arc) has enough bandwidth to send all r messages without interference at the same time. If we also make the sensible assumption that using different links may have different costs, then we see that we may model the problem above as an instance of the so-called MINIMUM COST k OUT-BRANCHINGS WITH BANDWIDTH CONSTRAINTS PROBLEM [187]. Here the goal is, given a digraph $D = (V, A)$, a root $s \in V$, an integer-valued capacity function b and a cost function c, both on A; find a collection of k out-branchings from s such that the arc a is used by at most $b(a)$ of the branchings for each $a \in A$ and the total cost[8] of the branchings is as small as possible.

The problem can be solved using matroid techniques as follows: Construct a new directed multigraph $D^* = (V, A^*)$ by replacing each arc $a \in A$ by $b(a)$ copies, each of cost $c(a)$. Clearly the desired branchings in D correspond to a minimum cost set of arcs in D^* which can be partitioned into k arc-disjoint out-branchings rooted at s. Define two matroids M_1, M_2 on the arc set of D as follows. A subset $A' \subset A$ is independent in M_1 if the corresponding edges in $UG(D)$ can be partitioned into k forests and a subset $A'' \subset A$ is independent in M_2 if s has in-degree zero and all other vertices have in-degree at most k in the subdigraph induced by A''. It is easy to show that M_2 is a matroid and the fact that M_1 is a matroid follows from the definition of the union of matroids (see Section 18.8). Now we can solve the problem of finding a minimum cost collection of k arc-disjoint out-branchings in D^* as

[8] Here the cost is the sum of the costs of all the branchings.

an instance of the weighted matroid intersection problem for the matroids M_1 and M_2.

The approach above has several drawbacks: first the number of arcs may increase drastically when we replace each arc a by $b(a)$ copies and second we need to apply an algorithm for matroid partitioning as a subroutine (to check whether the current set of arcs is independent in M_1). In [187] Cai, Deng and Wang showed how to formulate the minimum cost k out-branchings with bandwidth constraints problem as the problem of finding an optimal intersection of two weighted polymatroids on A. They also showed that this leads to a more efficient algorithm for the problem.

9.9.4 Out-Forests

An **out-forest** in a digraph is a spanning collection of disjoint out-trees. Below we describe a result due to El-Sahili and Kouider [294] which has several implications as shown in Section 11.3. By a **spanning out-forest** in D we mean a collection of disjoint out-trees which cover $V(D)$. The **level** of a vertex in an out-tree is its distance from the root. For a given out-forest $\mathcal{F}$ we define the ith level L_i of $\mathcal{F}$ to be the set of vertices whose level is i (in the out-tree to which they belong). Thus L_0 is the set of roots of out-trees in $\mathcal{F}$, L_1 is the set of out-neighbours of these roots in $\mathcal{F}$ and so on. Let $\ell_i = |L_i|$ and associate to each spanning out-forest $\mathcal{F}$ the vector $v(\mathcal{F}) = (\ell_0, \ell_1, \ldots, \ell_{p(\mathcal{F})})$, where $p(\mathcal{F})$ denotes the length of the longest path in $\mathcal{F}$.

Proposition 9.9.5 *[294] Every digraph contains a spanning out-forest $\mathcal{F}$ in which L_i is an independent set for $i = 0, 1, \ldots, p(\mathcal{F})$. In particular, every spanning out-forest $\mathcal{F}$ minimizing $v(\mathcal{F})$ lexicographically (among all spanning out-forests) has this property.*

Proof: Let $\mathcal{F}^*$ be chosen among all spanning out-forests so as to minimize $v(\mathcal{F}^*)$ lexicographically and let $L_0, L_1, \ldots$ be its levels defined as above. We claim that each L_i is an independent set in D. Suppose to the contrary that uv is an arc of D such that $u, v \in L_i$ for some i. Note that uv is not an arc of $\mathcal{F}^*$ as u and v have the same level. Thus we can modify the two out-trees $T_u, T_v \in \mathcal{F}^*$ containing u and v respectively by removing the part of T_v rooted in v from T_v and moving it to T_u by adding the arc uv. Note that we may have $T_u = T_v$. The resulting out-forest $\mathcal{F}$ has $v(\mathcal{F}) < v(\mathcal{F}^*)$, contradicting the choice of $\mathcal{F}^*$. □

9.9.5 The Maximum Weight Out-Forest Problem

The MAXIMUM WEIGHT OUT-FOREST PROBLEM is the problem of finding, in a weighted digraph $D = (V, A)$ (with weight function $c : A \to \mathbb{R}_+$), an out-forest in D whose total arc weight is maximum. In the special case when we want the forest to have only one out-tree we have the maximum weight

out-branching problem. This is clearly equivalent to the minimum weight out-branching problem since we can transform one to the other by modifying the weights as follows: $c'(a) = c(a^*) - c(a)$, where a^* is a maximum weight arc.

To find a maximum weight out-forest in D, we can simply extend D to a new weighted digraph D' by adding a new vertex r which dominates all vertices in V and assign weight zero to all of these arcs. Clearly a maximum weight out-branching rooted at r in D' corresponds to a maximum weight out-forest in D. As we have seen in Section 9.2, a maximum cost out-branching can be found quite efficiently in an arc weighted digraph. However, in some practical applications it is the calculation of the weight function c which consumes the most time. Such an example is given by Ouyang, Memon, Suel and Trendafilov [734], where a problem related to data compression is formulated as a maximum weight spanning out-forest problem. In this application, the calculation of the optimum out-forest, once the weights are found, takes only a very small fraction of the time it takes to calculate all arc weights (where the weight of an arc corresponds to how well one file can be compressed with respect to another file).

In the application above and others too, it is possible to estimate, for a given positive integer k, which are the k incoming arcs at each vertex with the highest weight. Thus it is relevant to see how well one can approximate the cost of an optimum out-forest by making the calculation on the digraph D_k consisting only of the k maximum weight arcs entering each vertex. Bagchi, Bhargava and Suel proved the following result.

Theorem 9.9.6 *[56] Let $D = (V, A)$ be a digraph and let c be a non-negative cost function on A. Let k be a natural number and define $D_k = (V, A_k)$ to be the subdigraph of D induced by the set of the k arcs of maximum cost entering each vertex in V. Denote by $OPT(H)$ the maximum weight of an out-forest in the digraph H. Then we have*

$$\frac{OPT(D_k)}{OPT(D)} \geq \frac{k}{k+1}. \tag{9.15}$$

□

This is best possible as seen from the class of digraphs $H_k = (V, A)$, where $V = \{u, v, v_1, v_2, \ldots, v_k\}$ and $A = \{uv, vv_1, v_1v, vv_2, v_2v, \ldots, vv_k, v_kv\}$ and letting all arcs of the form v_iv have cost $1 + \epsilon$ and the remaining arcs have cost 1. Here $OPT(D_k) = k + \epsilon$ and $OPT(D) = k + 1$ so as $\epsilon \to 0$ we get the ratio in the theorem [56].

We will not prove Theorem 9.9.6 here but just give a short argument for $k = 1$. Let D, c be given and define D' as we did above by adding a new vertex r and arcs of cost zero from r to all vertices of D. Let D'_1 be the digraph induced by the heaviest arc entering each vertex in D' except r (breaking ties arbitrarily). If D'_1 is an out-branching (from r) it is clearly

optimal. Otherwise we may discard the lowest weight arc from each cycle in D'_1 and add an arc from r to the head of the arc we remove. Clearly this results in an out-branching from r whose cost is at least half of the optimum one. Now we obtain the desired out-forest by deleting r.

If, instead of looking for an optimum out-forest, we want to find an optimum out-branching from a specified root in a given arc-weighted digraph, then there can be no such approximation guarantee. It is easy to construct examples for every k, where $\frac{OPT(D_k)}{OPT(D)}$ can be made arbitrarily small (Exercise 9.20). This does not contradict the argument above for $k = 1$, since generally arcs leaving the root may have any non-negative cost.

9.9.6 Branchings and Edge-Disjoint Trees

Clearly, if a directed multigraph D has two arc-disjoint out-branchings $B^+_{s_1}, B^+_{s_2}$ (possibly $s_1 \neq s_2$), then each underlying multigraph $UMG(D - A(B^+_{s_i}))$ is connected and hence has a spanning tree for $i = 1, 2$. The following problem attributed to Thomassé[9] can be seen as an attempt to find a result linking Tutte's characterization of graphs with two edge-disjoint trees and a weakening of Edmonds' branching theorem (for $k = 2$).

Problem 9.9.7 *Find a good characterization of directed multigraphs D for which there exists an out-branching B^+_s rooted at some vertex of D so that $UMG(D - B^+_s)$ is connected.*

9.10 Exercises

9.1. Show how to derive Menger's theorem (Theorem 5.4.1) from Edmonds' branching theorem (Theorem 9.3.1).

9.2. **Greedy min cost branching algorithms may fail.** Construct examples of weighted digraphs for which the natural generalization of Kruskal's algorithm for finding a minimum spanning tree to directed multigraphs will fail to find a minimum cost out-branching from the specified root.

9.3. **Efficient implementation of independence oracles for the matroid intersection formulation of the minimum cost branching problem.** Show how to implement the necessary oracles for testing independence in the two matroids M_1, M_2 which were used in Subsection 9.2.1. Your algorithms should have complexity around $O(m)$, where m is the number of arcs in the directed multigraph.

9.4. **(+) Finding a minimum cost subdigraph which has k arc-disjoint out-branchings rooted at s in a directed multigraph**. Show how to formulate this as a matroid intersection problem. Then sketch an algorithm to find the desired branchings. Hint: modify the matroids M_1, M_2 from Subsection 9.2.1.

[9] It is mentioned on the URL http://www.cs.elte.hu/egres/

9.5. **(+) Finding a minimum cost set of new arcs to add to a directed multigraph in order to ensure the existence of k arc-disjoint out-branchings with a specified root**. Show how to solve this problem (formulated just before Section 9.2.2) using an algorithm for weighted matroid intersection. Hint: use a similar approach as that in Exercise 9.4. Compare also with Exercise 11.67.

9.6. **Formulating the minimum spanning tree problem as a minimum cost branching problem.** Show that the minimum spanning tree problem (given a connected undirected graph with non-negative weights on the edges, find a spanning tree of minimum weight) can be formulated and solved as a minimum cost branching problem.

9.7. **(+) A polynomial algorithm for finding k arc-disjoint out-branchings from a specified root.** Show how to turn the proof of Theorem 9.3.1 into a polynomial algorithm which either finds a collection of k arc-disjoint branchings with root z, or a proof that no such collection of branchings exists. Hint: use flows.

9.8. **Greedy algorithm for arc-disjoint branchings.** Instead of applying the algorithmic version of Theorem 9.3.1 to find k arc-disjoint out-branchings with a given root, one may try a greedy approach: find an out-branching B_z^+ from z. Delete all arcs of B_z^+. Find a new out-branching, delete its arcs and so on. Give an example of a digraph D which has 2 arc-disjoint out-branchings with root z, but not every out-branching B_z^+ can be deleted while leaving another with root z.

9.9. **(+) Arc-disjoint out-branchings with possibly different roots.** Prove the following result due to Frank [336]: In a directed multigraph $D = (V, A)$ there are k arc-disjoint out-branchings (possibly with different roots) if and only if

$$\sum_{i=1}^{t} d^-(X_i) \geq k(t-1) \tag{9.16}$$

holds for every subpartition $\{X_1, X_2, \ldots, X_t\}$ of V. Hint: add a new vertex s and a minimal set of new arcs from s to V so that s is the root of k out-branchings in the new graph. Prove that this minimal set of arcs has precisely k arcs.

9.10. Generalize the example in Figure 9.3 to digraphs with arbitrarily many vertices.

9.11. Construct, for every $k \geq 2$, a k-arc-strong and k-regular directed multigraph which has no hamiltonian path. Hint: you may construct one which has two vertices so that removing these the remaining graph has four connected components.

9.12. Prove Theorem 9.5.3. Hint: use induction on $r = \lfloor \log k \rfloor$. Show how to transform your proof into a polynomial algorithm for finding the desired out-branching.

9.13. Show how to reduce the arc-disjoint in- and out-branching problem for the case $u \neq v$ to the case $u = v$.

9.14. (+) Extend Theorem 9.6.2 to the case when v is on some 2-cycle. Hint: how should the sets $E_{\mathcal{A}}, E_{\mathcal{B}}$ and the branchings described be modified?

9.15. Prove Theorem 9.6.4. Hint: use Theorem 9.6.2 and Exercise 9.14.

9.16. Prove Proposition 9.7.7.

9.17. Prove Lemma 9.9.2.

9.18. Show how to use submodular flows to decide in polynomial time whether a mixed graph M has k edge-disjoint mixed branchings from a given root. Hint: see Exercise 11.67 and adjust the upper/lower bounds on arcs appropriately.

9.19. (+) Prove Theorem 9.4.3. Hint: use Edmonds' branching theorem and Theorem 11.7.6.

9.20. For every choice of natural numbers k, K construct a digraph $D = (V, A)$ and a cost function c on A so that for some vertex r the cost of a maximum weight out-branching from r in D is at least K times higher than the cost of a maximum weight out-branching from r in D_k, where D_k is as in Theorem 9.9.6. Hint: start with $k = 1$ and generalize your construction to arbitrary k. Also note that we do not require that every vertex has at least k arcs entering in D (in which case D_k will contain all arcs entering that vertex).

9.21. Prove that the problem of finding an out-branching with maximum number of leaves in an acyclic digraph is $\mathcal{NP}$-hard. Hint: transform to this problem the set cover problem formulated as a bipartite graph problem, i.e., given a bipartite graph $B = (X, Y; E)$ find a minimum cardinality subset C of X such that $N(C) = Y$. (Alon, Fomin, Gutin, Krivelevich and Saurabh [23].)

9.22. **Arc-disjoint out-branchings with few leaves in tournaments.** Let T be a tournament with 3 arc-disjoint out-branchings rooted at $s \in V(T)$. Prove that T contains 3 arc-disjoint out-branchings $B^+_{s,1}, B^+_{s,2}, B^+_{s,3}$ such that $B^+_{s,i}$ has at most 4 leaves for $i = 1, 2, 3$. Explain briefly how to obtain 3 arc-disjoint out-branchings $B^+_{s,1}, B^+_{s,2}, B^+_{s,3}$ as above if we start from 3 arbitrary arc-disjoint out-branchings from s in a tournament T. Hint: consider the independence number of T minus one or more branchings.

9.23. (−) Prove that the problem of checking whether a digraph has an out-branching with at most k leaves is $\mathcal{NP}$-hard for each fixed natural number k.

9.24. Prove that the algorithm MINLEAF described is correct and of running time $O(m+n^{1.5}\sqrt{m/\log n})$. Hint: use the fact that there is an algorithm of running time $O(n^{1.5}\sqrt{m/\log n})$ for finding a maximum matching in a bipartite graph of order n and size m [39].

9.25. For a tournament T of order n, prove that $\ell_{\max}(T) \geq n - \log_2 n$ (Alon, Fomin, Gutin, Krivelevich and Saurabh [22]).

9.26. **Finding an out-tree which covers a prescribed vertex set.** Show how to decide in polynomial time whether a digraph $D = (V, A)$ has an out-tree with root s which contains all vertices of a prescribed subset $X \subseteq V$ (and possibly other vertices).

9.27. Show that the graph Steiner problem is $\mathcal{NP}$-hard by describing a reduction of the set covering problem to the graph Steiner problem.

10. Linkages in Digraphs

We saw in Chapter 5 that it is easy to check (e.g., using flows) whether a directed multigraph $D = (V, A)$ has k (arc)-disjoint paths $P_1, P_2, \ldots, P_k$ from a subset $X \subset V$ to another subset $Y \subset V$ and we can also find such paths efficiently. On many occasions (e.g., in practical applications) we need to be able to specify the initial and terminal vertices of each P_i, $i = 1, 2, \ldots, k$, that is, we wish to find a so-called **linkage** from $X = \{x_1, x_2, \ldots, x_k\}$ to $Y = \{y_1, y_2, \ldots, y_k\}$ such that P_i is an (x_i, y_i)-path for every $i \in [k]$. This problem is considerably more difficult and is in fact $\mathcal{NP}$-complete already when $k = 2$. In this chapter we start by giving a proof of this fact and then we discuss a number of results on sufficient conditions for the existence of linkages, polynomial algorithms for special classes of digraphs, including acyclic, planar and semicomplete digraphs in the case of vertex disjoint paths and acyclic digraphs and some generalizations of tournaments in the case of arc-disjoint paths. The reader will see that quite a lot can be said about the linkage problems for special classes of digraphs and that still the problems are not trivial for these classes of digraphs. Finally we briefly discuss topics such as multi commodity flows and subdivisions of transitive tournaments in digraphs with large out-degree.

10.1 Additional Definitions and Preliminaries

Recall from Chapter 5 that for a digraph $D = (V, A)$ with distinct vertices x, y we denote by $\kappa_D(x, y)$ the largest integer k such that D contains k internally disjoint (x, y)-paths. When discussing intersections between paths P, Q we will often use the phrase 'let u be the first (last) vertex on P which is on Q'. By this we mean that if, say, P is an (x, y)-path, then u is the only vertex of $P[x, u]$ ($P[u, y]$) which is also on Q.

Let $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ be distinct vertices of a digraph D. A ***k*-linkage** from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D is a system of vertex-disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path in D[1]. A digraph

[1] Sometimes one allows that the paths may share one or both of their end-vertices, i.e., $V(P_i) \cap V(P_j) \subseteq \{x_i, y_i, x_j, y_j\}$ whenever $i \neq j$, where $x_i = y_j$ or $x_i = x_j$ is possible.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_10,

$D = (V, A)$ is ***k*-linked** if it contains a k-linkage from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ for every choice of distinct vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$. A digraph D is ***k*-(arc)-cyclic** if it has a cycle containing the vertices (arcs) $x_1, x_2, \ldots, x_k$ $(a_1, a_2, \ldots, a_k)$ for every choice of k vertices (arcs). The following easy observation is left to the reader as Exercise 10.1.

Proposition 10.1.1 *Every k-linked digraph is k-cyclic and every $2k$-cyclic digraph is k-linked.* □

There is a close relation between linkage problems and problems concerning cycles through prescribed vertices or arcs as can be seen from the following complexity statement. The proof is left to the reader as Exercise 10.2.

Proposition 10.1.2 *For general digraphs the following problems are equivalent from a computational point of view (that is, if one is polynomially solvable or $\mathcal{NP}$-complete, then so are each of the others).*

(P1) Given four distinct vertices u_1, u_2, v_1, v_2 in a digraph D. Decide whether or not D has disjoint paths connecting u_1 to v_1 and u_2 to v_2. We call this the 2-LINKAGE PROBLEM.
(P2) Given two distinct arcs e_1, e_2 in a digraph D. Does D have a cycle through e_1 and e_2?
(P3) Given two distinct vertices u and v in a digraph D. Does D have a cycle through u and v?
(P4) Given two distinct vertices u and v in a digraph D. Does D have disjoint cycles C_x, C_y such that $x \in C_x$ and $y \in C_y$?
(P5) Given three distinct vertices x, y, z. Does D have an (x, z)-path which also contains the vertex y? □

We prove in Theorem 10.2.1 that the 2-linkage problem is $\mathcal{NP}$-complete. Hence it follows from Proposition 10.1.2 that all the problems mentioned in Proposition 10.1.2 are $\mathcal{NP}$-complete.

It is interesting to note that although problems (P1)-(P5) are all very hard for general digraphs, the difficulty of these problems may vary considerably for some classes of digraphs. For instance, problem (P3) is trivial for locally semicomplete digraphs since such a cycle exists if and only if x and y are in the same strong component of D. Problem (P4) is also easy for semicomplete digraphs, since such cycles exist if and only if there exist disjoint 3-cycles C, C' one containing x and the other containing y (Exercise 10.16). However, problems (P1) and (P2) are considerably more difficult to prove polynomial, even for tournaments (see Theorem 10.5.12). Note that (P2) and also (P5) may be considered as special cases of (P1) if we drop the requirement that the vertices must be distinct in (P1).

The k-LINKAGE PROBLEM is the following straightforward generalization of the 2-linkage problem. Given a digraph D and distinct vertices

$x_1, x_2, \ldots, x_k,\ y_1, y_2, \ldots,\ y_k$. Does D have a collection of disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path, for every $i \in [k]$?

10.2 The Complexity of the *k*-Linkage Problem

We start with the following result by Fortune, Hopcroft and Wyllie showing that already for $k = 2$ the k-linkage problem is very difficult for general digraphs.

Theorem 10.2.1 *[332] The 2-linkage problem is $\mathcal{NP}$-complete.*

Since this theorem is very important and the gadget[2] construction used in the proof is quite illustrative, we give the proof in detail below. We follow the proof in [332].

First we need a lemma whose proof is left as Exercise 10.4.

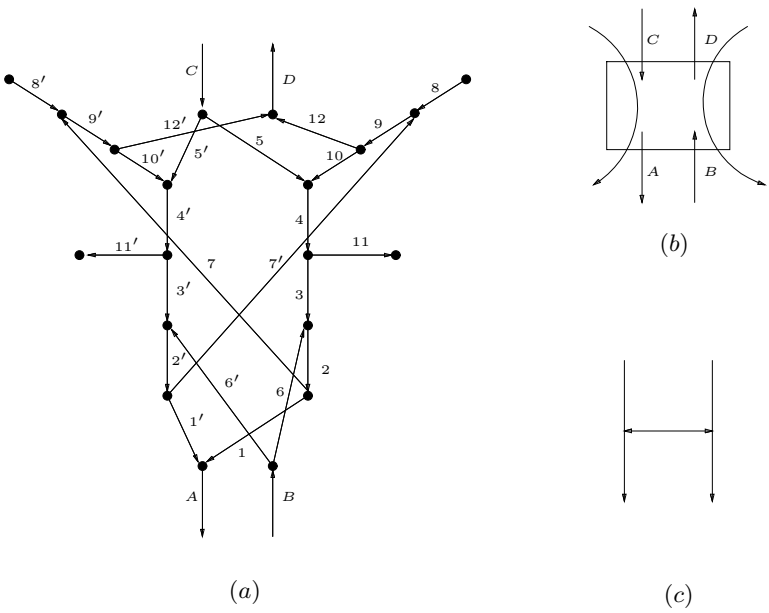


Figure 10.1 Part (a) shows a switch S. Parts (b) and (c) show schematic pictures of a switch ([332, Fig. 1]). In (c) the two vertical arcs correspond to the paths (8,9,10,4,11), respectively, (8',9',10',4',11'). Note that for convenience, we label the arcs, rather than the vertices, in this Figure.

[2] Quite often $\mathcal{NP}$-completeness proofs are constructed by piecing together certain **gadgets** about which one can prove certain properties. Based on these properties one then shows that the whole construction has the desired properties. For other instances of this technique, see e.g. Chapters 6 and 16.

Lemma 10.2.2 *[332] Consider the digraph S shown in Figure 10.1(a). Suppose there are two disjoint paths P, Q passing through S such that P leaves S at A and Q enters S at B. Then P must enter S at C and Q must leave S at D. Furthermore, there exists exactly one more path R passing through S which is disjoint from P, Q and this is either*

$$(8, 9, 10, 4, 11) \qquad or \qquad (8', 9', 10', 4', 11'),$$

depending on the actual routing of P. □

The digraph S in Figure 10.1 is called a **switch**. We can stack arbitrarily many switches on top of each other and still have the conclusion on Lemma 10.2.2 holding for each switch. The way we stack is simply by identifying the C and D arcs of one switch with the A and B arcs of the next (see Figure 10.2). A switch can be represented schematically as in Figure 10.1(c), or, when we want to indicate stacking of switches, as in Figure 10.1(b).

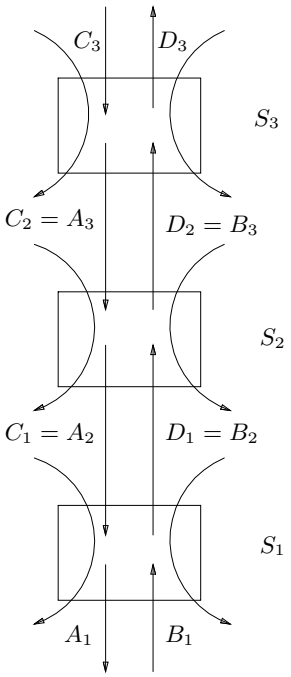


Figure 10.2 Stacking three switches on top of each other.

Proof of Theorem 10.2.1: The reduction is from 3-SAT (see the definition in Section 17.5). Let $\mathcal{F} = C_1 * C_2 * \ldots * C_r$ be an instance of 3-SAT with variables $x_1, x_2, \ldots, x_k$. For each variable x_i we let H_i be the digraph consisting

of two internally disjoint (u, v)-paths of length r (the number of clauses in $\mathcal{F}$). We associate one of these paths with the literal x_i and the other with the literal $\overline{x}_i$. We are now ready to explain the construction of the digraph $D[\mathcal{F}]$ and show that it contains disjoint (u_1, v_1)-, (u_2, v_2)-paths if and only if $\mathcal{F}$ is satisfiable.

See Figure 10.3. We form a chain $H_1 \rightarrow H_2 \rightarrow \ldots \rightarrow H_k$ on the subdigraphs corresponding to each variable (see the middle of the figure, H_i corresponds to the variable x_i). With each clause C_i we associate three switches, one for each literal it contains. The left paths of these switches (that is, the paths in the left-hand part of the figure) all start at the vertex n_{i-1} and end at n_i. The right path of each switch is substituted for a (private) arc of H_i such that the arc is taken from the path which corresponds to x_i if the literal is x_i and from the path which corresponds to $\overline{x}_i$ if the literal is $\overline{x}_i$. The substitution is shown for the clause $C_i = x_1 + \overline{x}_2 + x_5$ in the figure. By the choice of the lengths of the paths in H_i we can make this substitution so that different arcs in H_i are substituted by different switches corresponding to several clauses, all of which contain the literal x_i or $\overline{x}_i$. The switches corresponding to the clause C_i are denoted $S_{i,1}, S_{i,2}, S_{i,3}$. We stack these switches in the order $S_{1,1}S_{1,2}S_{1,3} \ldots S_{r,1}S_{r,2}S_{r,3}$ as shown in the right part of the figure. A two-way arc between a clause and some H_j (shown only for C_i) indicates a switch that is substituted for these arcs[3]. Finally, we join the D arc of the switch $S_{r,3}$ to the vertex z_1 of H_1, add an arc from w_k in H_k to n_0 and choose vertices u_1, u_2, v_1, v_2 as shown (that is, u_2 is the tail of the C arc for $S_{r,3}$, u_1 is the tail of the B arc of $S_{1,1}$ and v_2 is the head of the A arc of $S_{1,1}$). This completes the description of $D[\mathcal{F}]$.

We claim that $D[\mathcal{F}]$ contains disjoint (u_1, v_1)-, (u_2, v_2)-paths if and only if $\mathcal{F}$ is satisfiable. Suppose first that $D[\mathcal{F}]$ has disjoint (u_1, v_1)-, (u_2, v_2)-paths P, Q. It follows from the definition of $D[\mathcal{F}]$ that the paths P and Q will use all the arcs that go between two switches (i.e., those arcs that are explicitly shown in the right-hand side of Figure 10.3). Hence, by Lemma 10.2.2, after removing the arcs of Q and the arcs of P from u_1 to the first vertex z_1 of H_1, the only remaining way to pass through a switch $S_{i,j}$ is to use either the right path or the left path of $S_{i,j}$ but not both! By the construction of $D[\mathcal{F}]$, P must traverse the subdigraphs corresponding to the variables in the order $H_1, H_2, \ldots, H_k$ and each time P uses precisely one of the two paths in H_i (recall again that some of the arcs in H_i in Figure 10.3 correspond to the right path of some switch). Let T be the truth assignment which sets $x_i := 1$ if P uses the path corresponding to $\overline{x}_i$ and let $x_i := 0$ in the opposite case. We show that this is a satisfying truth assignment for $\mathcal{F}$.

It follows from the construction of $D[\mathcal{F}]$ and the remark above on arcs used by Q and the first part of P from u_1 to H_1 that the path P contains all the vertices $n_0, n_1, \ldots, n_r$ in that order. Since each of the paths from n_j to

[3] Note that this is the same switch which is shown in the right-hand side of the figure!

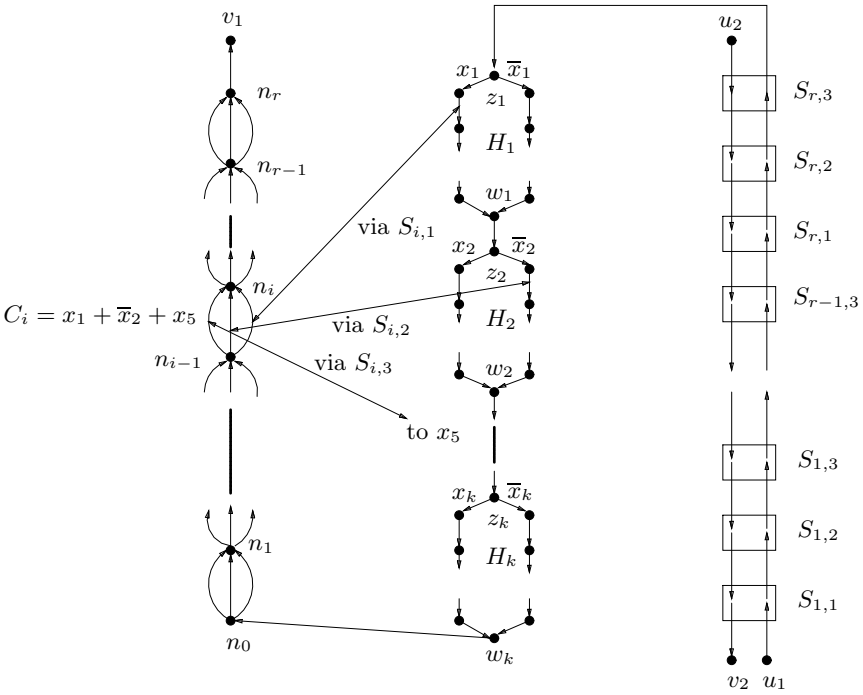


Figure 10.3 A schematic picture of the digraph $D[\mathcal{F}]$.

n_{j+1} is part of a switch for every $j = 0, 1, \ldots r-1$, we must use the left path of precisely one of these switches to go from n_j to n_{j+1}. By Lemma 10.2.2, every time we use a left path of a switch, the right path cannot also be used. From this we see that for each clause C_j, $j \in [r]$, it must be the case that at least one of the literals y (in particular the one whose left path we could use) of C_j becomes satisfied by our truth assignment. This follows because P must use the path corresponding to $\overline{y}$ in the middle. Thus we have shown that $\mathcal{F}$ is satisfiable.

Suppose now that T' is a satisfying truth assignment for $\mathcal{F}$. Then for every variable x_i which is true (false) we can use the subpath corresponding to $\overline{x}_i$ (x_i) in H_i. For each clause C_j we can fix one literal which is true and use the left path of the switch that corresponds to that literal (that path cannot be blocked by the way we chose subpaths inside the H_i's). By Lemma 10.2.2, we can find disjoint paths P, Q such that P starts in u_1 and ends in the initial vertex z_1 of H_1 and Q is a (u_2, v_2)-path in the right part of $D[\mathcal{F}]$. Furthermore, by the same lemma, after removing the vertices of P and Q, we still have the desired paths corresponding to each literal available. This shows that we can route the disjoint (u_1, v_1)-, (u_2, v_2)-paths in $D[\mathcal{F}]$. □

The digraph $D[\mathcal{F}]$ above is not strongly connected and one may ask whether the problem becomes easier if we require high vertex-strong connectivity. However, using Theorem 10.2.1, Thomassen [867] proved that the 2-linkage problem remains $\mathcal{NP}$-complete even for highly connected digraphs.

Johnson, Robertson, Seymour and Thomas [573] proved the following theorem for directed tree-width. By Lemma 2.13.9, this theorem holds also for directed path-width and DAG-width (see Section 2.13 for definitions of directed width parameters).

Theorem 10.2.3 *[573] Let k be a fixed positive integer. The k-linkage problem is polynomial-time solvable for digraphs of bounded directed tree-width (DAG-width, directed path-width, respectively).* □

Lynch proved that for undirected graphs the k-linkage problem is $\mathcal{NP}$-complete when k is part of the input [662]. The case $k = 2$ was proved to be polynomially solvable by Seymour [808], Shiloach [819] and Thomassen [855] and a complete characterization was obtained by Seymour [808] and Thomassen [855]. The results in [808, 855] (see also Jung's paper [580]) imply that every 6-connected undirected graph is 2-linked (see also the remark at the end of Section 10.6). For fixed $k \geq 3$ the k-linkage problem is also polynomially solvable [785]. This is just one of many important consequences of the deep work of Robertson and Seymour on Graph Minors. The interesting thing is that [785] only proves the existence of an $O(n^3)$ algorithm for fixed k (the constant depending heavily on k). As far as we know, no actual algorithm has ever been published, even in the case $k = 3$.

The following result, due to Thomassen, shows that for directed graphs the situation is quite different from the undirected case. Namely, there is no degree of vertex-strong connectivity which will guarantee a directed graph to be 2-linked.

Theorem 10.2.4 *[867] For every natural number k there exists an infinite family of k-strong and non-2-linked digraphs D_k.* □

In fact, Thomassen proved that even for the special case of cycles through two fixed vertices (Problem (P3) of Proposition 10.1.2) no degree of vertex-strong connectivity suffices to guarantee such a cycle. Recall that a digraph $D = (V, A)$ is 2-cyclic if it has a cycle containing x, y for every choice of distinct vertices $x, y \in V$.

Theorem 10.2.5 *[867] For every natural number k there exists an infinite family of k-strong digraphs D'_k which are not 2-cyclic.* □

10.3 Sufficient Conditions for a Digraph to Be k-Linked

In this section we briefly discuss some sufficient conditions for a digraph to be k-linked for some (prescribed) k. It is easy to see that the complete

digraph $\overleftrightarrow{K}_n$ is k-linked for all $k \leq \lfloor n/2 \rfloor$. The next result due to Manoussakis shows that digraphs which are close to being complete are k-linked whenever $|V(D)| \geq 2k$. The proof is left as Exercise 10.6.

Theorem 10.3.1 *[680] Let $D = (V, A)$ be a digraph of order n and let k be an integer such that $n \geq 2k \geq 2$. If $|A| \geq n(n-2) + 2k$, then D is k-linked.* □

The proof of Theorem 10.3.1 in [680] is based on the following lemma.

Lemma 10.3.2 *[680] If $D - x$ is k-linked for some vertex $x \in V$ which satisfies $d^+(x), d^-(x) \geq 2k - 1$, then D is k-linked.*

Proof: Exercise 10.8. □

The requirement on the number of arcs in Theorem 10.3.1 is very strong and hence the result is not very useful. However, Manoussakis showed by an example that the number of arcs in Theorem 10.3.1 is best possible [680].

The next result, due to Heydemann and Sotteau, shows that for 2-linkages one can also get a sufficient condition in terms of $\delta^0(D)$. The proof is easy and is left as Exercise 10.7. See also Theorem 10.3.4 below.

Theorem 10.3.3 *[525] If a digraph D satisfies $\delta^0(D) \geq n/2 + 1$, then D is 2-linked.* □

The condition above is still quite restrictive and one would expect a stronger result to hold. Examples from [525] show that we cannot weaken the degree condition. However, we can strengthen the result in the following way.

Theorem 10.3.4 *If a digraph D satisfies $\delta^0(D) \geq n/2 + 1$, then for every choice of distinct vertices $x, y, u, v \in V$, D contains internally disjoint paths P, Q such that P is an (x, y)-path, Q is a (u, v)-path and $V(P) \cup V(Q) = V$.*

Proof: Let $X = V - \{x, y, u, v\}$ and construct D' from $D - \{x, y, u, v\}$ by adding two new vertices p and q such that

$$N^-_{D'}(p) = N^-_D(v) \cap X, N^+_{D'}(p) = N^+_D(x) \cap X,$$
$$N^-_{D'}(q) = N^-_D(y) \cap X, N^+_{D'}(q) = N^+_D(u) \cap X.$$

It is easy to see that for every $w \in V - \{x, y, u, v\}$, $d^-_{D'}(w) \geq d^-_D(w) - 2$ and $d^+_{D'}(w) \geq d^+_D(w) - 2$. Hence the resulting digraph D' which has $n' = n - 2$ vertices satisfies $\delta^0(D') \geq n'/2$. By Corollary 6.4.3, D' has a hamiltonian cycle C. Let p^+, q^+ (p^-, q^-) denote the successors (predecessors) of p, q on C. Then $xC[p^+, q^-]y$ and $uC[q^+, p^-]v$ are the desired paths which cover V. □

Manoussakis extended Theorem 10.3.3 to 3-linkages.

Theorem 10.3.5 *[680] If a digraph D has $n \geq 9$ vertices and $\delta^0(D) \geq n/2 + 2$, then D is 3-linked.* □

Based on Theorems 10.3.3 and 10.3.5, Manoussakis posed the following problem.

Problem 10.3.6 *[680] Determine the minimum function $f(n,k)$ such that every digraph D on n vertices which satisfies $\delta^0(D) \geq f(n,k)$ is k-linked.*

Note that $f(n,k) \leq n-1$ for $n \geq 2k$, since the complete digraph on $n \geq 2k$ vertices is k-linked. According to Manoussakis [680], Hurkens proved that $f(n,4) = n/2+3$ when $n \geq 13$ and Manoussakis mentioned in [680] that perhaps $f(n,k) \leq n/2 + k - 1$ holds for $n \geq 4k-3$. This was confirmed for n sufficiently large by Kühn and Osthus [628].

Theorem 10.3.7 *[628] Let $k \geq 2$ be an integer. Every digraph D of order $n \geq 400k^3$ which satisfies $\delta^0(D) \geq n/2 + k - 1$ is k-linked.* □

Kühn and Osthus [628] also prove that the bound suggested by Manoussakis would be best possible for every k.

Proposition 10.3.8 *[628] For every integer $k \geq 2$ and every $n \geq 2k$ there exists a digraph D on n vertices with $\delta^0(D) \geq \lceil n/2 \rceil + k - 2$ which is not k-linked.* □

By Proposition 10.1.1, the next result immediately implies Theorem 10.3.7.

Theorem 10.3.9 *[628] Let $k \geq 2$ be an integer. Every digraph D of order $n \geq 200k^3$ which satisfies $\delta^0(D) \geq (n+k)/2 - 1$ is k-cyclic.* □

A digraph $D = (V, A)$ is **k-cyclic hamiltonian** if for every choice of distinct vertices $x_1, x_2, \ldots, x_k \in V$ there is a hamiltonian cycle in D which visits $x_1, x_2, \ldots, x_k$ in that order. In [630] Kühn, Osthus and Young use Theorem 10.3.9 to prove the following.

Theorem 10.3.10 *[630] For every integer $k \geq 3$ there exists an integer $n_0(k)$ such that every digraph D on at least $n_0(k)$ vertices and minimum semi-degree $\delta^0(D) \geq \lceil (n+k)/2 \rceil - 1$ is k-cyclic Hamiltonian.* □

Clearly this result also implies a result on hamiltonian cycles containing k prescribed arcs. We leave the details to the reader.

Let us conclude this section with a result in connection with problem (P3) of Proposition 10.1.2. It is easy to see that if a digraph is 2-linked, then it is also 2-arc-cyclic and hence 2-cyclic. Heydemann and Sotteau proved that if we only want a digraph to be 2-cyclic, then it is possible to weaken the condition in Theorem 10.3.1 somewhat.

Theorem 10.3.11 *[525] Every strong digraph $D = (V, A)$ with $\delta^0(D) \geq 2$ and $|A| \geq n^2 - 5n + 15$ is 2-cyclic.* □

10.4 The k-Linkage Problem for Acyclic Digraphs

When the digraph considered is acyclic, there is enough structure to allow an efficient solution of the k-linkage problem for every fixed k. Perl and Shiloach [746] proved that the 2-linkage problem is polynomially solvable for acyclic digraphs. In their elegant proof they showed how to reduce the 2-linkage problem for a given acyclic digraph to a simple path finding problem in another digraph. Fortune, Hopcroft and Wyllie extended Perl and Shiloach's result to arbitrary k. The proof of this result below is an extension of the proof by Perl and Shiloach (see also Thomassen's survey [865]).

Theorem 10.4.1 *[332] For each fixed k, the k-linkage problem is polynomially solvable for acyclic digraphs.*

Proof: Let $D = (V, A)$ be acyclic and let $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ be distinct vertices of D for which we wish to find a k-linkage from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. We may assume that $d^-_D(x_i) = d^+_D(y_i) = 0$ for all $i \in [k]$, since such arcs play no role in the problem and can therefore be deleted.

Form a new digraph $D' = (V', A')$ whose vertex set is the set of all k-tuples of distinct vertices of V. For any such k-tuple $(v_1, v_2, \ldots, v_k)$ there is at least one vertex, say v_r, which cannot be reached by any of the other v_i by a path in D. (Here we used that D is acyclic.) For each out-neighbour w of v_r such that $w \notin \{v_1, v_2, \ldots, v_k\}$, we let A' contain an arc from $(v_1, v_2, \ldots, v_{r-1}, v_r, v_{r+1}, \ldots, v_k)$ to $(v_1, v_2, \ldots, v_{r-1}, w, v_{r+1}, \ldots, v_k)$. Only arcs as those described above are in A'.

We claim that D' has a directed path from the vertex $(x_1, x_2, \ldots, x_k)$ to the vertex $(y_1, y_2, \ldots, y_k)$ if and only if D contains disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path for each $i \in [k]$.

Suppose first that D' has a path P from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. By definition, every arc of P corresponds to one arc in D. Hence we get a collection of paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path for each $i \in [k]$ by letting P_i contain those arcs that correspond to a shift in the ith vertex of a k-tuple. Suppose two of these paths, P_i, P_j, are not disjoint. Then it follows from the assumption that $d^-_D(x_i) = d^+_D(y_i) = 0$ for all $i \in [k]$ and the definition of D' that there is some vertex $u \in V - \{x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k\}$ such that $u \in V(P_i) \cap V(P_j)$. Let w (z) be the predecessor of u on P_i (P_j). We may assume without loss of generality that the arc on P corresponding to wu is used before that corresponding to zu. This means that at the time we change from w to u in the ith coordinate, the jth coordinate corresponds to a vertex z' which can reach u in D (through z). Now it follows from the definition of the arcs in A' that we could not have changed the ith coordinate again before we have used the arc corresponding to zu in D'. However, that would lead to a k-tuple which contains two copies of the same vertex u from D, contradicting the definition of D'. Hence P_i and P_j must be disjoint.

Suppose now that D contains disjoint paths $Q_1, Q_2, \ldots, Q_k$ such that Q_i is an (x_i, y_i)-path for all $i \in [k]$. Then we can construct a path from

$(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D' as follows. Start with the tuple $(x_1, x_2, \ldots, x_k)$. At any time we choose a coordinate j of the current k-tuple $(z_1, z_2, \ldots, z_k)$ such that the vertex z_j is not in $\{y_1, y_2, \ldots, y_k\}$ and z_j cannot be reached in D by any other vertex from the tuple. Note that such a vertex exists since D is acyclic and $d^+(y_i) = 0$ for all $i \in [k]$. It is easy to show by induction that we will always have $z_j \in V(Q_j)$. Now we use the arc $z_j w$ corresponding to the arc out of z_j on Q_j and change the jth coordinate from z_j to w. It follows from the fact that $Q_1, \ldots, Q_k$ are disjoint that this will produce a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D'.

Given any instance $(D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k)$ we can produce the digraph D' in time $O(k! n^{k+2})$ by forming all possible k-tuples and deciding which arcs to add based on the definition of D'. Then we can decide the existence of a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in polynomial time using BFS in D'. This proves that the k-linkage problem is polynomial for each fixed k. □

Note that we don't actually have to construct D' in advance. It suffices to introduce the vertices and arcs when they become relevant for the search for a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D'.

It is not difficult to see that we can also use the approach above to find the cheapest collection of k disjoint paths where the ith path is an (x_i, y_i)-path in a given acyclic digraph with non-negative costs on the arcs. Here the goal is to minimize the total cost of the arcs used by the paths (see Exercise 10.11).

Suppose that D is an acyclic graph and v is a vertex of in-degree 1. Let u be the unique in-neighbour of v. Then the digraph $D' = D//uv$ which we obtain by path-contracting the arc uv is also acyclic. Furthermore, contracting such an arc can have no effect on the existence of a certain linkage in the digraph since only one path in such a linkage may enter the vertex v. This shows that we may assume that all vertices except the terminals have in- and out-degree at least 2 when considering the 2-linkage problem (and more generally the k-linkage problem) for acyclic graphs. Furthermore we may assume that no arc enters x_i and no arc leaves y_i, $i = 1, 2$.

It is also easy to see that, given any acyclic digraph D with distinct vertices x_1, x_2, y_1, y_2, in polynomial time, we can either decide the existence of disjoint (x_1, y_1)-, (x_2, y_2)-paths, or obtain a new reduced digraph D^* such that $d^-_{D^*}(x_1) = d^-_{D^*}(x_2) = d^+_{D^*}(y_1) = d^+_{D^*}(y_2) = 0$, every other vertex has in- and out-degree at least 2 in D^* and D^* has the desired paths if and only if D has such paths. Hence, from a computational point of view, the following result due to Thomassen completely solves the 2-linkage problem for acyclic digraphs.

Theorem 10.4.2 *[862] Let D be an acyclic digraph on at least five vertices with vertices x_1, x_2, y_1, y_2 such that $d^-(x_1) = d^-(x_2) = 0$, $d^+(y_1) = d^+(y_2) = 0$ and every other vertex has in- and out-degree at least 2. Suppose D does not contain disjoint (x_1, y_1)-, (x_2, y_2)-paths. Let H denote the digraph one obtains from D by adding two new vertices x_0, y_0 and the arcs $x_0x_1, x_0x_2, y_1y_0, y_2y_0, x_1y_2, x_2y_1$. Then H can be drawn in the plane such that the outer cycle is formed by the two paths $x_0x_1y_2y_0, x_0x_2y_1y_0$ and every other facial cycle[4] is the union of two directed paths in H (see Figure 10.4).* □

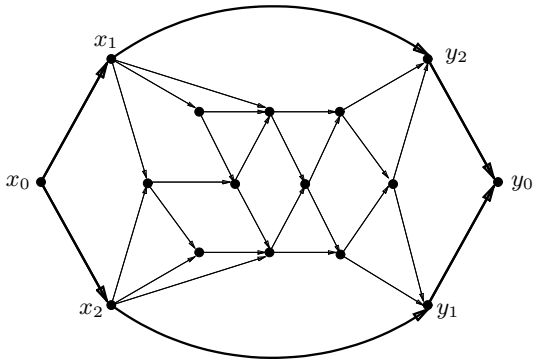


Figure 10.4 The digraph H obtained from the acyclic digraph D by adding x_0, y_0 and arcs $x_0x_1, x_0x_2, y_1y_0,\ y_2y_0, x_1y_2, x_2y_1$ (shown as bold arcs) as described in Theorem 10.4.2.

Theorem 10.4.2 was generalized by Metzlar [697]. The following interesting connection between the 2-linkage problem for undirected graphs and the 2-linkage problem for acyclic digraphs is a corollary of Theorem 10.4.2.

Corollary 10.4.3 *[862] Let $D = (V, A)$ be an acyclic digraph and suppose that the vertices x_1, x_2, y_1, y_2 are all distinct and satisfy that $d^-(x_i) = d^+(y_i) = 0$ for $i = 1, 2$ and that all other vertices of D have in- and out-degree at least 2. Then D contains disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if $UG(D)$ contains such paths.* □

Thomassen [862] mentioned that it would be interesting to have a direct proof of Corollary 10.4.3. Such a proof was given by Lucchesi and Giglio in [660]. In that paper the connection between the 2-linkage problem for acyclic digraphs and the 2-linkage problem for undirected graphs was studied. It was shown that there is a very close connection between the two problems.

The example in Figure 10.5 shows that Corollary 10.4.3 has no analogue when $k > 2$.

[4] A cycle C in a plane graph G is **facial** with respect to a planar drawing of G if C is the boundary of some face.

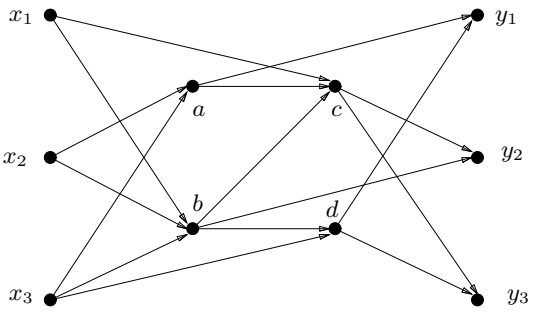


Figure 10.5 An acyclic digraph D in which every non-special vertex has in- and out-degree at least 2. There does not exist disjoint paths P_1, P_2, P_3 such that P_i is an (x_i, y_i)-path, $i = 1, 2, 3$. However, $UG(D)$ has such paths.

Theorem 10.4.4 *The weak k-linkage problem and k-linkage problem are W[1]-hard even for acyclic digraphs (k is the parameter).* □

The fact that the weak k-linkage problem for acyclic digraphs is W[1]-hard is proved in [824]. The k-linkage part of the theorem follows from the weak k-linkage part and the transformation from a digraph D to its line digraph $L(D)$ (see also [425]).

10.5 Linkages in (Generalizations of) Tournaments

We now turn to linkage problems for tournaments and their generalizations. It turns out that for semicomplete digraphs enough structure is present to allow a polynomial algorithm for the 2-linkage problem (Theorem 10.5.12). We show in Subsection 10.5.3 that this algorithm can be used as a subroutine in a polynomial algorithm for the 2-linkage problem for a large super class of the semicomplete digraphs.

We start out with some sufficient conditions in terms of the degree of (local) strong connectivity.

10.5.1 Sufficient Conditions in Terms of (Local-)Connectivity

The following proposition was proved by Thomassen [859] in the case when D is a tournament. By inspection of the proof in [859] one sees that the only place there where it is used that one is dealing with a tournament, rather than an arbitrary digraph, is to be sure that there is an arc between every successor of x and every predecessor of y on the paths $P_1, \ldots, P_p$ below. Hence we can state and prove Thomassen's result in the following much stronger form:

Proposition 10.5.1 *[74, 859] Let D be a digraph and x, y, u, v distinct vertices of D such that $\kappa(u, v) \geq q + 2$ and $P_1, \ldots, P_p$ are internally disjoint (x, y)-paths such that the subdigraph $D\langle V(P_1) \cup \ldots \cup V(P_p)\rangle$ has no (x, y)-path of length less than or equal to 3 and such that the successor of x on P_i is adjacent to the predecessor of y on P_j for all $i, j \in [p]$. Then D has q internally disjoint (u, v)-paths, the union of which intersects at most $2q$ of the paths $P_1, \ldots, P_p$.*

Proof: We may assume that $p \geq 2q+1$, since otherwise the claim is trivially true. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_q\}$ be internally disjoint (u, v)-paths in $D - \{x, y\}$. We define two collections of subpaths of the paths in $\mathcal{Q}$ as follows (in Exercise 10.17 the reader is asked to describe an algorithm for constructing such collections starting from $\mathcal{Q}$).

Let $Q'_1, Q'_2, \ldots, Q'_q$ be chosen such that either $Q'_i = Q_i$ or $Q'_i = Q[u, z]$ for some vertex $z \in V(P_j)$ where $j \in [p]$ and $P_j[z, y]$ has only the vertex z in common with $U = V(Q'_1) \cup \ldots \cup V(Q'_q)$. We also assume that $|U|$ is minimum subject to the conditions above. If some path P_r contains a vertex w from U and $P_r[w, y]$ contains no vertices from $U - w$, then the minimality of U implies that one of the paths $Q'_1, Q'_2, \ldots, Q'_q$ terminates in w. This implies that the collection $Q'_1, Q'_2, \ldots, Q'_q$ intersects at most q of the paths $P_1, P_2, \ldots, P_p$.

Analogously we can define a collection $Q''_1, Q''_2, \ldots, Q''_q$ where Q''_i is either Q_i or $Q''_i = Q_i[w, v]$ for a vertex w on some P_k satisfying that $P_k[x, w]$ contains only the vertex w from $V(Q''_1) \cup \ldots \cup V(Q''_q)$ and such that $Q''_1, Q''_2, \ldots, Q''_q$ intersect at most q of the paths $P_1, P_2, \ldots, P_p$.

Now we construct the desired paths as follows. For each $i \in [q]$, if $Q'_i = Q_i$ or $Q''_i = Q_i$, then let $R_i := Q_i$. Otherwise let z be the terminal vertex of Q'_i, let w be the initial vertex of Q''_i and let r, j be chosen such that $z \in V(P_j), w \in V(P_r)$. Let x' (y') be the successor (predecessor) of x (y) on P_r (P_j). By the assumption that D contains no (x, y)-path of length 3 and that every successor of x is adjacent to every predecessor of y on the paths $P_1, \ldots, P_p$, we get that $y'x' \in A$. Let $R_i := Q'_i P_j[z, y'] P_r[x', w] Q''_i$ (see Figure 10.6).

Now $R_1, R_2, \ldots, R_q$ are internally disjoint (u, v)-paths and by construction they contain no more than $2q$ vertices from the paths $P_1, P_2, \ldots, P_p$. □

Our proof above is constructive and can easily be turned into a fast algorithm for finding the desired collection of paths (Exercise 10.18). The following result by Thomassen is an easy corollary.

Corollary 10.5.2 *[859] Every 5-strong semicomplete digraph is 2-linked.*

Proof: Let D be a 5-strong semicomplete digraph and let x_1, x_2, y_1, y_2 be arbitrary distinct vertices of D. If $D - \{x_{3-i}, y_{3-i}\}$ has an (x_i, y_i)-path P of length at most 3 for $i = 1$ or $i = 2$, then $D - P$ is strong and hence contains an (x_{3-i}, y_{3-i})-path. Hence we may assume that every (x_i, y_i)-path in $D - \{x_{3-i}, y_{3-i}\}$ has length at least 4 for $i = 1, 2$.

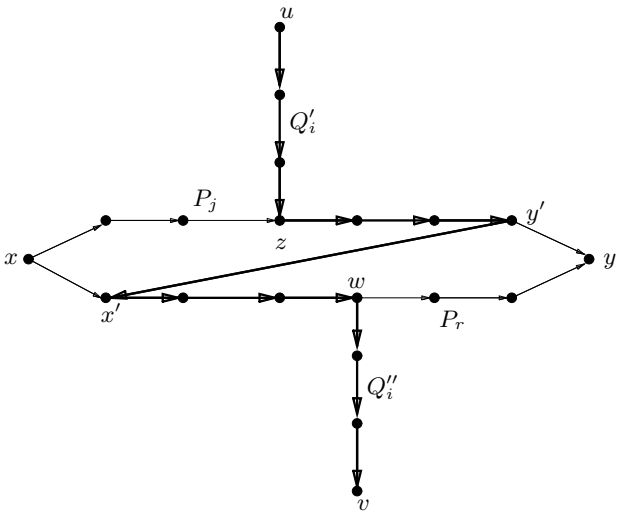


Figure 10.6 How to obtain R_i from Q_i', Q_i'', P_j and P_r. The bold arcs indicate the resulting (u, v)-path.

Let P_1, P_2, P_3 be internally disjoint (x_1, y_1)-paths in $D - \{x_2, y_2\}$. Then D and these paths satisfy the assumption of Theorem 10.5.1 for $q = 1$ and it follows that D has an (x_2, y_2)-path which intersects at most two of the paths P_1, P_2, P_3. Since x_1, x_2, y_1, y_2 were chosen arbitrarily, it follows that D is 2-linked. □

Bang-Jensen [65] constructed the 4-strong non-2-linked semicomplete digraph in Figure 10.7, showing that 5-strong connectivity is best possible for general semicomplete digraphs. We leave it to the reader to check that one can generalize this example to an infinite family of 4-strong semicomplete digraphs none of which is 2-linked.

We now turn our attention to special classes of generalizations of tournaments. The first lemma shows that for the class of round decomposable locally semicomplete digraphs one can improve the bound from Corollary 10.5.2. The proof is left as Exercise 10.22.

Lemma 10.5.3 *[74] For each natural number k, every $(3k-2)$-strong round decomposable locally semicomplete digraph is k-linked.* □

In order to get a result on k-linkages for locally semicomplete digraphs that are not round decomposable we use the following lemma which allows us to apply Proposition 10.5.1. Recall that by Exercise 2.34, $\alpha(D) \leq 2$ if D is locally semicomplete but not round decomposable.

Lemma 10.5.4 *[74] Let x and y be distinct vertices in a locally semicomplete digraph D such that $\alpha(D) \leq 2$ and let $P_1, \ldots, P_p$ be internally disjoint (x, y)-*

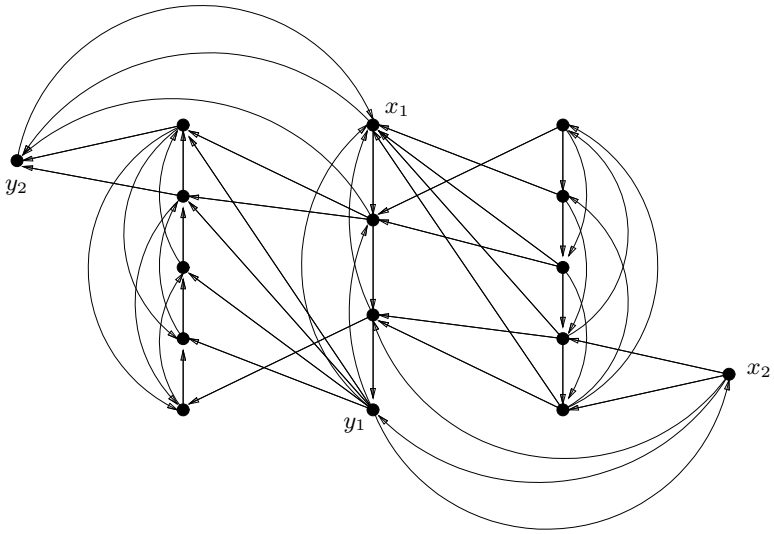


Figure 10.7 A 4-strong non-2-linked semicomplete digraph T. All arcs not shown go from left to right and $x_1y_2x_1, x_2y_1x_2$ are the only 2-cycles in T. There is no pair of disjoint (x_1, y_1)-,(x_2, y_2)-paths in T. The tournament which results from T by deleting the arcs y_2x_1 and y_1x_2 is also 4-strong.

paths such that the locally semicomplete digraph $D' = D\langle V(P_1) \cup \ldots \cup V(P_p)\rangle$ *has no* (x, y)*-path of length less than 6. Then for all* $1 \leq i, j \leq p$*, the predecessor* u *of* y *on* P_i *dominates the successor* v *of* x *on* P_j*.*

Proof: We may assume that each P_i is a minimal (x, y)-path. Suppose there exist i and j such that the predecessor u of y on P_i is not adjacent to the successor v of x on P_j. Note that the assumption of the lemma and Exercise 10.20 implies that $y \rightarrow x$. Therefore D' is strong and we conclude from Exercise 10.20 (applied to u, v) that D' contains an (x, y)-path of length at most 5, contradicting the assumption. Hence $u \rightarrow v$ must hold. □

The following theorem by Bang-Jensen gives a sufficient condition for the existence of a specified k-linkage in a locally semicomplete digraph which is not round decomposable in terms of local connectivities. It generalizes a result by Thomassen for tournaments [859]. Bang-Jensen also proved an analogous result for quasi-transitive digraphs, see [74] for details.

Theorem 10.5.5 *[74] There exists, for each natural number* k*, a natural number* $f(k)$ *such that the following holds. If* D *is a locally semicomplete digraph with* $\alpha(D) \leq 2$ *and* $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ *are distinct vertices in* D *such that* $\kappa(x_i, y_i) \geq f(k)$ *for all* $i \in [k]$*, then* D *has disjoint paths* $P_1, P_2, \ldots, P_k$ *where* P_i *is an* (x_i, y_i)*-path for all* $i \in [k]$*.*

Proof: Let $f(1) = 1$ and $f(k) = 2(k-1)f(k-1) + 2k + 1$ for $k \geq 2$. We prove by induction on k that this choice works for f. This is clear for $k = 1$, so we proceed to the induction step assuming $k \geq 2$. Suppose that $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ are distinct vertices in a locally semicomplete digraph D for which $\alpha(D) \leq 2$ and assume that $\kappa(x_i, y_i) \geq 2(k-1)f(k-1) + 2k+1$ for all $i \in [k]$. We prove that $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$ has an (x_1, y_1)-path P_1 such that $\kappa_H(x_i, y_i) \geq f(k-1)$ for $i = 2, \ldots, k$, where $H = D - V(P_1)$. Then the result follows by induction. If $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$ has an (x_1, y_1)-path of length at most 5, then this can play the role of P_1, so assume that no such path exists. Let $Q_1, Q_2, \ldots, Q_{2(k-1)f(k-1)+1}$ be internally disjoint (x_1, y_1)-paths in $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$. We show that one of these can play the role of P_1. First note that by Lemma 10.5.4 and the remark above, we have that for all $1 \leq i, j \leq 2(k-1)f(k-1) + 1$ the predecessor of y_1 on Q_i dominates the successor of x_1 on Q_j. Hence, by Proposition 10.5.1, for each $i = 2, 3, \ldots, k$, there are internally disjoint (x_i, y_i)-paths $P_{1,i}, P_{2,i}, \ldots, P_{f(k-1),i}$ which together intersect at most $2f(k-1)$ of the paths $Q_1, Q_2, \ldots, Q_{2(k-1)f(k-1)+1}$. Hence there is at least one path Q_r which intersects none of $P_{j,i}$, $2 \leq i \leq k$, $1 \leq j \leq f(k-1)$. Thus we can use that Q_r as P_1. □

Combining Lemma 10.5.3, Theorem 10.5.5 and Theorem 2.10.15 we obtain the following result by Bang-Jensen (extending a similar result for semicomplete digraphs by Thomassen [859]). Here and below $f(k)$ is the function which is defined in the proof of Theorem 10.5.5.

Theorem 10.5.6 *[74] There exists, for each natural number k, a natural number $f(k)$ such that every $f(k)$-strong locally semicomplete digraph is k-linked.* □

Corollary 10.5.7 *[74] Every $f(k)$-strong locally semicomplete digraph is k-arc-cyclic.* □

The function $f(k)$ is probably far from best possible for Theorem 10.5.6 and Corollary 10.5.7. In particular, $f(2) = 7$, but, using Theorem 2.10.15, it should be possible to prove that the following holds.

Conjecture 10.5.8 *[74] Every 5-strong locally semicomplete digraph is 2-linked.*

10.5.2 The 2-Linkage Problem for Semicomplete Digraphs

In the proof of Corollary 10.5.2 we really only used that $\kappa_{T-\{x_i,y_i\}}(x_{3-i}, y_{3-i})$ was at least 3 for $i = 1, 2$ in order to ensure the existence of three internally disjoint (x_1, y_1)-paths in $D - \{x_2, y_2\}$ and then we applied Proposition 10.5.1. Bang-Jensen strengthened this sufficient condition as follows.

Theorem 10.5.9 *[65] Let T be a semicomplete digraph and let x_1, x_2, y_1, y_2 be distinct vertices of T. Suppose that*

$$\min\{\kappa_{T-\{x_2,y_2\}}(x_1,y_1), \kappa_{T-\{x_1,y_1\}}(x_2,y_2)\} \geq 2 \text{ and} \tag{10.1}$$

$$\max\{\kappa_{T-\{x_2,y_2\}}(x_1,y_1), \kappa_{T-\{x_1,y_1\}}(x_2,y_2)\} \geq 3, \tag{10.2}$$

then T has a pair of disjoint (x_1,y_1)-, (x_2,y_2)-paths. □

This is best possible with respect to local connectivities. The semicomplete digraph in Figure 10.7 shows that we cannot replace 3 by 2 above. However, see Theorem 10.5.13 for a special case where we can do this.

Bang-Jensen showed that for cycles through two arcs (the special case when $y_1 \to x_2$ and $y_2 \to x_1$), we can strengthen Corollary 10.5.2 in the case of tournaments. The digraph in Figure 10.7 shows that for semicomplete digraphs we cannot always weaken the connectivity requirement.

Theorem 10.5.10 *[65] Every 3-strong tournament and every 5-strong semicomplete digraph is 2-arc-cyclic.* □

It follows from the proof of Theorem 10.5.10 in [65] that for a fixed pair of arcs e, e' we can replace the connectivity requirement that D is 5-strong by $(5-i)$-strong provided that i of the arcs e, e' are not in a 2-cycle $(i = 1, 2)$.

Conjecture 10.5.11 *[74] Every 3-strong locally tournament digraph is 2-arc-cyclic.*

The example in Figure 10.7 indicates that finding a complete generalization of those semicomplete digraphs that do not have disjoint (x,y)-, (u,v)-paths for a given set of distinct vertices x, y, u, v may be very difficult. In the special case where we allow u and y to be equal, that is, we are seeking an (x,v)-path which passes through the vertex u (that is, the problem (P5) in Proposition 10.1.2), it is indeed possible to give a characterization. Such a characterization was given by Bang-Jensen in [67].

From the algorithmic point of view, the 2-linkage problem for semicomplete digraphs was solved by Bang-Jensen and Thomassen who proved the following result:

Theorem 10.5.12 *[118] The 2-linkage problem is solvable in time $O(n^5)$ for semicomplete digraphs.* □

The proof of this result in [118] is highly non-trivial. The basic approach is divide and conquer and several non-trivial results and steps are needed to make the algorithm work. We state the most important of these results below since it is of independent interest.

Recall from Section 7.3 that an (s,t)-separator S is trivial if t has in-degree zero, or s has out-degree zero in $D - S$. The following result, which complements Theorem 10.5.9, is very important for the proof of correctness

of the algorithm of Bang-Jensen and Thomassen, since it corresponds to a case where no problem reduction is possible (using the approach taken in the algorithm).

Theorem 10.5.13 *[118] Let x_1, x_2, y_1, y_2 be distinct vertices of a semicomplete digraph T, such that $\kappa_{T-\{x_i,y_i\}}(x_{3-i}, y_{3-i}) = 2$ for $i = 1, 2$. Suppose that all (x_i, y_i)-separators of size 2 in $T - \{x_{3-i}, y_{3-i}\}$ are trivial, for $i = 1, 2$. Then T has a pair of disjoint (x_1, y_1)-, (x_2, y_2)-paths. Furthermore such a pair of paths can be constructed in time $O(n^3)$.* □

Note that the semicomplete digraph in Figure 10.7 does not satisfy the assumption of Theorem 10.5.13 since the two non-labeled vertices in the middle form a non-trivial (x_2, y_2)-separator of size 2 in $T - \{x_1, y_1\}$.

10.5.3 The 2-Linkage Problem for Generalizations of Tournaments

Now we show that the 2-linkage problem can be solved in polynomial time for quite large classes of digraphs which can be obtained by starting from semicomplete digraphs and then performing certain substitutions. The algorithm we describe uses the polynomial algorithm from Theorem 10.5.12 for the case of semicomplete digraphs as a subroutine. The results in this section are due to Bang-Jensen [74].

Theorem 10.5.14 *[74] Let $D = F[S_1, S_2, \ldots, S_f]$ where F is a strong digraph on $f \geq 2$ vertices and each S_i is a digraph with n_i vertices and let x_1, x_2, y_1, y_2 be distinct vertices of D. There exist semicomplete digraphs $T_1, \ldots, T_f$ such that $V(T_i) = V(S_i)$ for all $i \in [f]$, and the digraph $D' = F[T_1, T_2, \ldots, T_f]$ has vertex-disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if D has such paths. Furthermore, given D and x_1, x_2, y_1, y_2, D' can be constructed in time $O(n^2)$, where n is the number of vertices of D.*

Proof: If D has the desired paths, then so does any digraph obtained from D by adding arcs. Hence if D has the desired paths, then trivially D' exists and can be constructed in time $O(n^2)$ once we know a pair of disjoint (x_1, y_1)-, (x_2, y_2)-paths.

If no S_i contains both of x_1, y_1 or both of x_2, y_2, then it is easy to see that D has the desired paths if and only if it has such paths which do not use an arc inside any S_j. Thus in this case we can add arcs arbitrarily inside each S_i to obtain a D' which satisfies the requirement.

Suppose next that some S_i contains all of the vertices x_1, x_2, y_1, y_2. If there is an (x_j, y_j)-path P in $S_i - \{x_{3-j}, y_{3-j}\}$, $j = 1$ or 2, then it follows from that fact that F is strong that D has the desired paths and we can find such a pair in time $O(n^2)$. Thus, by our initial remark, we may assume that there is no (x_j, y_j)-path P in $S_i - \{x_{3-j}, y_{3-j}\}$ for $j = 1, 2$. Now it is easy to see that D has the desired paths if and only if it has such paths which do not use an arc inside any S_j. Thus we can replace S_i by a tournament in which

x_1 and x_2 both have no out-neighbours in $S_i - \{x_1, x_2\}$ and every other S_k by an arbitrary tournament on the same vertex set. Clearly the digraph D' obtained in this way satisfies the requirement.

Suppose now without loss of generality that $x_1, y_1 \in V(S_j)$ for some j but $x_2 \notin V(S_j)$. Suppose first that $y_2 \in V(S_j)$. If there is no (x_1, y_1)-path in $S_j - y_2$, then D has the desired paths if and only if it has such paths which do not use an arc inside any S_i and we can construct D' by adding arcs in S_j in such a way that no (x_1, y_1)-path avoiding y_2 is created (that is, y_2 will still separate x_1 from y_1 in $D'\langle V(S_j)\rangle$) and arbitrary arcs in every other S_i. On the other hand, if $S_j - y_2$ contains an (x_1, y_1)-path avoiding y_2, then it follows from the fact that F is strong that D has the desired paths and hence D' exists as remarked above. Hence we may assume that $y_2 \notin V(S_j)$.

If S_j contains an (x_1, y_1)-path which does not cover all the vertices of S_j, then it follows from the fact that F is strong that D has the desired paths. Thus we may assume that either S_j has no (x_1, y_1)-path, or every (x_1, y_1)-path in S_j contains all the vertices of S_j. In the last case we may assume that $V(S_j)$ separates x_2 from y_2. Now D has the desired paths if and only if it has such a pair which does not use any arcs from S_j. Thus in both cases we can construct D' by replacing S_j by a tournament with no (x_1, y_1)-path and every other S_i by an arbitrary tournament on the same vertex set, except in the case when x_2 and y_2 belong to some S_i, $i \neq j$. In this case we replace that S_i by a tournament with no (x_2, y_2)-path (by the remark above we may assume that S_i has no (x_2, y_2)-path).

It follows from the considerations above that D' can be constructed in time $O(n^2)$. □

Recall that quasi-transitive digraphs can be decomposed according to Theorem 2.7.5. Hence we can apply Theorem 10.5.14 to these digraphs.

Theorem 10.5.15 *[74] There exists a polynomial algorithm for the 2-linkage problem for quasi-transitive digraphs.*

Proof: Let D be a quasi-transitive digraph and x_1, x_2, y_1, y_2 specified distinct vertices for which we want to determine the existence of vertex-disjoint (x_1, y_1)-,(x_2, y_2)-paths. First check that $D - \{x_i, y_i\}$ contains an (x_{3-i}, y_{3-i})-path for $i = 1, 2$. If not, then we stop. Now it follows from Theorem 2.7.5 that either x_1, x_2, y_1, y_2 are all in the same strong component of D, or the paths exist. For example, if D is not strong and y_1, say, is not in the same strong component as x_1 then, by Theorem 2.7.5, x_1 and y_1 belong to different sets W_i, W_j in the canonical decomposition $D = Q[W_1, \ldots, W_{|Q|}]$, where Q is a transitive digraph. Hence $x_1 \rightarrow y_1$ and the desired paths clearly exist.

Thus we may assume that D is strong. Let $D = S[W_1, W_2, \ldots, W_{|S|}]$ be a decomposition of D according to Theorem 2.7.5. Now apply Theorem 10.5.14 and construct the digraph D' which has the desired paths if and only if D does. As remarked in Theorem 10.5.14, D' can be constructed in polynomial

time. By the construction of D' (replacing each W_i by a semicomplete digraph) it follows that D' is a semicomplete digraph and hence we can apply the polynomial algorithm of Theorem 10.5.12 to D' in order to decide the existence of the desired paths in D. The algorithm of Theorem 10.5.12 can be used to find vertex-disjoint (x_1, y_1)-, (x_2, y_2)-paths in D' if they exist and given these paths it is easy to construct the corresponding paths in D (it suffices to take minimal paths). □

By inspecting the proof of Theorem 10.5.14 it is not difficult to see that the following much more general result is true. The main point is that in the proof of Theorem 10.5.14 we either find the desired paths or decide that they exist if and only if there are such paths that use no arcs inside any S_i. Hence instead of making each T_i semicomplete, we may just as well make it an independent set, by deleting all arcs inside S_i.

Theorem 10.5.16 *[74] Let Φ be a class of strongly connected digraphs, let Φ_0 denote the class of all extensions of graphs in Φ and let*

$$\Phi^* = \{F[D_1, \ldots, D_{|F|}] : F \in \Phi, \text{ each } D_i \text{ is an arbitrary digraph}\}.$$

There is a polynomial algorithm for the 2-linkage problem in Φ^* if and only if there is a polynomial algorithm for the 2-linkage problem for all digraphs in Φ_0. □

This result shows that studying extensions of digraphs can be quite useful.

One example of such a class Φ, for which Theorem 10.5.16 applies, is the class of strong semicomplete digraphs. This follows from the fact that we can reduce the 2-linkage problem for extended semicomplete digraphs to the case of semicomplete digraphs in the same way as we did for quasi-transitive digraphs in the proof of Theorem 10.5.15. Hence the 2-linkage problem is polynomially solvable for all digraphs that can be obtained from strong semicomplete digraphs by substituting arbitrary digraphs for vertices. It is important to note here that Φ must consist only of strong digraphs, since it is not difficult to reduce the 2-linkage problem for arbitrary digraphs (which is $\mathcal{NP}$-complete by Theorem 10.2.1) to the 2-linkage problem for those digraphs that can be obtained from the digraph H consisting of just an arc uv by substituting arbitrary digraphs for the vertex v.

The proof of the following easy lemma is left to the reader as Exercise 10.23. Note that four is best possible as can be seen from the complete biorientation of the undirected graph consisting of 4-cycle $x_1x_2y_1y_2x_1$ and a vertex z joined to each of the four other vertices.

Lemma 10.5.17 *Let D be a digraph of the form $D = \vec{C}_2[S_1, S_2]$, where S_i is an arbitrary digraph on n_i vertices, $i = 1, 2$. If D is 4-strong, then D is 2-linked.* □

The following result generalizes Corollary 10.5.2.

Theorem 10.5.18 *[74] Let $k \geq 4$ be a natural number and let F be a digraph on $f \geq 2$ vertices with the property that every k-strongly connected digraph of the form $F[T_1, T_2, \ldots, T_f]$, where each T_i, $i \in [f]$, is a semicomplete digraph, is 2-linked. Let $D = F[S_1, S_2, \ldots, S_f]$, where S_i is an arbitrary digraph on n_i vertices for all $i \in [f]$. If D is k-strongly connected, then D is 2-linked.*

Proof: Let $D = F[S_1, S_2, \ldots, S_f]$, where S_i is an arbitrary digraph on n_i vertices for each $i \in [f]$, be given. By Lemma 10.5.17 we may assume that D cannot be decomposed as $D = \vec{C}_2[R_1, R_2]$, where R_1 and R_2 are arbitrary digraphs. Construct D' as described in Theorem 10.5.14. Note that by Lemma 5.8.1, $\kappa(D') = \kappa(D)$. Thus D' is k-strong and using Theorem 10.5.14 and the assumption of the theorem we conclude that D is 2-linked. □

Corollary 10.5.19 *[74] Every 5-strong quasi-transitive digraph is 2-linked.*

Proof: By Theorem 2.7.5, every strong quasi-transitive digraph is of the form $D = F[S_1, S_2, \ldots, S_f]$, $f = |F|$, where F is a strong semicomplete digraph and each S_i is a non-strong quasi-transitive digraph on n_i vertices. By Lemma 2.7.4 and the connectivity assumption, $|F| \geq 3$. Note that for any choice of semicomplete digraphs $T_1, \ldots, T_f$ the digraph $D' = F[T_1, T_2, \ldots, T_f]$ is semicomplete. Hence the claim follows from Theorem 10.5.18 and the fact that, by Corollary 10.5.2, every 5-strong semicomplete digraph is 2-linked. (Since F has at least three vertices, it follows from Lemma 5.8.1 that $\kappa(D') = \kappa(D)$.) □

10.6 Linkages in Planar Digraphs

In this section we briefly discuss the k-linkage problem for planar digraphs (recall the definition of a planar digraph from Section 2.12). The constraint that the digraph in question can be embedded in the plane clearly poses some restrictions to the structure of vertex-disjoint paths. This is illustrated by the following result.

Proposition 10.6.1 *Suppose that $D = (V, A)$ is a planar digraph with distinct vertices $x, y, u, v \in V$ and that D is embedded in the plane in such a way that the vertices x, v, y, u appear on the bounding cycle C of the outer face in that order (see Figure 10.8). Then D does not have a pair of disjoint (x, y)-, (u, v)-paths.*

Proof: We first prove that no matter how we connect x and y by a simple (that is, not self-intersecting) curve R and u, v by another simple curve R', both inside the bounded disc with boundary C (see Figure 10.8) the two curves must intersect. Suppose we can choose simple curves R, R' so that R connects x and y and R' connects u and v. Then we can add a new point z in the interior of the outer face and join it to each of the vertices x, y, u, v

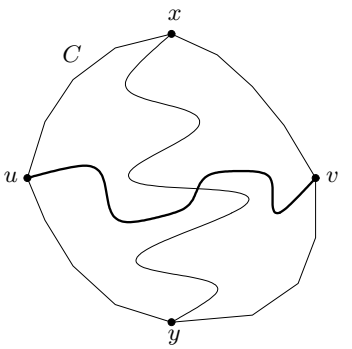


Figure 10.8 A topological obstruction for the existence of disjoint (x, y)- and (u, v)-paths in a planar graph G. The cycle C is the boundary of the outer face of G.

by disjoint simple curves which lie entirely in the closed disc formed by the outer face and its boundary C. This gives us an embedding of K_5 in the plane, contradicting Theorem 2.12.1.

Suppose now that P, Q are disjoint paths in D such that P is an (x, y)-path and Q is a (u, v)-path. In the embedding of D these correspond to simple curves and hence, by the argument above, they must intersect at some point in the plane. Since D is planar, no two arcs intersect in the interior (as curves) and hence we see that P and Q must intersect in some vertex v of D. However, this contradicts the assumption that they are disjoint. □

We point out that the first part of the proof above can be established using the Jordan curve theorem directly to establish that R and R' must intersect somewhere in the disc with boundary C (see, e.g., the book by Bondy and Murty [170]).

It was shown by Lynch [662] that when k is part of the input, then the k-linkage problem remains $\mathcal{NP}$-complete even for planar digraphs. For fixed k, Schrijver has developed a polynomial algorithm.

Theorem 10.6.2 *[799, 800] For every integer $k \geq 1$ the k-linkage problem is polynomially solvable for planar digraphs*[5]. □

The proof method is based on cohomology over free (non-abelian) groups, a topic which would require too much space to cover in the present book. Schrijver mentions that part of the group theory and topology is mainly used to keep notation fairly simple, but in any case the proof is too complicated to include here even as a (convincing) sketch. For additional discussion on and applications (for digraphs embedded on surfaces) of this very powerful proof technique we refer the reader to Schrijver's papers [799, 800, 801]. We should

[5] Note that k is not part of the input.

mention though that arguments like those used in the proof of Proposition 10.6.1 play an important role in Schrijver's approach.

To further illustrate how to use planarity in arguments in linkage problems, we consider a special case of the k-linkage problem for which a good characterization for the existence of a prescribed linkage has been found by Ding, Schrijver and Seymour [261].

Suppose that we are given a planar digraph $D = (V, A)$ which is embedded in the plane in such a way that the vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ all belong to the boundary of the outer face F of D. Ding, Schrijver and Seymour [261] proved that in this case there is a simple polynomial algorithm to decide the existence of a collection of disjoint paths $P_1, P_2, \ldots, P_k$, where P_i is an (s_i, t_i)-path for every $i \in [k]$.

In fact, as we will see below, it turns out to be easier to describe an algorithm for the following slight extension of the problem: in addition to the vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ we are also given subsets $A_1, A_2, \ldots, A_k$ of A and we demand that P_i can only use[6] arcs from A_i for all $i \in [k]$.

Motivated by the example in Figure 10.8 we say that two pairs of terminals (s_i, t_i) and (s_j, t_j) on $bd(F)$ **cross** if each simple curve from s_i to t_i in $\mathcal{R}^2 - F$ (considered as a subspace of $\mathcal{R}^2$) crosses each simple curve from s_j to t_j in $\mathcal{R}^2 - F$. By Proposition 10.6.1 a necessary condition for the existence of disjoint $(s_1, t_1), \ldots, (s_k, t_k)$-paths in D is that the following cross-freeness condition is satisfied:

$$\text{for every } i \neq j \ (s_i, t_i) \text{ and } (s_j, t_j) \text{ do not cross.} \tag{10.3}$$

Using the cross-freeness condition we see that there is no solution unless the terminals occur in the order $u_1, v_1, u_2, v_2, \ldots u_k, v_k$ around $bd(F)$, where $\{u_i, v_i\} = \{s_{\pi(i)}, t_{\pi(i)}\}$ for some permutation π of $[k]$. Clearly this condition can be checked in polynomial time if we are given the (polygonal) embedding of D.

We measure **closeness** of two polygonal paths with the same end-points by the area between the two paths. See Figure 10.9 for an illustration. The proof of the following lemma is left as Exercise 10.25.

Lemma 10.6.3 *Let R be a path from x to y along the boundary of the outer face (ignoring the orientation of the arcs in D) and let D' be a subdigraph of D which contains the vertices x and y. Then either D' has no (x, y)-path or there exist a unique (x, y)-path Q in D' which is closest to R. Given the embedding of D, we can find Q in polynomial time if it exists. Furthermore, no other (x, y)-path 'crosses over' Q at any point (e.g., in Figure 10.9 the path $v_8v_9v_5$ crosses over the path $v_2v_9v_{10}$ at the vertex v_9).* □

[6] In [261] Ding, Schrijver and Seymour consider an even more general case where not all paths linking different pairs of terminals must be disjoint, but for simplicity we assume that they are all disjoint.

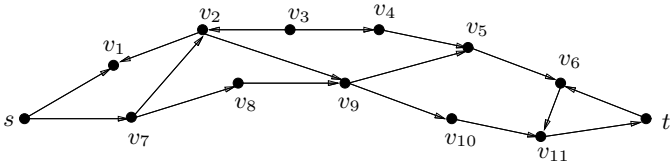


Figure 10.9 Let R be the path $sv_1v_2v_3v_4v_5v_6t$ in the underlying graph of D. The (s,t)-path $sv_7v_2v_9v_5v_6v_{11}t$ is closer to R than the (s,t)-path $sv_7v_8v_9v_5v_6v_{11}t$.

Now we are ready to describe a greedy algorithm which either finds the desired paths in D, or a proof that no such paths exist (using only arcs from the sets $A_1, A_2, \ldots, A_k$).

Start with s_k, t_k. Since D satisfies the cross-freeness condition, one of the two paths between s_k and t_k along $bd(F)$ contains no other terminals. Denote this path by P.

If $D\langle A_k\rangle$ contains no (s_k, t_k)-path, then there is no solution, so assume below that such a path exists.

Let P_k be the unique (s_k, t_k)-path in $D\langle A_k\rangle$ which is closest to P. Modify each A_i, $i \in [k-1]$, by removing from A_i every arc that is incident to a vertex on P_k. Now repeat the steps above for the pair s_{k-1}, t_{k-1} and continue recursively.

After at most k iterations we either find the required linkage or conclude that no such linkage exists.

To prove the correctness of the algorithm we observe that if $Q_1, Q_2, \ldots, Q_k$ is a solution, then so is $Q_1, Q_2, \ldots, Q_{k-1}, P_k$. Indeed, if P_k intersects some Q_i, then so does Q_k because P_k is either equal to Q_k or strictly closer to P than Q_k. This shows that the greedy choice is legal and the correctness follows. It also follows from Lemma 10.6.3 that the algorithm above is polynomial in the size of D.

We finish this section with some remarks on the problem (P3) in Proposition 10.1.2 for the case of planar digraphs. By Theorem 10.2.5 there is no degree of vertex-strong connectivity which guarantees that a digraph is 2-cyclic (that is, has a cycle containing x, y for every choice of vertices x, y). For planar digraphs the maximum degree of vertex-strong connectivity is 5 (Exercise 5.8). One may ask whether there is some degree of vertex-strong connectivity which suffices to guarantee that the planar digraph is 2-cyclic. However, this is not the case as shown by the 5-strong non-2-cyclic planar digraph D_k ($k = 20$) in Figure 10.10 (Exercise 10.27). This example arose from a personal communication with Böhme and Harant (October 1999). The fact that there exist 5-strong non-2-cyclic planar digraphs was also mentioned by Bermond and Thomassen in the survey paper [152]. Note also that these examples of 5-strong non-2-cyclic planar digraphs show that for directed graphs there is no analogue of Tutte's theorem on hamiltonian planar graphs (every 4-connected planar graph is hamiltonian [878]).

Using the same family of planar undirected graphs G_k, $k \geq 20$, as in Figure 10.10 one can easily construct 5-strong planar graphs which do not contain disjoint $[s_1, t_1]$-, $[s_2, t_2]$-paths, hence providing the proof that the condition of being 6-connected cannot be lowered to being 5-connected for undirected graphs (recall the discussion at the end of Section 10.2).

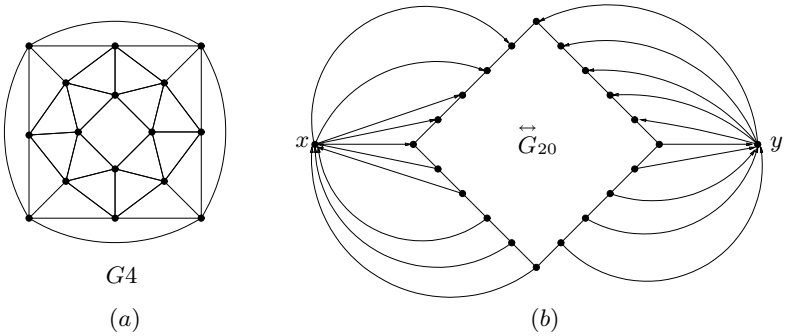


Figure 10.10 Part (a) shows a planar 5-connected graph G_k with $k = 4$; Part (b) shows a 5-strong planar digraph D_k that is obtained from the complete biorientation of G_k (shown for $k = 20$) by adding two new vertices x, y and joining these by the arcs indicated. The digraph has no cycle through x and y.

10.7 Weak Linkages

Let D be a directed multigraph and let $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ be a collection of not necessarily distinct vertices of D. A **weak k-linkage** from $(s_1, s_2, \ldots, s_k)$ to $(t_1, t_2, \ldots, t_k)$ is a collection of k arc-disjoint paths $P_1, \ldots, P_k$ such that P_i is an (s_i, t_i)-path for each $i \in [k]$. A directed multigraph $D = (V, A)$ is **weakly k-linked** if it contains a weak k-linkage from $(s_1, s_2, \ldots, s_k)$ to $(t_1, t_2, \ldots, t_k)$ for every choice of (not necessarily distinct) vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$. The WEAK k-LINKAGE PROBLEM is the following. Given a directed multigraph $D = (V, A)$ and distinct vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$; decide whether D contains k arc-disjoint paths $P_1, \ldots, P_k$ such that P_i is an (x_i, y_i)-path.

In view of Theorem 10.7.3 below, the following result by Fortune, Hopcroft and Wyllie may seem slightly surprising.

Theorem 10.7.1 *[332] The weak k-linkage problem is $\mathcal{NP}$-complete already for $k = 2$.*

Proof: Let $[D, x, y, u, v]$ be an instance of the 2-linkage problem. Transform $D = (V, A)$ into the directed multigraph H by performing the vertex splitting

procedure (see Section 4.2). Then it is easy to show that H has a pair of arc-disjoint (x_t, y_s)-, (u_t, v_s)-paths if and only if D has disjoint (x, y)-, (u, v)-paths (Exercise 10.28). Since H can be constructed from D in polynomial time, the claim now follows from Theorem 10.2.1. □

The following problem is mentioned by Schrijver in [799, page 265] and [801].

Problem 10.7.2 *Does there exist a polynomial algorithm to decide the existence of two arc-disjoint paths with prescribed end-vertices in a planar directed multigraph?*

Even the complexity of the special case when we are looking for arc-disjoint (x, y)- and (y, x)-paths is open! Hence we see from Theorem 10.6.2 that the weak 2-linkage problem is much more difficult for planar digraphs than the 2-linkage problem. This is not really surprising since planarity certainly has implications on vertex-disjoint paths, whereas the implications on arc-disjoint paths are not so obvious although there clearly are some.

Observe that if D is weakly k-linked, then D is k-arc-strong. To see this it suffices to take $s_i = x$ and $t_i = y$ for each i, then there are k arc-disjoint (x, y)-paths in D and since x, y may be chosen arbitrarily, it follows that D is k-arc-strong.

Shiloach observed [818] that Edmonds' branching theorem implies that k-arc-strong connectivity is also sufficient for the existence of k arc-disjoint paths with specified initial and terminal vertices:

Theorem 10.7.3 *[818] A directed multigraph D is weakly k-linked if and only if D is k-arc-strong.*

Proof: Above we have argued on the necessity. To see the sufficiency, let $x_1, x_2, \ldots, x_k, y_1, \ldots, y_k$ be given. Construct a new directed multigraph D' by adding a new vertex s and arcs sx_i for all $i \in [k]$ to D. Since D is k-arc-strong, it is not difficult to check that $d^-_{D'}(X) \geq k$ for every subset X of V. Hence by Theorem 9.3.1, D' has arc-disjoint out-branchings $B^+_{s,1}, \ldots, B^+_{s,k}$ all rooted at s. Since s has out-degree k in D', each $B^+_{s,i}$ must use precisely one arc out of s and without loss of generality $B^+_{s,i}$ uses the arc sx_i. Now it is clear that $B^+_{s,i}$ contains an (x_i, y_i)-path P_i and the paths $P_1, \ldots, P_k$ form the desired linkage. □

Using Theorem 9.3.3 we can obtain, in an analogous way, the following sufficient condition, due to Bang-Jensen, Frank and Jackson, for the existence of k arc-disjoint paths with prescribed initial and terminal vertices (Exercise 10.29).

Theorem 10.7.4 *[78] Let $(s_1, t_1), \ldots, (s_k, t_k)$ be k pairs of vertices in a directed multigraph $D = (V, A)$ so that for every vertex x with $d^-(x) < d^+(x)$ or $x = t_j$ there are arc-disjoint paths from s_i to x for every $i \in [k]$. Then there are arc-disjoint paths from s_i to t_i $(i = 1, 2, \ldots, k)$.* □

Note that if we only impose the condition in Theorem 10.7.4 on the vertices $t_1, t_2, \ldots, t_k$, then D may not have arc-disjoint paths from s_i to t_i $(i = 1, 2, \ldots, k)$. This can be seen from the example in Figure 10.11. The example can easily be generalized to arbitrary local strong connectivities from s_i to t_i, $i = 1, 2$, while preserving planarity. We formulate this as a theorem below.

Theorem 10.7.5 *For every natural number k there exists a planar digraph D with distinct vertices s_1, s_2, t_1, t_2 such that D has $\kappa_D(s_i, t_i) \geq k$ for $i = 1, 2$, but D has no arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths.* □

This shows that there is no sufficient condition for the existence of a weak linkage from $(s_1, s_2, \ldots, s_r)$ to $(t_1, t_2, \ldots, t_r)$ in terms of local vertex-strong connectivities from s_i to t_i, $i = 1, 2, \ldots, r$.

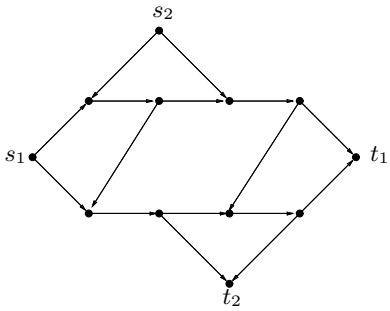


Figure 10.11 An example of a planar digraph with $\kappa(s_i, t_i) = 2$, $i = 1, 2$, and no arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths.

10.7.1 Weak Linkages in Acyclic Directed Multigraphs

The following easy observation, due to Fortune, Hopcroft and Wyllie, can be used to reduce the weak k-linkage problem for acyclic directed multigraphs to the k-linkage problem for the same class. We need the following lemma whose proof is left as Exercise 10.30.

Lemma 10.7.6 *If D is acyclic, then so is its line digraph $L(D)$.* □

Theorem 10.7.7 *[332] For each k, there exists a polynomial algorithm for the weak k-linkage problem for the class of acyclic directed multigraphs.*

Proof: Let $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ be an instance of the weak k-linkage problem where D is an acyclic directed multigraph. If some x_i has

out-degree zero or some y_j has in-degree zero, then trivially the desired paths do not exist. Hence we may assume that this is not the case.

Transform the instance $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ into a new instance $[D', x'_1, x'_2, \ldots, x'_k, y'_1, y'_2, \ldots, y'_k]$ as follows. If x_i has out-degree two or more, we add a new vertex x'_i and the arc $x'_i x_i$ to D; otherwise let $x'_i := x_i$, $i = 1, 2, \ldots, k$. Similarly, for each $j \in [k]$, if y_j has in-degree more than one, we add a new vertex y'_j and the arc $y_j y'_j$; otherwise let $y'_j := y_j$. Clearly, D' has arc-disjoint paths $P'_1, \ldots, P'_k$ such that P'_i is an (x'_i, y'_i)-path, $i = 1, 2, \ldots, k$, if and only if D has arc-disjoint paths $P_1, \ldots, P_k$, where P_i is an (x_i, y_i)-path, $i = 1, 2, \ldots, k$.

Now consider $D^* := L(D')$ and let s_i (t_i) be the vertex of D^* which corresponds to the unique arc with tail (head) x'_i (y'_i). Then it is easy to show that D^* has a collection $Q_1, Q_2, \ldots, Q_k$ of disjoint paths so that Q_i is an (s_i, t_i)-path, $i = 1, 2, \ldots, k$, if and only if D' has arc-disjoint paths $P'_1, \ldots, P'_k$ such that P'_i is an (x'_i, y'_i)-path, $i = 1, 2, \ldots, k$.

Since we have transformed the instance $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ into $[D^*, s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k]$ by a polynomial algorithm, the theorem now follows from Theorem 10.4.1. □

In [799], Schrijver shows how to apply a polynomial algorithm for the weak k-linkage problem in acyclic digraphs to solve a scheduling problem in the airline industry.

10.7.2 Weak Linkages in Eulerian Directed Multigraphs

As we will see below, questions about weak linkages are slightly easier for eulerian directed multigraphs than for arbitrary directed multigraphs. However, the weak 2-linkage problem seems difficult and is still open. As we also mention in Section 14.1, eulerian directed multigraphs often have properties similar to those of undirected multigraphs. This is also illustrated by their properties with respect to arc-disjoint paths as can be seen from some of the results mentioned in this subsection (see, e.g., Figure 10.13).

We start with a very simple, yet quite important observation. As mentioned earlier the complexity version of the corresponding problem for planar digraphs is still open (Problem 10.7.2).

Lemma 10.7.8 *Let D be a eulerian directed multigraph and let s, t be distinct vertices of D. Then D has arc-disjoint (s,t)-, (t,s)-paths if and only if D has an (s,t)-path.*

Proof: Let P be an arbitrary (s,t)-path. Let D' be obtained from D by removing the arcs of P. In D', every vertex distinct from s, t has in-degree equal out-degree and we have $d^-_{D'}(s) = d^+_{D'}(s) + 1$, $d^+_{D'}(t) = d^-_{D'}(t) + 1$. Let $\mathcal{N}(D')$ be the network representation of D' (recall Definition 5.1.4) and let x be the flow that has value equal to the capacity on every arc. By the flow

decomposition theorem (Theorem 4.3.1), x can be decomposed into a (t,s)-flow of value one and some cycle flows. Since the (t,s)-path in $\mathcal{N}(D')$ is also a path in D', D' contains a (t,s)-path as claimed. □

Let $x_1,\ldots,x_k$ be a k-tuple of (not necessarily distinct) vertices, which will be called **terminals**. We say that a trail $T=(v_0v_1v_2\ldots v_{t-1}v_t)$ **visits the terminals in the order** $x_1,x_2,\ldots,x_k$ if $x_1=v_{i_1},x_2=v_{i_2},\ldots,x_k=v_{i_k}$ for some[7] $0\leq i_1\leq\ldots\leq i_k\leq t$. Based on the following lemma (whose proof is left as Exercise 10.34), we could restrict ourselves only to eulerian trails. However, it is sometimes convenient to work also with non-eulerian trails.

Lemma 10.7.9 *Let D be an eulerian directed multigraph. Assume that there is a trail visiting some terminals in the order $x_1,x_2,\ldots,x_k$. Then there exists an eulerian trail visiting the terminals in the same order.* □

Given an eulerian directed multigraph and terminals $x_1,x_2,\ldots,x_k$ there are at least three different problems one may consider [546]:

Specific Trail (ST) Problem

Instance: An eulerian directed multigraph G and an ordered k-tuple of terminals $x_1,x_2,\ldots,x_k$.
Question: Does there exist a trail visiting the terminals in the order $x_1,\ldots,x_k$?

Unique Trail (UT) Problem

Instance: An eulerian directed multigraph G and an unordered k-tuple of terminals $x_1,x_2,\ldots,x_k$.
Question: Do all eulerian trails visit the terminals in the same cyclical order?

All Trail (AT) Problem

Instance: An eulerian directed multigraph G and an unordered k-tuple of terminals $x_1,x_2,\ldots,x_k$.
Question: Does there exist a trail T_π visiting the terminals in the order $x_{\pi(1)},\ldots,x_{\pi(k)}$ for every permutation π of $[k]$?

We will denote by k-ST, k-UT and k-AT the corresponding problems when the number of terminals is exactly k. The ST-problem seems to be the most important among these three problems, since it is equivalent to the eulerian weak linkage problem (see Lemma 10.7.10). However, the remaining two problems occur naturally in the study of the ST-problem.

[7] We do not exclude some additional occurrences of terminals in a trail. In general, a trail may visit given terminals in several different orders.

As we show below, results on these three problems for eulerian directed multigraphs are, in fact, strongly related to weak linkages in directed multigraphs which are not eulerian, but become eulerian if we add the so-called **demand arcs**. Let $[D, s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k]$ be an instance of the weak k-linkage problem. The **demand directed multigraph** H associated with this instance is the directed multigraph consisting of the arcs[8] $t_1s_1, t_2s_2, \ldots, t_ks_k$. The special case of the weak k-linkage problem when $D + H$ is eulerian (here H is the demand directed multigraph of D) is called the EULERIAN WEAK k-LINKAGE PROBLEM. When, instead of being a fixed number k, the number of demand arcs is part of the input, we call the above problem the EULERIAN WEAK LINKAGE PROBLEM.

Lemma 10.7.10 *The k-ST-problem is equivalent to the eulerian weak k-linkage problem.*

Proof: We show that the k-ST-problem is a special case of the eulerian weak k-linkage problem using the following reduction. Let $[D, x_1, \ldots, x_k]$ be an instance of the k-ST-problem. Define $s_1, t_1, \ldots, s_k, t_k$ by $s_i = x_i$ and $t_i = x_{i+1}, i = 1, 2, \ldots, k$, $(x_{k+1} = x_1)$ and let H consist of the arcs t_is_i, $i = 1, 2, \ldots, k$. Then $D + H$ is eulerian and it is easy to see that $D + H$ has arc-disjoint paths $P_1, \ldots, P_k$, where P_i is an (s_i, t_i)-path, $i = 1, 2, \ldots, k$, if and only if D has a trail visiting the terminals in the order $x_1, x_2, \ldots, x_k$.

Conversely, given an instance $[D, s_1, \ldots, s_k, t_1, \ldots, t_k]$ of the eulerian weak k-linkage problem (thus $D + H$ is eulerian), we construct an instance of the k-ST-problem as follows. Let $\tilde{D}$ be the directed multigraph obtained from D by adding new vertices $x_1, \ldots, x_k$, and arcs $x_is_i, t_ix_{i+1}, i = 1, 2, \ldots, k$. Clearly, $\tilde{D}$ is an eulerian directed multigraph, and it admits a closed trail visiting the terminals in the order $x_1, \ldots, x_k$ if and only if D admits a weak k-linkage for the prescribed pairs (s_i, t_i), $i = 1, 2, \ldots, k$, of terminals. □

Now we see from Lemma 10.7.8 that the weak 2-linkage problem is easy in the case when the directed multigraph in question becomes eulerian if we add the two demand arcs t_1s_1, t_2s_2. This was also observed by Frank in [341]. The eulerian weak 3-linkage problem is already considerably harder. It was solved by Ibaraki and Poljak [546]. We describe their main result in Theorem 10.7.11.

It is easy to see that for $k = 3$, the problems 3-ST, 3-UT, and 3-AT are mutually equivalent from a complexity point of view. The reason is that for $k = 3$ there are only two distinct cyclical orders of terminals, (x_1, x_2, x_3) and (x_1, x_3, x_2). Moreover, we may assume that one eulerian trail T of G is already given (since it may be constructed by a polynomial time algorithm according to Exercise 18.3). The trail T visits the terminals in one of the possible orders, say (x_1, x_2, x_3). Hence it only remains to decide whether there is a trail visiting the terminals in the other order.

[8] Hence, if $s_1 = s_2 = \ldots = s_k$ and $t_1 = t_2 = \ldots = t_k$, the demand directed multigraph consists of k parallel arcs from t_1 to s_1.

We recall the solution, due to Ibaraki and Poljak [546], of the UT-problem, since it suggests a possible approach to the remaining two problems. Recall that, for an arc a of D, D/a denotes the directed multigraph obtained from D by (set-)contracting the arc a. We allow terminals to be identified by the contraction. Below we denote the set of terminals by X and an instance of the UT-problem by $[D, X]$. Clearly, if $[D, X]$ admits several orders of visiting terminals, then $[D/a, X]$ admits several orders as well, but the converse need not be true. We say that $[D, X]$ is **UT-minimal**, if $[D, X]$ admits unique cyclical order of visiting terminals by an eulerian trail, but $[D/a, X]$ admits several orders whenever any arc a is contracted. Ibaraki and Poljak characterized UT-minimal instances.

Theorem 10.7.11 *[546] Let $[D, X]$ be a UT-minimal instance. Then*

(a) $d^+(x) = d^-(x) = 1$ for every terminal x, and $d^+(u) = d^-(u) = 2$ for every non-terminal u,
(b) D can be embedded in the plane such that every face is a directed cycle, and all terminals lie on one common face. □

Observe that the first part of the condition (b) is equivalent to the property that the four edges incident to a non-terminal vertex u are oriented alternatively out of and into the vertex u (in the planar representation). See Figure 10.12.

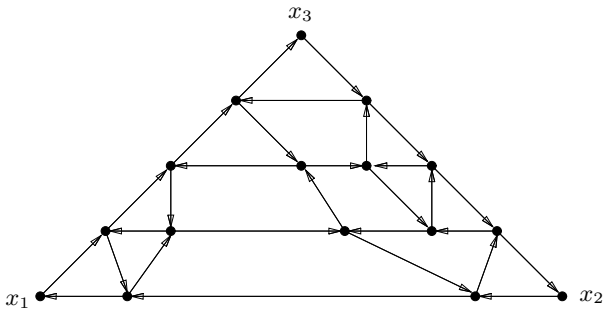


Figure 10.12 An eulerian digraph with no (eulerian) trail visiting x_1, x_2, x_3 in that order.

Theorem 10.7.12 *[546] Both the UT-problem and the 3-ST-problem are polynomially solvable.* □

Furthermore, Ibaraki and Poljak proved that the eulerian weak linkage problem and hence the ST-problem are $\mathcal{NP}$-complete.

Theorem 10.7.13 *[546] The eulerian weak linkage problem is $\mathcal{NP}$-complete.*

Proof: We sketch the construction used in [546]. The reduction is from the weak 2-linkage problem, which is $\mathcal{NP}$-complete by Theorem 10.7.1. Let $[D = (V, A), s_1, s_2, t_1, t_2]$ be an instance of the weak 2-linkage problem. Let $D^* = D + H$ be the directed multigraph we obtain from D by adding the two demand arcs t_1s_1 and t_2s_2.

Form a directed multigraph D' from D by adding two new vertices s, t and, for every $v \in V$, appending $\max\{0, d^+_{D^*}(v) - d^-_{D^*}(v)\}$ arcs of the form sv as well as $\max\{0, d^-_{D^*}(v) - d^+_{D^*}(v)\}$ arcs of the form vt. Let p be the sum of $d^+_{D^*}(v) - d^-_{D^*}(v)$ taken over those vertices for which this number is positive. Now let $s_i = s$ and $t_i = t$, $i = 3, 4, \ldots, p+2$, be new terminals. Then $[D', s_1, s_2, \ldots, s_{p+2}, t_1, t_2, \ldots, t_{p+2}]$ is an instance of the eulerian weak linkage problem and it is easy to show that D has arc disjoint (s_1, t_1)-, (s_2, t_2)-paths if and only if D' has arc-disjoint (s_i, t_i)-paths, $i = 1, 2, \ldots, p + 2$ (Exercise 10.35). □

Ibaraki and Poljak posed the following conjecture:

Conjecture 10.7.14 *[546] The k-ST-problem is polynomial for any fixed k.*

The condition of minimality which was used in Theorem 10.7.11 can be replaced by a more technical notion of irreducibility. Let us say that an instance $[D, X]$ is **2-irreducible** if there is no set S of vertices such that $|S| > 1$ and one of the following holds:

(a) $|(S, \bar{S})| = |(\bar{S}, S)| \leq 2$, $D\langle S\rangle$ is connected and $S \cap X = \emptyset$,
(b) $|(S, \bar{S})| = |(\bar{S}, S)| = 1$, and $|S \cap X| = 1$.

Note that D/S (the directed multigraph obtained by contracting S) is eulerian whenever D is eulerian. It is not difficult to see the following:

Lemma 10.7.15 *Let $[D, X]$ be an instance of the UT-problem which admits a unique order, and let S satisfy one of the conditions (a) and (b) above. Then $[D/S, X]$ admits a unique order as well.* □

It is also easy to see that D/S can be realized by a series of arc contractions, and hence every minimal UT-instance is 2-irreducible. Thus, the following theorem is a generalization of Theorem 10.7.11.

Theorem 10.7.16 *[546] Let $[D, X]$ be a UT-instance which is 2-irreducible and admits eulerian trail with unique order of terminals. Then the conditions (a) and (b) of Theorem 10.7.11 hold.*

The polynomial time algorithm for the UT-problem is a consequence of Theorem 10.7.16. The algorithm proposed in [546] consists of the following steps:

1. Reduce an instance $[D, X]$ to a 2-irreducible one. This can be done by applying network flow techniques.
2. Check the degree conditions.
3. Using a planarity test, decide whether D has a planar drawing, and if yes, then test the remaining conditions of Theorem 10.7.16.

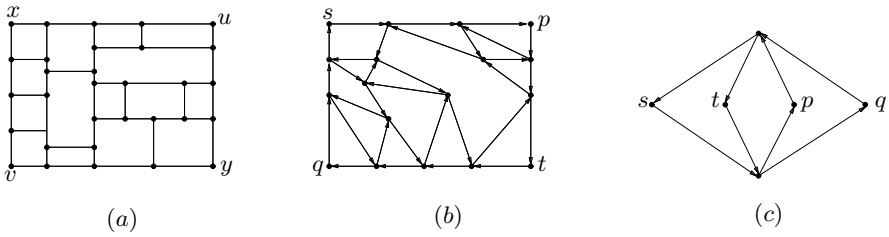
The notion of 2-irreducibility formulated here is weaker than the notion of irreducibility used in [546] where it was required, in addition, that $[D, X]$ does not contain any non-terminal vertex of in- and out-degree one. However, using the general definition of irreducibility given in [115, Section 3], it can be seen that this additional condition is automatically satisfied by any AT-infeasible and irreducible instance.

Let $[D, X]$ be an instance of AT-problem. Let us say that $[D, X]$ is **AT-minimal**, if $[D, X]$ does not admit an eulerian trail visiting the terminals for every given order, but $[D/a, X]$ does whenever any arc a is contracted. The following result by Bang-Jensen and Poljak shows that there are also degree restrictions on AT-minimal instances.

Theorem 10.7.17 *[115] Let $[D, X]$ be k-AT-minimal. Then $d^+(u) \leq k-1$ for every non-terminal u, and $d^+(x) \leq k-2$ for every terminal x.* □

The weak 2-linkage problem for undirected graphs is polynomially solvable and a complete characterization of undirected graphs having no edge-disjoint s_1t_1 and s_2t_2-paths is available (Dinic and Karzanov [263, 264], Seymour [808] and Thomassen [855]). Such a graph G can be reduced to a graph G' that has a planar representation with the following properties (see Figure 10.13(a)):

(a) Each of the four terminals has degree 2 and all other vertices have degree 3, and
(b) the terminals are located on the outer face in the order s_1, s_2, t_1, t_2.



(a) (b) (c)

Figure 10.13 Part (a) shows an infeasible instance for the edge-disjoint 2-linkage problem for undirected graphs. The graph shown has no xy-path and uv-path which are edge-disjoint; Parts (b) and (c) show infeasible instances of the arc-disjoint $[s, t]$-, $[p, q]$-paths problem for eulerian directed multigraphs.

The complete biorientation $\overleftrightarrow{G}$ of an undirected graph G is eulerian and it contains arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths if and only if G contains edge-disjoint s_1t_1, s_2t_2-paths. Hence, the weak 2-linkage problem for eulerian digraphs generalizes the weak 2-linkage problem. So far the weak 2-linkage problem for eulerian digraphs remains unsolved. However, even the simpler version in which we just require arc-disjoint $[s_1, t_1]$, $[s_2, t_2]$-paths (that is, the order of s_i, t_i is not fixed in the ith path, $i = 1, 2$) still generalizes the edge-disjoint 2-linkage problem. This problem was solved by Frank, Ibaraki and Nagamochi in [353]. They proved that the problem is solvable in polynomial time. Furthermore they showed the following result. Below, by a **reduction**, we mean a series of transformations such that the desired paths exist in the new digraph if and only if they exist in the previous digraph (for details see [353]).

Theorem 10.7.18 *[353] Let D be an eulerian directed multigraph and let s_1, s_2, t_1, t_2 be not necessarily distinct vertices of D. Then D contains arc-disjoint $[s_1, t_1]$, $[s_2, t_2]$-paths, unless it can be reduced to an eulerian directed multigraph D' such that either D' has six vertices and is isomorphic to the digraph in Figure 10.13(c), or each of (a),(b) and (c) below hold.*

(a) Each of s_1, s_2, t_1, t_2 has in- and out-degree one and all other vertices have in- and out-degree two in D'.
(b) There is at most one cut vertex[9] in $UG(D')$.
(c) D has a planar embedding such that every face is a directed cycle and all terminals are located on the outer face in the order s, p, t, q where $\{s, t\} = \{s_1, t_1\}$ and $\{p, q\} = \{s_2, t_2\}$. □

10.7.3 Weak Linkages in Tournaments and Generalizations of Tournaments

We now consider the weak 2-linkage problem for some generalizations of tournaments. We prove that this problem and a related special case (the arc version of problem (P5) from Proposition 10.1.2) are polynomially solvable for semicomplete digraphs. The corresponding algorithms are used as subroutines in a much more complicated algorithm by Bang-Jensen [68] for the problem concerning arc-disjoint in- and out-branchings in tournaments. We prove the first results for the class of extended locally in-semicomplete digraphs instead of just for semicomplete digraphs. We do this to show that not much extra effort is needed to obtain the result (which also has the same statement as for semicomplete digraphs alone) for this much larger class of digraphs. The results in this subsection are due to Bang-Jensen [68, 73]

[9] Recall that a vertex x in a connected undirected graph G is a cut vertex if $G - x$ is not connected.

Recall that two vertices are similar if and only if they are non-adjacent and have the same in- and out-neighbours. Note that if x, y are non-adjacent vertices with a common out-neighbour w in an extended locally in-semicomplete digraph, then x and y are similar vertices, by the definition of an extension and the definition of a locally in-semicomplete digraph.

The following lemma can be proved along the same lines as Lemma 10.7.20. The proof is left to the reader as Exercise 10.31.

Lemma 10.7.19 *Let D be a strong extended locally in-semicomplete digraph and let x, y be distinct vertices of D. Then D has arc-disjoint (x,y)-,(y,x)-paths if and only if there is no arc a such that $D-a$ contains no (x,y)-path and no (y,x)-path.* □

Lemma 10.7.20 *[73] Let $D=(V,A)$ be an extended locally in-semicomplete digraph and x, y, z vertices of D such that $x \neq z$ and D contains a path from y to z. If D has arc-disjoint (x,y)-, (x,z)-paths, then D contains arc-disjoint (x,y)-, (y,z)-paths. Similarly, if an extended locally out-semicomplete digraph D' has a path from x to y and arc-disjoint (x,z)-, (y,z)-paths, then D' has arc-disjoint (x,y)- and (y,z)-paths.*

Proof: Let P_1 and P_2 be arc-disjoint paths such that P_2 is an (x,z)-path and P_1 is a minimal (x,y)-path. If $y \in V(P_2)$, or $yx \in A$, then the claim is trivial so we assume that none of these hold. We can also assume that x and y are not similar vertices, because if they are, then y dominates the successor of x on P_2 and again the claim is trivial.

If D has a (y,z)-path whose first intersection with $V(P_1) \cup V(P_2)$ (starting from y) is on P_2, then the desired paths clearly exist. Hence we may assume that D contains a path from y to $V(P_1) \cup V(P_2) - y$ whose only vertex w from $V(P_1) \cup V(P_2) - y$ is in $V(P_1) - V(P_2)$. Now choose P among all such paths so that w is as close as possible to x on P_1. By the assumption above $w \neq x$. Let u (v) denote the predecessor of w on P_1 (P), i.e., $u = w^-_{P_1}$ and $v = w^-_P$.

Suppose first that u and v are not adjacent. Then, by the remark just before Lemma 10.7.19, u and v are similar. Now the choice of P implies that $v = y$ (otherwise the predecessor of v on P dominates u, contradicting the choice of P). By the assumption that x and y are not similar we conclude that $u \neq x$, but then $u^-_{P_1} y \in A$, contradicting the minimality of P_1.

Thus we may assume that u and v are adjacent. By the choice of P, this implies that $uv \in A$. Choose r as the first vertex on P which is dominated by u. By the minimality of P_1, $r \neq y$. Let s be the predecessor of r on P. The choice of r and P implies that u and s are similar. Thus as above, we must have $s = y$, and since $u \neq x$ we reach a contradiction as before.

The second half of the lemma follows from the first by considering the converse and interchanging the names of x and z. □

The digraph $D = (V, A)$ with vertex set $V = \{x, u, v, y, z\}$ and arc set $A = \{xu, uv, vy, yu, vz, xz\}$ shows that the conclusion of Lemma 10.7.20 does not hold for general digraphs.

Using Lemma 10.7.20 we can now characterize those extended locally in-semicomplete digraphs which do not have arc-disjoint (x, y)-, (y, z)-paths.

Theorem 10.7.21 *[73] An extended locally in-semicomplete digraph D has arc-disjoint (x, y)-, (y, z)-paths if and only if it has an (x, y)-path and a (y, z)-path and D has no arc e such that $D - e$ has no (x, y)-path and no (y, z)-path*[10].

Proof: Clearly if D has such an arc e, then the paths cannot exist. Now assume that D has no such arc and that D has an (x, y)-path and a (y, z)-path. We prove that D has the desired paths. By Lemma 10.7.19 we may assume $x \neq z$.

By Lemma 10.7.20, we may assume that D contains no pair of arc-disjoint (x, y)-,(x, z)-paths. Thus, by Menger's theorem, there exists an arc $e = uv$ such that $D - e$ has no path from x to $\{y, z\}$. Let $X = \{w : \exists (x, w) - \text{path in } D - e\}$ and $B = V(D) - X$. Then $x \in X$, $y, z \in B$ and the only arc from X to B is e.

Since D contains an (x, y)-path, $D\langle X\rangle$ has an (x, u)-path and $D\langle B\rangle$ has a (v, y)-path. $D\langle B\rangle$ also has a (y, z)-path, since e does not destroy all paths from y to z.

If $v = y$, the desired paths clearly exist (and can in fact be chosen vertex disjoint). If $v = z$, then it follows from our assumption that there is no arc a in $D\langle B\rangle$ which separates y from z and also z from y. Now it follows from Lemma 10.7.19 that $D\langle B\rangle$ contains arc-disjoint (z, y)-, (y, z)-paths and hence D contains the desired paths. Thus we may assume $v \neq y, z$.

Now it is clear that the desired paths exist if and only if $D\langle B\rangle$ has arc-disjoint (v, y)-, (y, z)-paths. By induction this is the case unless there exists an arc $e' = ab$ in $D\langle B\rangle$ such that $D\langle B\rangle - e'$ has no path from v to y and no path from y to z, but then e' separates x from y and y from z in D, contradicting the assumption that D has no such arc. □

Our proof above is constructive and hence we have the following (see also Exercise 10.32):

Corollary 10.7.22 *[68] There exists a polynomial algorithm which, given an extended in-semicomplete digraph D and distinct vertices x, y, z, either returns a pair of arc-disjoint (x, y)-, (y, z)-paths or an arc a such that $D - a$ has no (x, y)-path and no (y, z)-path.* □

We can now prove the main result of this section.

[10] Figure 10.11 with $s_2 = t_1$ shows that the theorem does not hold for planar digraphs.

Theorem 10.7.23 *[68] The weak 2-linkage problem is polynomially solvable for semicomplete digraphs.*

Proof: (Sketch) Let $[D, x_1, x_2, y_1, y_2]$ be an instance of the weak 2-linkage problem for semicomplete digraphs. By relabelling if necessary, we can assume that $x_1 \to x_2$. Below it is understood that we stop as soon as the existence of the desired paths has been decided.

It is easy to check whether there is any arc e such that $D-e$ has no (x_i, y_i)-path for $i = 1, 2$. If such an arc exists, then D does not have the desired paths and we stop. Now check whether D contains arc-disjoint (x_2, y_1)-, (x_2, y_2)-paths P, P'. If this is the case, then either x_1P or $P[x_1, y_1]$ (if $x_1 \in V(P)$) and P' are the desired paths and we stop.

Hence, by Menger's theorem, there is an arc e such that $D - e$ has no path from x_2 to $\{y_1, y_2\}$. Let

$$Y := \{v : v \text{ has a path to } \{y_1, y_2\} \text{ in } D - e\} \quad \text{and} \quad X := V(D) - Y.$$

Then $x_2 \in X$ and $x_1 \in Y$, because the arc e does not separate x_1 from $\{y_1, y_2\}$. Furthermore, e is the only arc from X to Y. Let z be the head of e and let w be its tail. Note that $D\langle X\rangle$ contains an (x_2, w)-path Q since D contains an (x_2, y_2)-path.

If $z = x_1$, then the desired paths exist: We cannot have another arc e' which separates x_1 from $\{y_1, y_2\}$ in $D' = D\langle Y\rangle$ because then e' separates $\{x_1, x_2\}$ from $\{y_1, y_2\}$ and we would have stopped earlier. Thus, by Menger's theorem, D' contains arc-disjoint (x_1, y_1)-, (x_1, y_2)-paths P_1, P_2, implying that P_1 and QP_2 are the desired paths.

If $z = y_2$, then the desired paths exist since any (x_1, y_1)-path in D' and Qy_2 will work.

If $z = y_1$, then the desired paths exist if and only if D' contains arc-disjoint (x_1, y_1)-, (y_1, y_2)-paths. This can be decided in polynomial time by the algorithm whose existence follows from Corollary 10.7.22.

Finally, if $z \notin \{x_1, y_1, y_2\}$, then the desired paths exist if and only if D' contains arc-disjoint (x_1, y_1),- (z, y_2)-paths. Hence we have reduced the problem to a smaller one of the same kind.

We leave it to the reader to verify that our steps above can be performed in polynomial time and to estimate the time complexity of the algorithm (Exercise 10.33). □

10.8 Linkages in Digraphs with Large Minimum Out-Degree

In this section we consider linkages in digraphs with lower bounds on the out-degree, but no condition on the degree of (arc)-strong connectivity. By Menger's theorem, the following result due to Mader would be trivial if we also required that D was k-arc-strong.

Theorem 10.8.1 *[670] For every integer $k \geq 1$, every digraph D with $\delta^+(D) \geq k$ contains a pair of distinct vertices x, y so that $\lambda(x, y) \geq k-1$.* □

Mader [670] gave an example of a digraph D with $\delta^+(D) = 12k$ such that no pair of distinct vertices are joined by more than $11k$ internally disjoint paths. On the other hand, he proved the following:

Theorem 10.8.2 *[673] There exists a function $f(k)$ such that every digraph D with $\delta^+(D) \geq f(k)$ contains a pair of distinct vertices x, y such that $\kappa(x, y) \geq k$. Furthermore $f(k) = k$ for $k \leq 3$.* □

The value of $f(k)$ above is not known for $k \geq 4$. Mader [673] proved that $f(4) < 40$ and gave an example showing that $f(9) > 9$.

It is natural to ask whether one can also guarantee the existence of a pair of vertices x, y so that $\min\{\kappa(x, y), \kappa(y, x)\} \geq k$, provided that the minimum out-degree is sufficiently high (as a function of k). However, already for $k = 2$ this is not true, as shown by a construction by Mader in [670]. The example in [670] also shows that there is no function $f(k)$ so that $\delta(D) \geq f(k)$ implies the existence of a pair of vertices x, y so that $\min\{\kappa(x, y), \kappa(y, x)\} \geq k$.

10.8.1 Subdivisions of Transitive Tournaments in Digraphs of Large Out-Degree

A fundamental result of Mader [663] states that for every integer k there is a smallest integer $d(k)$ such that every undirected graph of average degree at least $d(k)$ contains a subdivision of the complete graph on k vertices. The following conjecture by Mader would provide a digraph analogue of this result and would generalize Theorem 10.8.2.

Conjecture 10.8.3 *[673] There exists a function $h : \mathbb{Z}_+ \to \mathbb{Z}_+$ such that every digraph D with $\delta^+(D) \geq h(k)$ contains a subdivision of TT_k.*

The conjecture is easily seen to hold for $k \leq 3$ with $h(k) = k - 1$ for $k \leq 3$. For $k = 2$ this is trivial and for $k = 3$ we can argue as follows: Let D have minimum out-degree at least 2. Let v be an arbitrary vertex and start growing a BFS tree from v. First observe that either we find a TT_3 or each vertex in $N^+(v)$ has an out-neighbour in $N^{+2}(v)$. Now, if we do not find the desired subdivision of TT_3 in $D\langle\{v\} \cup N^+(v) \cup N^{+2}(v) \cup \ldots \cup N^{+r}(v)\rangle$, then each vertex in $N^{+r}(v)$ must have a private out-neighbour in $N^{+(r+1)}(v)$. Since D is finite, this process cannot continue indefinitely and hence we will eventually find the desired subdivision of TT_3 in D.

Mader [674] proved that $h(4) = 3$. Even the existence of $h(5)$ is open. It is easy to see that we cannot replace the lower bound on the out-degree by a lower bound on the average out-degree since the orientation of the complete bipartite graph $K_{k,k}$, where we orient all edges from the first colour class to

the second, has average out-degree $k/2$ but no subdivision of TT_3 and no directed cycle.

In [629] Kühn, Osthus and Young showed that if the minimum out-degree of a digraph is sufficiently large compared to its order, then one can in fact obtain a subdivision of a complete digraph.

Theorem 10.8.4 *[629] Let D be a digraph on n vertices and out-degree at least d. Then D contains a subdivision of the complete digraph of order $\lfloor d^2/(8n^{3/2})\rfloor$.* □

The bound in the theorem is non-trivial as soon as $d \geq 4n^{3/4}$ which is the bound for having a subdivision of a 2-cycle. Thomassen [860] gave a construction of a digraph on n vertices and minimum out-degree at least $\frac{1}{2}\log_2 n$ which has no even cycle and hence contains no subdivision of the complete digraph on three vertices.

10.9 Miscellaneous Topics

10.9.1 Universal Arcs in 2-Cyclic Digraphs

Recall that a digraph D is 2-cyclic if the vertices x, y are on a common cycle for every choice of vertices in $x, y \in V(D)$. Ádám [5] calls an arc a in digraph D **universal** if D contains a cycle through a and x for every $x \in V(D)$. He also asked which 2-cyclic digraphs have the property that they contain a universal arc. Hetyei [521] conjectured that this always holds.

Conjecture 10.9.1 *[521] Every 2-cyclic digraph has a universal arc.*

Hubenko [543] proved that every 2-cyclic bipartite tournament has a universal arc. She also proved the following slightly stronger statement.

Theorem 10.9.2 *[543] Let D be a 2-cyclic bipartite tournament such that $\delta^0(D) \geq 2$. Then every longest cycle contains a universal arc.* □

Hubenko [543] also posed the following problem.

Problem 10.9.3 *[543] Let D be a 2-cyclic bipartite tournament. Is it true that for every maximal cycle[11] C all arcs of C are universal?*

Volkmann and Winzen generalized Hubenko's results to arbitrary multipartite tournaments.

Theorem 10.9.4 *[896] Every 2-cyclic multipartite tournament contains a universal arc.* □

Theorem 10.9.5 *[896] Let D be a strong multipartite tournament with $\delta^0(D) \geq 2$. Then every longest cycle of D contains a universal arc.* □

[11] A cycle is maximal in D if there is no cycle C' in D such that $V(C) \subset V(C')$. Note that maximal cycles are non-extendable, but the converse is not always the case.

10.9.2 Integer Multicommodity Flows

Recall the definition of a network and a flow from Chapter 4. In this section we consider briefly the following common generalization of flows and weak linkages called the INTEGER MULTICOMMODITY FLOW PROBLEM (if k is fixed in advance, we call it the INTEGER k-COMMODITY FLOW PROBLEM): Given a natural number $k \geq 1$, a network $\mathcal{N} = (V, A, \ell \equiv 0, u)$, $2k$ not necessarily distinct vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ and integers $r_1, r_2, \ldots, r_k$, decide whether there exist integer-valued flows $f^1, f^2, \ldots, f^k$ such that each of the following holds (recall that $|f^i|$ is the value of the flow f^i):

(i) f^i is an (s_i, t_i)-flow in $\mathcal{N}$,
(ii) $|f^i| \geq r_i$ for all $i \in [k]$,
(iii) $f^p_{ij} \geq 0$ for every $ij \in A$, $p \in [k]$,
(iv) For every $ij \in A$: $\sum_{p=1}^{k} f^p_{ij} \leq u_{ij}$.

A collection of flows $f^1, f^2, \ldots, f^k$ which satisfies (i)-(iv) is called a **feasible k-commodity flow** with respect to (s_i, t_i), $i = 1, 2, \ldots, k$. We can also consider the **maximization version** where no demands $r_1, r_2, \ldots, r_k$ are specified (or they are to be considered as lower bounds) and the goal is to maximize the sum of the values of the flows.

If we take $k = 1$, we have the standard (maximum) (s, t)-flow problem for which several polynomial algorithms were described in Chapter 4. However, Even showed that already when $k = 2$ the problem becomes very hard.

Theorem 10.9.6 *[307] The integer 2-commodity problem is $\mathcal{NP}$-complete.*

Proof: The problem clearly belongs to $\mathcal{NP}$ since given a feasible instance we can take specifications of two feasible flows, one from s_1 to t_1 and the other from s_2 to t_2, as a valid certificate.

Now let $[D = (V, A), x_1, x_2, y_1, y_2]$ be an instance of the weak 2-linkage problem. Let $\mathcal{N} = (V, A, \ell \equiv 0, u \equiv 1)$, take $s_i = x_i, t_i = y_i$, $i = 1, 2$ and let $r_1 = r_2 = 1$. Then it is easy to see that D has arc-disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if $\mathcal{N}$ has a feasible integer 2-commodity flow with respect to the pairs (s_i, t_i), $i = 1, 2$. Now the claim follows from Theorem 10.7.1. □

What we really observed above was simply that the weak 2-linkage problem is nothing but a very special case of the 2-commodity flow problem. This is not surprising since if we concentrate on one of the two flows f^i in a feasible integer 2-commodity flow (with respect to the values r_1, r_2 and the capacities of the given network), then f^i is just a normal (s_i, t_i)-flow and hence can be decomposed into r_i (s_i, t_i)-paths and some cycle flows by Theorem 4.3.1. Hence the integer multicommodity flow problem is nothing but a generalization of weak linkage problems.

The name **multicommodity flow** comes from the interpretation of each flow as representing a different commodity that has to be shipped from the source of that commodity to its sink while respecting the total capacity of

the network. Problems of this type are of importance in practical applications such as telecommunications and routing problems. For a number of results on how to solve multicommodity flow problems in practice see the book by Gondran and Minoux [422]. See also the survey [52] by Assad and Chapter 70 of Schrijver's book [805]

10.10 Exercises

10.1. Prove Proposition 10.1.1.

10.2. Prove Proposition 10.1.2.

10.3. Prove that problem (P5) of Proposition 10.1.2 for semicomplete digraphs can be reduced to the 2-linkage problem for semicomplete digraphs in polynomial time.

10.4. Prove Lemma 10.2.2.

10.5. **The 2-linkage problem is NP-hard for digraphs of maximum outdegree 2.** Prove this claim. Hint: modify the digraph $D[\mathcal{F}]$ in Figure 10.3.

10.6. Prove Theorem 10.3.1. Hint: use Lemma 10.3.2.

10.7. Prove Theorem 10.3.3 without using Theorem 10.3.4.

10.8. Prove Lemma 10.3.2.

10.9. Let D be the acyclic digraph in Figure 10.14. Show that the digraph D' defined as in the proof of Theorem 10.4.1 has a directed path from (x_1, x_2, x_3) to (y_1, y_2, y_3).

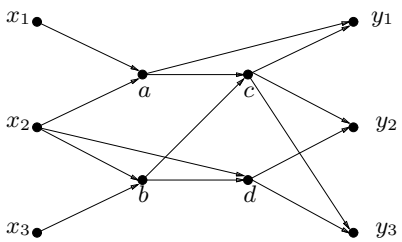


Figure 10.14 An instance of the 3-path problem for acyclic digraphs.

10.10. (+) Argue that in the algorithm which is implicit in the proof of Theorem 10.4.1 we do not really need to construct D' when searching for a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. Does that lead to an improvement in the complexity estimate?

10.11. **Finding a cheapest collection of k disjoint paths with prescribed ends in an acyclic digraph with costs on the arcs.** Show that the approach used in the proof of Theorem 10.4.1 can be modified so that one

can find the cheapest collection of disjoint paths joining x_i to y_i for $i = 1, 2, \ldots, k$.

10.12. (+) Prove that under the assumption of Corollary 10.4.3, for every non-special vertex v, the digraph D contains directed (x_1, v)-, (x_2, v)-, (v, y_1)-, (v, y_2)-paths such that the only common vertex of any two of these paths is v (Lucchesi and Giglio [660]). Hint: use Menger's theorem and the fact that D is acyclic.

10.13. **A sufficient condition for digraph to be 2-linked.** Let $D = (V, A)$ satisfy $d^+(x) + d^-(y) \geq n + 2$ whenever D does not contain the arc xy. Prove that D is 2-linked. Hint: first show that if $xy \notin A$, then there are three internally disjoint (x, y)-paths of length 2 in D (Heydemann and Sotteau [525]).

10.14. Prove that every k-linked digraph is also k-strong.

10.15. Prove that if a digraph $D = (V, A)$ is 2-linked, then for every choice of distinct vertices x, y, D contains disjoint cycles C_x, C_y such that $x \in V(C_x), y \in V(C_y)$. Generalize this to k-linked digraphs and k vertices.

10.16. (−) **Disjoint cycles containing prescribed vertices in tournaments.** Prove that a tournament T contains disjoint cycles C_x, C_y such that $x \in V(C_x), y \in V(C_y)$ if and only if T contains disjoint 3-cycles such that one contains x and the other contains y.

10.17. Describe how to construct the collection $Q'_1, Q'_2, \ldots, Q'_q$ of subpaths in the proof of Proposition 10.5.1. What is the complexity of your algorithm?

10.18. Show how to turn the proof of Proposition 10.5.1 into an algorithm which takes as input a collection $P_1, P_2, \ldots, P_p$ of internally disjoint (x, y)-paths and a collection $Q_1, Q_2, \ldots, Q_q$ of internally disjoint (u, v)-paths in $D - \{x, y\}$ and finds a collection of q (u, v)-paths which intersect no more than $2q$ vertices of $P_1, P_2, \ldots, P_p$.

10.19. Let D be a locally semicomplete digraph and let x, y be distinct non-adjacent vertices. Prove that every minimal (x, y)-path is an induced path (Bang-Jensen [66]).

10.20. (−) Let D be a locally semicomplete digraph such that $\alpha(D) = 2$. Prove that if x and y are non-adjacent vertices of D and D has an (x, y)-path, then there exists an (x, y)-path P of length at most 3.

10.21. (+) Prove the following statement. Let $k \geq 3$, let D be a k-strong locally semicomplete digraph which is round decomposable and let $D = R[S_1, \ldots, S_r]$ be the round decomposition of D. Let x and y be vertices such that $x \in V(S_i)$ and $y \in V(S_j)$, where $i \neq j$, and let P be a minimal (x, y)-path. Then $D - V(P)$ is $(k - 2)$-strong (Bang-Jensen [74]). Hint: use Exercise 10.20.

10.22. (+) Prove Lemma 10.5.3. Hint: use Exercise 10.21.

10.23. Prove Lemma 10.5.17.

10.24. (++) Prove Theorem 10.5.13.

10.25. Prove Lemma 10.6.3. Hint: show how to modify a given (x, y)-path which is not closest to R into one which is closer by a stepwise (but finite and polynomially bounded) improvement. For the algorithmic part you can use that the embedding is with polygonal curves.

10.26. Prove that the graph G_4 in Figure 10.10(a) is 5-connected.

10.27. Prove that the digraph D_k in Figure 10.10(b) is 5-strong and has no cycle through x, y. Hint: use Exercise 14.8 and Proposition 10.6.1.

10.28. Supply the missing details in the proof of Theorem 10.7.1.

10.29. Prove Theorem 10.7.4.

10.30. (−) Prove Lemma 10.7.6.

10.31. Prove Lemma 10.7.19.

10.32. Determine the complexity of the algorithm of Corollary 10.7.22.

10.33. Fill in the missing details of the proof of Theorem 10.7.23. What is the complexity of this recursive algorithm?

10.34. Prove Lemma 10.7.9.

10.35. Prove the last Claim in the proof of Theorem 10.7.13. Hint: use the same approach as in the proof of Lemma 10.7.8.

10.36. **Fan-in, fan-out in eulerian directed multigraphs.** Let D be an eulerian directed multigraph and suppose D has arc-disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i starts at x_i and ends at u for every $i \in [k]$. Prove that D contains arc-disjoint paths $P'_1, P'_2, \ldots, P'_k$ such that P'_i is a (u, x_i)-path.

10.37. **(+) Arc-disjoint (x, y)-, (y, z)-paths in quasi-transitive digraphs.** Prove that the characterization in Theorem 10.7.21 can be extended to quasi-transitive digraphs.

10.38. Show that the 3-ST-problem for eulerian digraphs can be reduced in polynomial time to the problem of deciding the existence of arc-disjoint $[s_1, t_1]$-, $[s_2, t_2]$-paths in an eulerian digraph with specified vertices s_1, t_1, s_2, t_2. Hint: use Exercise 10.36.

10.39. Prove that the arc-version of problem (P5) of Proposition 10.1.2 is $\mathcal{NP}$-complete.

11. Orientations of Graphs and Digraphs

The purpose of this chapter is to discuss various aspects of orientations of (multi)graphs. There are many ways of looking at such questions. We can ask which graphs can be oriented as a digraph of a certain type (e.g., a locally semicomplete digraph). We can try to obtain orientations containing no directed cycles of even length, or no long paths. We can try to relate certain parameters of a graph to the family of all orientations of this graph (e.g., what does high chromatic number imply for orientations of a graph). We can also look for conditions which guarantee orientations with high arc-strong connectivity or high in-degree at every vertex, etc. There are hundreds of papers dealing with orientations of graphs in one way or another and we can only cover some of these topics. Hence we have chosen some of those mentioned above. Finally, we also study briefly the theory of submodular flows, which generalizes standard flows in networks and turns out to be a very useful tool (not only theoretically, but also algorithmically) for certain types of connectivity questions as well as orientation problems. We illustrate this by applying the submodular flow techniques to questions about orientations of mixed graphs and by giving a short proof of Nash-Williams' orientation theorem. In Section 13.1 we also use submodular flows to give an algorithmic proof of the Lucchesi-Younger Theorem.

We also discuss orientations of digraphs and point out here that although the concept of orienting mixed graphs and orienting digraphs (that is, deleting one arc from every 2-cycle) are very closely related, they are actually different problems in many cases. This is because a mixed graph may contain directed 2-cycles and when we orient a mixed graph M we only orient the undirected part of M.

11.1 Underlying Graphs of Various Classes of Digraphs

In this section we discuss the underlying undirected graphs of some generalizations of tournaments. As can be seen, these include important classes of undirected graphs such as comparability graphs and proper circular arc graphs. For much more information about these classes and their relations to each other, the reader is encouraged to consult the books [178] by Brandstädt, [421] by Golumbic, and [754] by Prisner. Here we will just define those classes

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 417
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_11,

that we need. A graph G is a **circular arc graph** if there exists a family of circular arcs indexed by the vertices of the graph such that two vertices are adjacent if and only if the two corresponding arcs intersect. This family of circular arcs form a **representation** of G. A **proper circular arc graph** is a circular arc graph which has a representation by circular arcs, none of which is properly contained in another. A graph G is **chordal** if every cycle of length at least 4 has a chord, that is, G has no induced cycle of length four or more. Finally, G is a **comparability graph** if it has a transitive orientation (that is, there exists a transitive oriented graph T such that $UG(T)$ is isomorphic to G).

We will always use Δ to denote the maximum degree of the undirected graph in question.

11.1.1 Underlying Graphs of Transitive and Quasi-Transitive Digraphs

Since every transitive digraph is also quasi-transitive, every comparability graph has a quasi-transitive orientation. The next theorem by Ghouila-Houri shows that the other direction also holds.

Theorem 11.1.1 *[404] A graph G has a quasi-transitive orientation if and only if it has a transitive orientation.*

Proof: To illustrate the usefulness of the decomposition theorem for quasi-transitive digraphs (Theorem 2.7.5), we give a proof which is quite different from the one in [404]. We prove the non-trivial part of the statement by induction on the number of vertices. When $n \leq 3$ the claim is easily verified, so we proceed to the induction step, assuming $n \geq 4$. Suppose D is a quasi-transitive orientation of G and that D is not transitive. If D is not strongly connected, then, by Theorem 2.7.5, we can decompose D as $D = T[W_1, W_2, \ldots, W_t]$, $t = |V(T)| \geq 2$, where T is transitive and each W_i is a strong quasi-transitive digraph. As $t \geq 2$ it follows by induction that we can reorient each $UG(W_i)$ as a transitive digraph T_i, $i = 1, 2, \ldots, k$. This gives a transitive orientation $D' = T[T_1, T_2, \ldots, T_t]$ of G.

Suppose now that D is strong. By Theorem 2.7.5, D can be decomposed as $D = S[W_1, W_2, \ldots, W_s]$, $s = |V(S)| \geq 2$, where S is a strong semicomplete digraph and each W_i is either a single vertex or a non-strong quasi-transitive digraph. It follows by induction (as above) that we can orient each $UG(W_i)$ as a transitive digraph T'_i, $i = 1, 2, \ldots, s$. Let TT_s be the transitive tournament on s vertices. Then $D' = TT_s[T'_1, T'_2, \ldots, T'_s]$ is a transitive orientation of G. □

The following construction is due to Ghouila-Houri [404]. Let $G = (V, E)$ be an undirected graph. Construct a graph G_{qtd} from G as follows: $V(G_{qtd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} in G_{qtd} precisely if $w = v$ and $uz \notin E$, or $u = z$ and $vw \notin E$. In particular, there is an

edge $x_{uv}x_{vu}$ for each $uv \in E$. See Figure 11.1 for an illustration of this construction. Note that if $x_{uv}x_{vw}$ is an edge of G_{qtd}, then so is $x_{wv}x_{vu}$. Every edge of G_{qtd} corresponds to a forbidden pair of oriented edges of G. The interest in this construction lies in the following very useful fact.

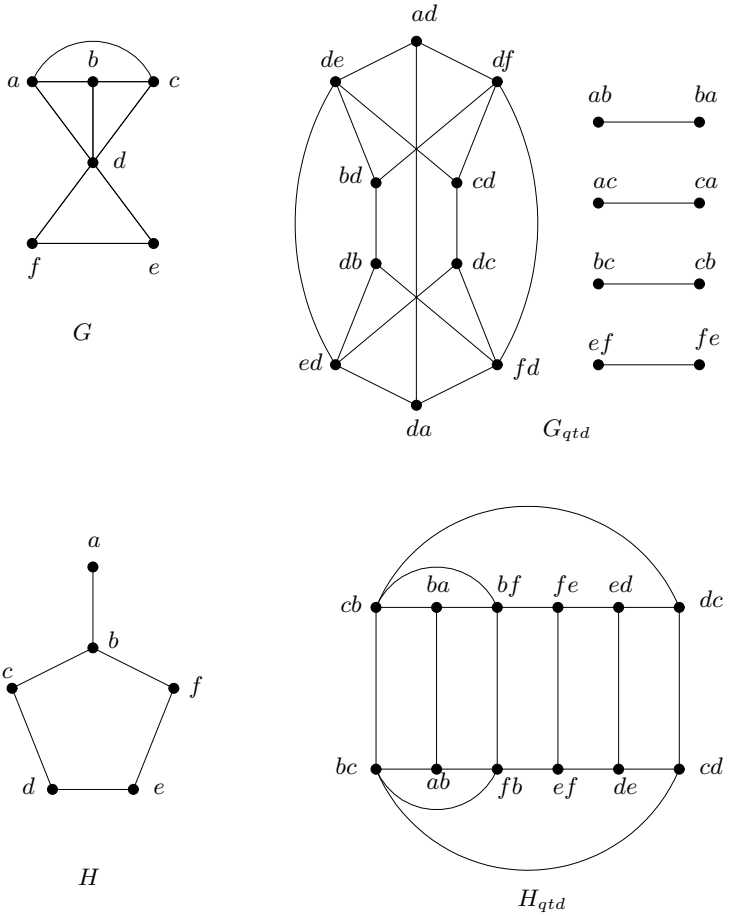


Figure 11.1 An illustration of the construction of G_{qtd} for two graphs. Due to space considerations we have dropped the x's in the name of the vertices of G_{qtd}, H_{qtd}. The graph G is a comparability graph. The graph H is not a comparability graph. Note that bf, cb, dc, ed, fe, bf is a 5-cycle in H_{qtd}.

Theorem 11.1.2 *[404] A graph G is a comparability graph (and hence has a transitive orientation) if and only if G_{qtd} is bipartite.*

Proof: Suppose $G = (V, E)$ is a comparability graph and let $T = (V, A)$ be a transitive orientation of G. In G_{qtd} the vertices X_1 corresponding to the

arcs of T (that particular orientation of the edge uv for each $uv \in E$) form an independent set. By symmetry of the definition of the edges of G_{qtd}, the remaining vertices X_2 of G_{qtd} also induce an independent set. Hence G_{qtd} is bipartite with bipartition (X_1, X_2).

Conversely, suppose that G_{qtd} is bipartite with bipartition (X, Y). Because G_{qtd} contains a perfect matching consisting of edges of the form $x_{uv}x_{vu}$ it follows that $|X| = |Y|$ and X contains precisely one of the vertices x_{uv}, x_{vu} for each $uv \in E$. It follows from the definition of G_{qtd} that orienting the edges corresponding to the vertices in X (Y) results in a quasi-transitive orientation D of G[1]. By Theorem 11.1.1, G has a transitive orientation. □

Corollary 11.1.3 *Comparability graphs can be recognized in time $O(\Delta m)$, where m is the number of edges in the input graph.*

Proof: This follows from Theorem 11.1.2 and the fact that the number of edges in G_{qtd} is $O(\Delta|E|)$. Note that we can check whether a given undirected graph is bipartite in linear time using BFS (Exercise 3.2). □

For various results on recognition of comparability graphs see the papers [420] by Golumbic, [513] by Hell and Huang, [706] by Morvan and Viennot and [709] by Muller and Spinrad.

Consider the comparability graph G in Figure 11.1 and suppose that our goal is to obtain a quasi-transitive orientation of G. If we choose the orientation $a \to d$, then this forces the edge between d and e to be oriented as $e \to d$. This in turn forces the orientations $c \to d$ and $b \to d$ and each of these force $f \to d$. Similarly it can be seen that the five edges ad, bd, cd, de, df force each other. It is easy to see that the corresponding ten vertices in G_{qtd} form one connected component of G_{qtd}.

It is not difficult to see that this observation holds for arbitrary comparability graphs, i.e., if x_{uv} and x_{wz} are in the same connected component of G_{qtd} and $wz \neq vu$, then, once we decide on an orientation for the edge uv in G, that orientation forces one on the edge wz. An **implication class** for $G = (V, E)$ is a maximal set of edges E' with the property that in every orientation of G as a quasi-transitive digraph the choice of an orientation of one edge $e \in E'$ forces the orientation of all other edges in E'.

By our remark above the implication classes for G coincide with the connected components of G_{qtd}. More precisely the connected component C of G_{qtd} corresponds to the implication class $E' = \{uv \in E : x_{uv} \in V(C)\}$. It is not difficult to see that the implication classes form a partition of E. Given G_{qtd} we can obtain the implication classes of G just by finding the connected components of G_{qtd}. Hence we can find the implication classes in time $O(\Delta m)$ (recall that G_{qtd} has $O(\Delta m)$ edges).

Let G be a comparability graph and suppose we want to find a transitive orientation of G. We can obtain a quasi-transitive orientation just by picking

[1] If $x_{uv} \in X$, then orient uv from u to v, otherwise orient it from v to u.

an arbitrary edge from each implication class, choosing an orientation for this edge and then orient the remaining edges in that class the way they are forced to be oriented. The problem is that this orientation will in general not be transitive. Consider for example the graph G in Figure 11.1. Since each of the edges ab, bc and ac forms an implication class of size one, there is nothing that prevents us from orienting these three edges as the 3-cycle $a \to b \to c \to a$.

We now describe a simple and very useful technique, due to Hell and Huang [513], for obtaining a transitive orientation of a given comparability graph G. Let $1, 2, \ldots, n$ be a fixed labelling of the vertices of G. We say that a vertex x_{ij} of G_{qtd} is **lexicographically smaller** than a vertex x_{rs} if either $i < r$ or $i = r$ and $j < s$.

The **lexicographic 2-colouring** of G_{qtd} is the unique 2-colouring (on colours A, B) which is obtained as follows. Mark all vertices of G_{qtd} non-coloured. Next, as long as there are uncoloured vertices, choose the lexicographically smallest vertex x_{ij} which is not coloured yet and colour it A. Colour all other vertices in the same connected component as they are forced (that is, by A if the distance from x_{ij} is even and by B otherwise). When all vertices of G_{qtd} are coloured the process stops.

The usefulness of lexicographic 2-colourings comes from the following result (see also Theorem 11.1.9).

Theorem 11.1.4 *[513] Let G be a comparability graph with vertex set $\{1, 2, \ldots, n\}$ and let $f : V(G_{qtd}) \to \{A, B\}$ be the lexicographic 2-colouring of $V(G_{qtd})$. Define an orientation D of G such that an edge ij is oriented as $i \to j$ precisely when x_{ij} receives colour A by the colouring f. Then D is a transitive orientation of G.*

Proof: Exercise 11.3. □

Note that if we apply the lexicographic 2-colouring procedure to a non-comparability graph, then this will be discovered after G_{qtd} has been formed when we try to 2-colour a non-bipartite connected component H of G_{qtd}. The algorithm will discover that H is not bipartite and hence G does not have any orientation as a quasi-transitive digraph. Thus we have obtained another proof of Theorem 11.1.1 (the lexicographic 2-colouring algorithm either finds a transitive orientation of G, or concludes that G has no quasi-transitive orientation).

The whole algorithm (including the construction of G_{qtd}) can be performed in time $O(\Delta m)$, where m is the number of edges of G, since we can find the connected components of G_{qtd} using BFS.

11.1.2 Underlying Graphs of Locally Semicomplete Digraphs

For a given proper circular-arc graph G, with a prescribed circular-arc representation, we get a natural order on the vertices of G by fixing a point on

the circle and labelling the vertices $v_1, v_2, \ldots, v_n$ according to the clockwise ordering of the right endpoints of their intervals (circular arcs) on the circle with respect to this point. Since every proper circular-arc graph has a representation in which no two arcs cover the whole circle [421], we may assume that we are working with such a representation. Now it is not difficult to see that the following process leads to a round local tournament orientation of G (see Chapter 2 for the definition of a round local tournament[2]): orient the edge between v_i and v_j from v_i to v_j just if the left endpoint of the jth interval is contained in the ith interval. Thus we have the following result due to Skrien (see also [66, 512, 539]):

Proposition 11.1.5 *[823] Every proper circular arc graph has an orientation as a round local tournament.* □

In fact, Hell and Huang showed that the other direction holds as well.

Theorem 11.1.6 *[513] A connected graph is a proper circular arc graph if and only if it is orientable as a round local tournament.*

Proof: We proved one direction above. To prove the other direction assume that D is a round local tournament and that $v_1, v_2, \ldots, v_n$ is a round enumeration of $V(D)$. If no such labelling is given, then we can find one in time $O(n+m)$ (Exercise 11.5). Now represent $UG(D)$ by circular arcs as follows. Let ϵ be a fixed number such that $0 < \epsilon < 1$. Make an n-scale-clock on a cycle and associate with the vertex v_i the circular arc from i to $i + d_D^+(i) + \epsilon$ in the clockwise order for each $i \in [n]$ (indices modulo n). It is easy to check that this gives a proper circular arc representation of $UG(D)$. Note that here we use the fact that the out-neighbours of every vertex of D induce a transitive tournament (see Chapter 2) to see that no arc is properly contained in any other arc. □

By Theorem 11.1.6, the class of underlying graphs of locally semicomplete digraphs contains the class of proper circular arc graphs. The next result, due to Skrien [823] (see also [512, 539]), says that there are no other graphs that can be oriented as locally semicomplete digraphs.

Theorem 11.1.7 *[823] The underlying graphs of locally semicomplete digraphs are precisely the proper circular arc graphs.* □

Hell, Bang-Jensen and Huang [512] showed that, just as in the case of comparability graphs, there is a useful auxiliary graph G_{ltd} related to orientations as a local tournament digraph: Let $G = (V, E)$ be given and define G_{ltd} as follows: $V(G_{ltd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} precisely if $v = z$ and $uw \notin E$, or $u = w$ and $vz \notin E$. Furthermore, the edge $x_{uv}x_{vu}$ is in $E(G_{ltd})$ for each $uv \in E$. The proof of the following result is left as Exercise 11.6.

[2] Hell and Huang use the name **local transitive tournament** instead of round local tournament [513].

Theorem 11.1.8 *[512] The graph G has an orientation as a local tournament digraph if and only if the graph G_{ltd} is bipartite.* □

Suppose G is a proper circular arc graph. Then it follows from Theorems 11.1.7 and 11.1.8 that G_{ltd} is bipartite. Again each connected component of G_{ltd} corresponds to an implication class E' of edges of G. Hence we can find a local tournament orientation of G by fixing the orientation of one arc from each implication class arbitrarily and then giving all remaining arcs the forced orientation.

If our goal is to find a representation of G as a proper circular arc graph, then we are not interested in just any local tournament orientation of G, but we need an orientation as a round local tournament (compare with Theorem 11.1.6). Again we can use the lexicographic method from Section 11.1.1: Since G_{ltd} is bipartite, we can apply the lexicographic 2-colouring procedure which was defined in Section 11.1.1. It follows from the next theorem and the proof of Theorem 11.1.6 that the lexicographic method is also of use in recognition of proper circular arc graphs.

Theorem 11.1.9 *[513] Let G be a proper circular arc graph and let $f : V(G_{ltd}) \rightarrow \{A, B\}$ be the lexicographic 2-colouring of $V(G_{ltd})$. Define an orientation D of G such that an edge ij is oriented as $i \rightarrow j$ precisely when x_{ij} receives colour A by the colouring f. Then D is a round local tournament orientation of G.* □

This shows that using the lexicographic method one can obtain an $O(\Delta m)$ algorithm for recognizing and representing proper circular arc graphs.

In fact an even faster and optimal algorithm for recognizing proper circular arc graphs has been found by Deng, Hell and Huang [257]. This algorithm also uses the fact that a graph is a proper circular arc graph if and only if it has an orientation as a round local tournament.

Theorem 11.1.10 *[257] There is an $O(n + m)$ algorithm to find a local tournament orientation of a graph G or to report that G does not admit such an orientation. Moreover, if a local tournament orientation exists, the algorithm also identifies all balanced arcs.* □

We will define the notion of a balanced arc in the next subsection.

11.1.3 Local Tournament Orientations of Proper Circular Arc Graphs

In this subsection we describe a deep result by Huang [538, 539] which gives a complete characterization of all the possible local tournament orientations of a given proper circular arc graph. In order to state Theorem 11.1.12 below we need several definitions.

Let $G = (V, E)$ be an undirected graph. An edge xy of G is **balanced** if every vertex $z \in V - \{x, y\}$ is adjacent to both or none of x and y. An edge is **unbalanced** if it is not balanced. If all edges of G are unbalanced, then G is **reduced** and otherwise G is **reducible** . It follows from this definition that a graph which is not reduced can be decomposed as described in the next lemma. See Figure 11.2 for an illustration.

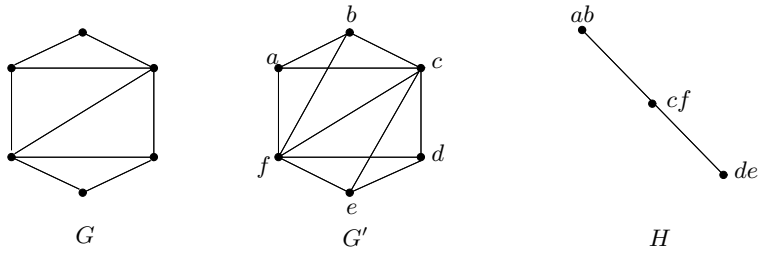


Figure 11.2 A reduced graph G and a reducible graph G'. The graph G' can be reduced to the graph H by identifying the pairs $\{a, b\}$, $\{c, f\}$ and $\{d, e\}$.

Lemma 11.1.11 *If G is not a reduced graph, i.e., it has a balanced edge, then there exist a reduced subgraph H of G and complete subgraphs $K_{a_1}, K_{a_2}, \ldots, K_{a_h}$ of G such that*[3] *$G = H[K_{a_1}, K_{a_2}, \ldots, K_{a_h}]$, $h = |V(H)|$. Furthermore we can find this (unique) decomposition in time $O(n^3)$.*

Proof: We leave the easy proof to the reader. □

Actually such a decomposition can be found even faster in $O(n^2)$ time, see the paper [290] by Ehrenfeucht, Gabow, McConnell and Sullivan.

Let $G = (V, E)$ be a proper circular arc graph. As we mentioned in the last subsection, one can partition E into disjoint non-empty subsets $E_1, \ldots, E_r$ with the property that if we fix the orientation of one edge in each E_i, then there is precisely one way to orient all the remaining edges in E so that the resulting digraph is a local tournament digraph. In other words, the orientation of one edge in E_i implies the orientation of all other edges in E_i. As in the last section we call the sets $E_1, \ldots, E_r$ the implication classes of G (see Theorems 11.1.12 and 11.1.13 below).

Theorem 11.1.12 *[539, Huang] Let G be a connected proper circular arc graph and let $C_1, \ldots, C_k$ be the connected components of $\overline{G}$. Then one of the following two statements holds.*

[3] Here the **composition** $H[G_1, G_2, \ldots, G_{|V(H)|}]$ is defined analogously to the composition of digraphs in Section 1.3.

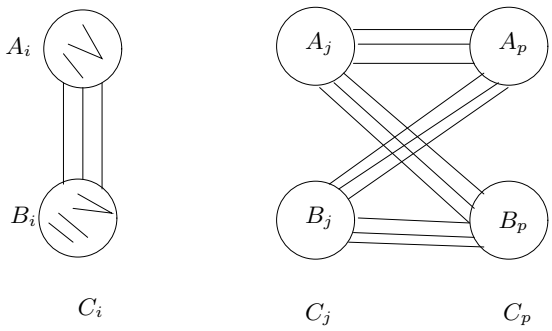


Figure 11.3 Implication classes for orientations of a graph G as a local tournament digraph.The sets C_i, C_j, C_p denote distinct connected components of $\overline{G}$. For each component a bipartition A_r, B_r is shown. The edges shown inside C_i form one implication class and the edges shown between C_j and C_p form another implication class.

(a) $\overline{G}$ is bipartite, the set of all unbalanced edges of G with both ends in a fixed C_i form an implication class and the set of all unbalanced edges of G between two distinct C_i and C_j form an implication class (see Figure 11.3).
(b) $\overline{G}$ is not bipartite, $k = 1$ and all unbalanced edges of G form one implication class. □

Observe that an edge forms an implication class by itself if and only if it is balanced. Hence Theorem 11.1.12 can be reformulated as follows.

Theorem 11.1.13 (Huang) *[539] Let G be a proper circular arc graph which is reduced (that is, every edge is unbalanced), let $\bar{G}$ denote the complement graph of G and let $C_1, \ldots, C_k$ denote the connected components of $\bar{G}$.*

(a) If $\bar{G}$ is not bipartite, then $k = 1$ and (up to a full reversal) G has only one orientation as a locally tournament digraph, namely, the round orientation.
(b) If $\bar{G}$ is bipartite, then every orientation of G as a locally tournament digraph can be obtained from the round locally tournament digraph orientation D of G by repeatedly applying one of the following operations:
(I) reverse all arcs in D that go between two different C_i's,
(II) reverse all arcs in D that have both ends inside some C_i. □

It is also possible to derive a similar result characterizing all possible orientations of G as a locally semicomplete digraph. We refer the reader to [539] for the details.

As an example of the power of Huang's result (Theorems 11.1.12 and 11.1.13) we state and prove the following corollary which was implicitly stated in [539] (see also Exercise 2.34).

Corollary 11.1.14 *If D is a locally tournament digraph such that $\overline{UG(D)}$ is not bipartite, then $D = R[S_1, \ldots, S_r]$, where R is a round locally tournament digraph on r vertices and each S_i is a strong tournament.*

Proof: If $UG(D)$ is reduced, then this follows immediately from Theorem 11.1.13, because according to Theorem 11.1.13, there is only one possible locally tournament digraph orientation of $UG(D)$. So suppose that $UG(D)$ is not reduced. By Lemma 11.1.11, $UG(D) = H[K_{a_1}, \ldots, K_{a_h}]$, $h = |V(H)|$, where H is a reduced proper circular arc graph, each K_{a_i} is a complete graph and some $a_i \geq 2$. Because we can obtain an isomorphic copy of H as a subgraph of $UG(D)$ by choosing an arbitrary vertex from each K_{a_i}, we conclude, from Theorem 11.1.12, that in D all arcs between two distinct K_{a_i}, K_{a_j} have the same direction (note that $\overline{H}$ is non-bipartite). Thus $D = R[S_1, \ldots, S_r]$, where (up to reversal of all arcs) R is the unique round locally tournament digraph orientation of H and each S_i is the tournament $D\langle V(K_{a_i})\rangle$. Note that $D\langle V(K_{a_i})\rangle$ may not be a strong tournament, but according to Corollary 2.10.7 we can find a round decomposition of D so that this is the case. □

11.1.4 Underlying Graphs of Locally In-Semicomplete Digraphs

The structure of the underlying graphs of locally in-tournament digraphs is more complicated than in the case of local tournaments and quasi-transitive digraphs

In [882][4] an algorithm is given for recognizing graphs orientable as locally in-tournament digraphs (as well as finding a locally in-tournament digraph orientation if one exists). The complexity is $O(nm)$ which is worse than the simple algorithm based on 2-satisfiability given in Proposition 11.1.15 below.

The proposition below gives an illustration that algorithms for the 2-SAT problem (see Section 17.5) are useful for certain orientation problems.

Proposition 11.1.15 *[105] There is an $\mathcal{O}(\Delta m)$ algorithm for recognizing graphs that are orientable as locally in-tournament digraphs.*

Proof: Let a graph $G = (V, E)$ be given, and let $D = (V, A)$ be an arbitrary orientation of the edges of G, where $A = \{a_1, a_2, \ldots, a_m\}$. If a_i is an orientation of an edge yz of G, then the reverse orientation of that edge is denoted by $\overleftarrow{a_i}$. We now construct an instance of the 2-SAT problem as follows. The set of variables is $X = \{x_1, \ldots, x_m\}$. The variables are interpreted as follows. If $x_i = 1$, then we keep the orientation a_i, otherwise we take the opposite orientation $\overleftarrow{a_i}$. The clauses consist of those pairs of literals $(\ell_i + \ell_j)$ for which $\overline{\ell_i}, \overline{\ell_j}$ correspond to arcs with the same terminal vertex and non-adjacent initial

[4] In [882] Urrutia and Gavril studied locally in-tournament digraphs under another name, **fraternally oriented graphs**.

vertices in D. It is easy to see that G is orientable as a locally in-tournament digraph if and only if the above-defined instance of 2-SAT is satisfiable. By Theorem 17.5.5 the complexity of 2-SAT is $O(K)$, where K is the number of clauses. Hence, it follows from the way we construct the clauses above that we can recognize graphs orientable as locally in-tournament digraphs in time $O(\Delta m)$. □

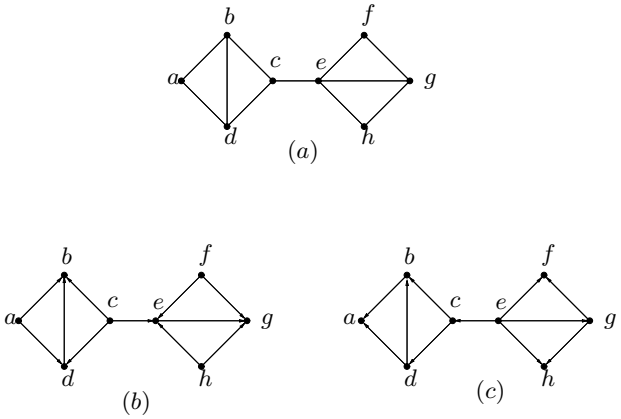


Figure 11.4 An undirected graph G and two orientations of G.

The construction used in the proof above is illustrated in Figure 11.4. Part (a) shows an undirected graph G; part (b) an arbitrary orientation D of G. The instance of 2-satisfiability corresponding to this orientation contains one variable for each arc of D and the following clauses:

$$(\bar{x}_{ab} + \bar{x}_{cb}), (\bar{x}_{ad} + \bar{x}_{cd}), (x_{cb} + x_{ce}), (x_{cd} + x_{ce}), (\bar{x}_{ce} + \bar{x}_{fe}),$$
$$(\bar{x}_{ce} + \bar{x}_{he}), (\bar{x}_{fe} + \bar{x}_{he}), (\bar{x}_{fg} + \bar{x}_{hg}), (\bar{x}_{ce} + x_{eg}).$$

Part (c) shows an orientation of G as an in-tournament digraph corresponding to the truth assignment $(x_{ab}, x_{ad}, x_{cb}, x_{cd}, x_{ce}, x_{db}, x_{eg}, x_{fe}, x_{fg}, x_{he}, x_{hg}) = (0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0)$.

In Exercise 18.13 a useful correspondence between the 2-SAT problem and the problem of deciding the existence of an independent set of size $n/2$ in graphs with a perfect matching is indicated. Using this correspondence, it is no surprise that for graphs which are orientable as in-tournament digraph there is a construction similar to the one used in Theorem 11.1.2 for comparability graphs. (In Theorem 11.1.8 we saw a similar one for the underlying graphs of locally semicomplete digraphs.)

Let $G = (V, E)$ be an undirected graph and define the undirected graph G_{itd} as follows: $V(G_{itd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} precisely if $w = v$ and $z = u$, or $v = z$ and $uw \notin E$.

The proof of the following lemma is left to the reader as Exercise 11.1. It is useful to compare this lemma with Exercise 18.13.

Proposition 11.1.16 *A graph $G = (V, E)$ is orientable as a locally in-tournament digraph if and only if the graph G_{itd} has an independent set of size $|E|$.* □

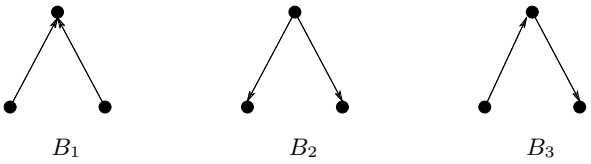


Figure 11.5 The digraphs B_1, B_2, B_3.

Let $\mathcal{B}$ be the family of the three digraphs shown in Figure 11.5 and let F be any subset of $\mathcal{B}$ other than $\{B_1\}$ or $\{B_2\}$. Skrien [823] characterized the classes of those graphs which can be oriented without a member of F as an induced subdigraph. These are the classes of complete graphs, comparability graphs, proper circular arc graphs and nested interval graphs. Since each of the forbidden configurations contains just two arcs, 2-SAT could be used to solve the recognition problem for each of these four classes, all in time $O(\Delta m)$.

11.2 Orientations with No Even Cycles

The problem of deciding whether a given digraph has an even cycle is polynomially solvable, but very complicated (see Section 8.3). The corresponding problem for undirected graphs is easy (see Exercise 11.14). Here we will consider a somewhat opposite orientation problem where we wish to achieve orientations with no even cycles. Since we can concentrate on strong components when looking for even cycles, we only consider strong orientations without even cycles. Clearly we can also concentrate on graphs that are non-bipartite since otherwise every cycle will be even and the answer is trivial. It is also clear that it suffices to consider graphs which are 2-connected.

Let G be an undirected graph and let us call an orientation D of G **odd** if there is no directed cycle of even length in D. The following problem was posed by Bang-Jensen in 1992 (see, e.g., [400]).

Problem 11.2.1 *Is there a polynomial algorithm which given an undirected graph G either returns a strong odd orientation D of G or a proof (in the form of a certificate that can be checked in polynomial time) that G has no such orientation?*

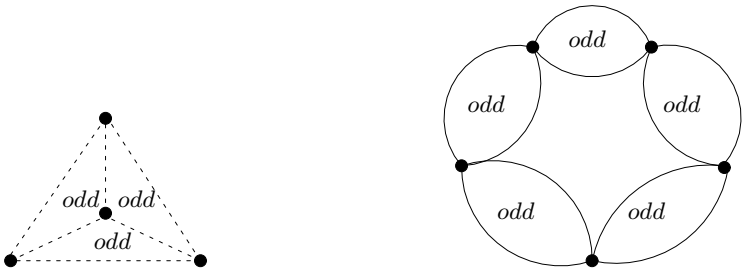


Figure 11.6 Illustration of an odd-K_4 and an odd necklace. Each of the six dashed lines in the odd-K_4 in part (a) corresponds to internally disjoint paths and the word *odd* inside a cycle in part (b) indicates that the length of the bounding cycle is odd.

This seems to be a very hard problem and so far only a partial answer (Theorem 11.2.3 below) is known. In order to state Theorem 11.2.3, we need the following definitions. An **odd-K_4** is an undirected graph which is a subdivision of the complete graph on four vertices in which each of the four 3-cycles of K_4 becomes odd cycles (see Figure 11.6(a)). An **odd necklace** is any undirected graph which can be obtained from an odd number t of odd cycles $C_1, C_2, \ldots, C_t$ by identifying one vertex of C_i with one vertex of C_{i+1} (modulo t) in such a way that $|V(C_i) \cap V(C_j)| = 1$ if $|i-j| = 1 \pmod{k}$ and $|V(C_i) \cap V(C_j)| = 0$ otherwise (see Figure 11.6(b)).

The proof of the following lemma is left as Exercise 11.13.

Lemma 11.2.2 *[400] Let G be a graph which is either an odd-K_4 or an odd necklace. Then every strong orientation of G has an even cycle.* □

However, graphs that contain odd-K_4's may have strong odd orientations as shown in Figure 11.7. Note that in this orientation the 2-connected subgraph corresponding to the odd-K_4 is not oriented as a strong digraph.

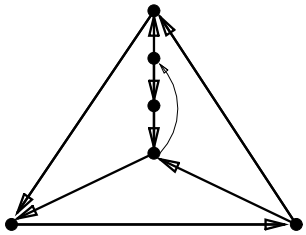


Figure 11.7 A strong odd orientation of a graph with an odd-K_4 (shown as fat arcs).

Gerards and Shepherd proved the following result:

Theorem 11.2.3 *[400] Let G be 2-connected and non-bipartite. If G contains neither an odd-K_4 nor an odd necklace as a subgraph, then G has a strong odd orientation.* □

By Lemma 11.2.2, Theorem 11.2.3 can be reformulated as follows:

Theorem 11.2.4 *[400] Let G be an undirected graph. Then each 2-connected non-bipartite subgraph of G has a strong odd orientation if and only if G contains neither an odd-K_4 nor an odd necklace as a subgraph.* □

The proof of Theorem 11.2.3 is based on a constructive characterization of graphs with no odd-K_4's and no odd necklaces [400, Theorem 7, Corollary 8] (see also [398]).

It is shown in [400] that graphs which contain no odd-K_4 and no odd necklace can be recognized in polynomial time. Furthermore the proof of Theorem 11.2.3 in [400] is constructive and implies that there is a polynomial algorithm for Problem 11.2.1 for graphs with no odd-K_4 and no odd chain.

For further results on orientations of graphs with no odd-K_4 see the papers [397, 399] by Gerards.

How many edges can a graph G have before every strong orientation of G has an even cycle? Since every strong orientation of a complete graph on n vertices is pancyclic by Theorem 1.5.1 it is clear that there is some upper bound on the number of edges (as a function on n) for graphs which have strong orientations without even cycles.

Let A and B be disjoint sets of size $\lfloor (n-1)/2 \rfloor$ and $\lceil (n-1)/2 \rceil$, respectively. Form a graph H_n by taking $V(H_n) = A \cup B \cup v$, where v is a new vertex and $E(H_n) = \{ab : a \in A, b \in B\} \cup \{vc : c \in A \cup B\}$. Then $|E(H_n)| = \lfloor (n+1)^2/4 \rfloor - 1$ and we can orient H_n so that it is strong and all cycles are 3-cycles just by orienting all arcs from v to A, from A to B and from B to v.

Let $C_n = v_0 v_1 \dots v_{n-2} v_{n-1} v_0$ be a cycle. Let L_n be obtained from C_n by adding all chords $v_i v_j$ such that $i - j$ is a positive even number. It is not difficult to check that each graph L_n has $|E(L_n)| = \lfloor (n+1)^2/4 \rfloor - 1$ and that L_n has a strong orientation with no even cycles (Exercise 11.15).

These two classes show that the following result, due to Chung, Goddard and Kleitman, is best possible in terms of the number of edges. We formulate it as a theorem for oriented graphs.

Theorem 11.2.5 *[219] Every strong oriented graph on n vertices and at least $\lfloor (n+1)^2/4 \rfloor = f(n)+1$ arcs contains an even cycle. Furthermore every strong oriented graph D on n vertices with $f(n)$ arcs which has no even cycle consists of a maximal hamiltonian arc-critical subdigraph H of D on an odd number $(2r+1$, for some $r)$ of vertices and an acyclic bipartite tournament B on the remaining vertices, having the partite sets as equal in size as possible, each vertex of which is joined to $r+1$ vertices of H.* □

By a maximal hamiltonian arc-critical subdigraph of D we mean a subdigraph on, say, n' vertices which has $f(n')$ edges, is hamiltonian, and is maximal with respect to these conditions (that is, every subdigraph of D with $n'' > n'$ vertices is either non-hamiltonian or has less than $f(n'')$ arcs).

Although Theorems 11.2.3 and 11.2.5 do give some information as to which graphs have strong orientations without even cycles, there are large classes of graphs for which they give no information. One such class is the cubic graphs one can obtain by joining two odd cycles of the same length by a perfect matching. The Petersen graph[5] is one of these graphs. It is easy to see that the Petersen graph (an orientation of which is shown in Figure 11.8) contains an odd-K_4 and hence is not covered by Theorem 11.2.3. In Exercise 11.12 the reader is asked to prove that every strong orientation of the Petersen graph contains an even cycle.

Obviously an oriented graph has an even cycle if it has two cycles whose length differ by one. Hence the following problem may be interesting to study. The analogous problem was considered for undirected graphs by Bondy and Vince in [172].

Problem 11.2.6 *Is there a polynomial algorithm to decide whether a given 2-connected graph has a strong orientation without two cycles whose length differ by one?*

11.3 Colourings and Orientations of Graphs

In this section we discuss connections between a very important parameter of an undirected graph G, its chromatic number, and properties of orientations of G.

Recall that the chromatic number of an undirected graph $G = (V, E)$, denoted $\chi(G)$, is the smallest natural number k for which V can be partitioned into disjoint independent sets $V_1, V_2, \ldots, V_k$. A more popular and obviously equivalent definition is that $\chi(G)$ is the smallest number k such that we can assign each vertex $v \in V$ a colour from the set $[k]$ without ever using the same colour for vertices that are adjacent (joined by an edge) in G. A ***k*-colouring** of an undirected graph G is any function mapping $V(G) \to [k]$. A k-colouring is **proper** if $f(u) \neq f(v)$ for every edge $uv \in E(G)$. For convenience we also define the chromatic number of a digraph as $\chi(D) = \chi(UG(D))$.

For an arbitrary digraph $\mathrm{lp}(D)$ denotes the length of a longest path in D. The first relation we will discuss is between the numbers $\mathrm{lp}(D)$ and $\chi(G)$ for an arbitrary orientation D of G.

[5] The Petersen graph, due to the Danish pioneer of graph theory, Julius Petersen (1839-1910), is very important in several problems on undirected graphs (see, e.g., [902]).

If $\chi(G) = k$, then we can obtain an acyclic orientation D of G with $\mathrm{lp}(D) = k - 1$ just by orienting all edges between V_i and V_j from V_i to V_j for all $1 \leq i < j \leq k$, where $V_1, V_2, \ldots, V_k$ is a partition of V into k disjoint independent sets. Hence if $\chi(G)$ is small, then G has an orientation without long directed paths. The interesting thing is that the opposite direction also holds as was discovered independently by Gallai, Roy and Vitaver.

Theorem 11.3.1 (Gallai-Roy-Vitaver theorem) *[387, 789, 887] For every digraph D, $\chi(D) \leq \mathrm{lp}(D) + 1$.*

Proof: This is an easy consequence of Proposition 9.9.5 and the notation introduced there. Indeed, if $\chi(D) = n$, then we must have $p(\mathcal{F}^*) \geq n - 1$, where $\mathcal{F}^*$ is any spanning out-forest minimizing the vector v. Since each vertex in L_i, $i > 1$, has an in-coming arc from L_{i-1} we can trace back a path $x_0 x_1 \ldots x_{n-1}$ of length $n - 1$ by starting from a vertex $x_{n-1} \in L_{n-1}$. This shows that $lp(D) \geq \chi(D) - 1$ □

Gallai asked [387] whether every graph G has an orientation with precisely one path of length $\chi(G)$. This is not true, as shown by an example by Youngs [924]. For a detailed discussion of this topic and related problems see the book by Jensen and Toft [564].

An alternative formulation[6] of Theorem 11.3.1 is that the chromatic number of a graph is given by

$$\chi(G) = \min\{\mathrm{lp}(D) + 1 : D \text{ is an orientation of } G\}.$$

For any orientation D of an undirected graph G, we obtain an upper bound k on $\chi(G)$ from Theorem 11.3.1. It follows from the fact that the problem of finding the minimum k such that an undirected graph has a k-colouring is $\mathcal{NP}$-hard (as shown by Karp [585]) that it is an $\mathcal{NP}$-hard problem to find an orientation D of a given undirected graph G which minimizes $\mathrm{lp}(D)$. The next theorem by Tuza shows that given an orientation D of G, in linear time, one can find a colouring using at most $\mathrm{lp}(D) + 1$ colours.

Theorem 11.3.2 *[880] If D is a digraph such that $\mathrm{lp}(D) < k$, then a proper k-colouring of $UG(D)$ can be found in time $O(n + m)$.* □

By the Gallai-Roy-Vitaver theorem, the directed path of length $k - 1$ is contained in every k-chromatic digraph. More generally, one can ask which digraphs are contained in every k-chromatic digraph. Such digraphs are called ***k*-universal.** It follows from the well-known result of Erdős [296], showing that k-chromatic graphs may have arbitrarily high girth, that k-universal graphs must be oriented trees.

[6] Combining this with the famous 4-Colour Theorem by Appel and Haken [43] which says that every planar graph has chromatic number at most four, we see that the 4-Colour Theorem is equivalent to the statement that every planar graph has an orientation such that no directed path has length more than 3.

Burr proved the existence of a function $f(n)$ such that every tree on n vertices is $f(n)$-universal. To prove this, we need the following lemma.

Lemma 11.3.3 *[183] If $\Delta^+(D) \leq k$, then $\chi(D) \leq 2k+1$.*

Proof: Clearly D contains a vertex v whose in-degree is also at most k and removing v from D we can repeat the argument, so $UG(D)$ is $2k$-degenerate[7] which immediately implies the claim. □

Theorem 11.3.4 *[183] Every oriented tree on $n \geq 3$ vertices is $(n-1)^2$ - universal.*

Proof: We proceed by induction on n. The case $n = 3$ is easily seen to hold. Now assume that the theorem holds for some $n \geq 3$ and let T be an oriented tree on $n+1$ vertices. By considering the converse of T if necessary, we may assume that T contains some arc uv where v has in-degree 0 in T. Let $T' = T - v$. Now let $D = (V, A)$ be an arbitrary digraph with $\chi(D) \geq n^2$ and partition V into $V_1, V_2 = V - V_1$ where $V_1 = \{v \in V | d^+(v) < n\}$. By Lemma 11.3.3 $\chi(D\langle V_1\rangle) \leq 2n-1$ and hence $\chi(D\langle V_2\rangle) \geq (n-1)^2$. By the induction hypothesis, $D\langle V_2\rangle$ contains T'. Let u be a vertex in $D\langle V_2\rangle$ corresponding to u in some copy of T' in $D\langle V_2\rangle$. As $d^+(u) \geq n$ we can find a vertex $z \in V(D) - V(T')$ so that u dominates z. Now $T' + z$ is the desired copy of T. □

Burr conjectured that $f(n)$ is, in fact, much smaller.

Conjecture 11.3.5 *[183] Every oriented tree on $n \geq 3$ vertices is $2(n-1)$-universal.*

Clearly this would be best possible as an out-tree on k vertices with root r and $k-1$ vertices dominated by the root is not contained in any $(k-2)$-regular tournament. Conjecture 11.3.5 contains as a special case the following longstanding conjecture due to Sumner (see [912]) which has attracted considerable attention (see, e.g., [489, 508]).

Conjecture 11.3.6 (Sumner) *Every tournament on at least $2n-2$ vertices contains every oriented tree on n vertices.*

Currently the best known bound for Sumner's conjecture is due to El-Sahili [293] who proved that every oriented tree is contained in every tournament of order at least $3n-3$.

Bondy obtained the following generalization of Theorem 11.3.1 to strong digraphs. Note that Camion's theorem is a direct consequence of Theorem 11.3.7.

[7] An undirected graph G is **p-degenerate** if and only if every induced subgraph of G has a vertex of degree at most p.

Theorem 11.3.7 *[167] Every strong digraph contains a directed cycle of length at least* $\chi(D)$. □

This result was generalized as follows by Addario-Berry, Havet and Thomassé.

Theorem 11.3.8 *[6] Let* D *be a strong digraph with* $\chi(D) = r$. *For every* $3 \leq k \leq r$, D *contains a cycle* C *of length at least* k *such that* $\chi(D\langle V(C)\rangle) \leq k$. □

Denote by $P(k,l)$ the oriented path consisting of k forward arcs followed by l backward arcs for some $k, l \geq 1$. Such a path is called a **path with two blocks**. Note that $P(k,l)$ and $P(l,k)$ are isomorphic. El-Sahili conjectured in [292] that every path with two blocks and $r \geq 4$ vertices is r-universal. This was proved by Addario-Berry, Havet and Thomassé.

Theorem 11.3.9 *[6] Let* k, l, r *be positive integers so that* $k+l+1 = r \geq 4$ *and let* D *be an* r*-chromatic digraph. Then* D *contains a copy of* $P(k,l)$. □

The proof of Theorem 11.3.9 is too long to be included here but in order to illustrate the usefulness of Proposition 9.9.5 we will prove the following weaker result due to El-Sahili and Kouider.

Theorem 11.3.10 *[294] Every digraph* D *contains either a* $P(k, l-1)$ *or a* $P(k-1, l)$ *for every choice of positive integers* k, l *so that* $k+l = \chi(D)$. *In particular* D *contains a copy of* $P(k-1, l-1)$.

Proof: By Proposition 9.9.5, D contains a spanning out-forest $\mathcal{F}$ with levels $L_0, L_1, \ldots, L_p$ such that each L_i is an independent set. Thus D is $(p+1)$-colourable and we have $p \geq \chi(D)-1 = k+l-1$. For $j = 0, \ldots, l-1$ let $X_j =$[8] $\bigcup_{r\geq 0} L_{k-1+j+rl}$. Then $(L_0, L_1, \ldots, L_{k-2}, X_0, X_1, \ldots, X_{l-1})$ is a partition of $V(D)$ into $k+l-1$ sets. As $\chi(D) = k+l$, some X_j contains adjacent vertices x, y. Note that $x \in L_{k-1+j+sl}$ and $y \in L_{k-1+j+tl}$ where $s \neq t$ as each L_i is independent and we may assume $s < t$. Denote by T_x, T_y the out-trees in $\mathcal{F}$ which contain x, y, respectively, possibly $T_x = T_y$. Let Q be the path from the root of T_x to x and let R be the path to y from its ancestor in L_{k+j+sl}. Then Q has length $k-1+j+sl \geq k-1$ and R has length $(t-s)l-1 \geq l-1$. Thus D has a path $P(k, l-1)$ if $xy \in A$ or a path $P(k-1, l)$ if $yx \in A$. □

The case when D is strong in Theorem 11.3.9 also follows easily from Theorem 11.3.8 and Proposition 9.9.5. To see this let D and $k, l \geq 1$ with $k+l = \chi(D)-1$ be given. By Theorem 11.3.8, D contains a directed cycle C of length at least $k+1$ so that $\chi(D\langle V(C)\rangle) \leq k+1$. Let $D_1 = D\langle V(C)\rangle$ and let D_2 be obtained from D by contracting C to one vertex c and then reversing

[8] That is, X_j contains those levels of $\mathcal{F}$ whose index is at least $k-1$ and is congruent to $k-1+j$ modulo l.

all remaining arcs. It is not difficult to see that the proof of Proposition 9.9.5 can be modified to show that D_2 has a spanning out-forest $\mathcal{F}$ where all levels are independent and $L_0 = \{c\}$. Since $\chi(D\langle V - V(C)\rangle) \geq \chi(D) - (k+1) \geq l$ it follows that $\mathcal{F}$ has at least $l+1$ levels and thus D_2 contains a path of length l starting in c. Taking the reverse of that path back in D and a path of length k ending in c in D_1 we obtain the desired copy of $P(k,l)$ in D.

Analogously to the definition of a path with two blocks, a **cycle with two blocks** $C(k,l)$ is a digraph consisting of two vertices x, y and two internally disjoint (x,y)-paths of lengths k, l, respectively.

Benhocine and Wojda [136] proved that every tournament on $n \geq 4$ vertices contains every $C(k,l)$ with $k,l \geq 1$ and $k+l = n$. This does not hold for general digraphs as shown by an example by Gyárfás and Thomassen (see [6]). However, the digraphs in the example by Gyárfás and Thomassen are not strongly connected and it is easy to see that every strongly connected digraph of order n which is not a directed cycle contains a $C(k,l)$ for some $k,l \geq 1$.

Addario-Berry, Havet and Thomassé asked whether we can always find, in a strong digraph D, a pair of vertices that are joined by two internally disjoint paths which are both long, where the measure is with respect to the chromatic number of D.

Problem 11.3.11 *[6] Let D be an n-chromatic strongly connected digraph ($n \geq 4$) and let k,l be positive integers such that $k+l = n$. Does D contain a $C(k',l')$ for some integers $k' \geq k$ and $l' \geq l$?*

Minty showed that one can also measure the chromatic number of a graph by how much one can balance oriented cycles in orientations.

Theorem 11.3.12 *[700] If G has an orientation such that every oriented cycle contains at least $|V(C)|/k$ arcs in each direction, then $\chi(G) \leq k$.* □

This was strengthened by Tuza as follows.

Theorem 11.3.13 *[880] If G has an orientation such that every cycle of length $|V(C)| \equiv 1 \pmod{k}$ contains at least $|V(C)|/k$ arcs in each direction, then $\chi(G) \leq k$.* □

For more relations between chromatic number and paths and cycles in digraphs see Bondy's survey [169, Section 4.4] and the paper [839] by Szigeti and Tuza.

11.4 Orientations and Nowhere-Zero Integer Flows

In this section, unless otherwise stated, we assume that all undirected multigraphs in question are connected.

Let $G = (V, E)$ be an undirected multigraph. A ***k*-flow** on G is an assignment of an orientation a to each edge $e \in E$ as well as an integer $x(a)$ from the set $[k-1]$ such that for each vertex v the sum of the values of x on arcs into v equals the sum of the values of x on arcs leaving v. That is, x is a circulation in the resulting oriented multigraph D. Hence we can think of a k-flow on a multigraph G as a pair (D, x) where $D = (V, A)$ is an orientation of G and x is an integer circulation in D with the property that $x(a) \in [k-1]$ for each $a \in A$. Below we use this notation. The flow x is sometimes called a **nowhere-zero *k*-flow** to stress the fact that x never takes the value zero on an arc. We say that G has a k-flow if there exists a k-flow on G. It is easy to see that a multigraph G has a k-flow for some k if and only if each connected component of G has a k-flow. Furthermore, it is easy to show that a connected multigraph with a bridge cannot have a k-flow for any k (see Exercise 11.19). It is easy to see that a pseudograph G has a k-flow if and only if the multigraph H that we obtain by deleting all loops from G has a k-flow. This is why we assume that we are working with a multigraph rather than a pseudograph below.

For convenience, we will always specify the value of a flow x on an arc uv by $x(uv)$, rather than x_{uv} as we did in Chapter 4. We start with a very easy result on 2-flows.

Proposition 11.4.1 *A multigraph G has a 2-flow if and only if all degrees of G are even.*

Proof: Clearly, if G has a 2-flow x, then all degrees are even, since x is a circulation which only takes the value 1. Suppose now that all degrees of G are even. We may assume that G is connected as otherwise we consider each component in turn. By Euler's theorem, G has a closed walk $W = w_0w_1w_2w_3 \dots w_{m-1}w_m$, where $w_0 = w_m$ which uses each edge precisely once. Let D be the orientation obtained by orienting the edge w_iw_{i+1} from w_i to w_{i+1} for $i = 0, 1, \dots, m-1$. Then $(D, x \equiv 1)$ is a 2-flow in G. □

For any abelian[9] group $(\Gamma, +)$ we can define a flow in a multigraph $G = (V, E)$ as follows. A ***Γ*-flow** in G is a pair (D, x) where D is an orientation of G, x maps $A(D)$ to the non-zero elements $\{g_1, g_2, \dots, g_{|\Gamma|-1}\}$ of Γ and x satisfies

$$\sum_{uv \in A(D)} x(uv) = \sum_{vw \in A(D)} x(vw) \qquad \text{for all } v \in V, \tag{11.1}$$

where addition is in the group Γ and $|\Gamma|$ denotes the number of elements in the group Γ. That is, x is a circulation which takes values from $\Gamma - g_0$, where g_0 is the neutral element of $(\Gamma, +)$.

Tutte proved the following important theorem, relating k-flows on a multigraph G to arbitrary group-valued circulations on orientations of G.

[9] Recall that an additive group $(\Gamma, +)$ is **abelian** if $a + b = b + a$ holds for all elements a, b of Γ.

Theorem 11.4.2 (Tutte) *[877] If $(\Gamma, +)$ is a finite abelian group, then an undirected multigraph G has a Γ-flow if and only if it has a k-flow, where $k = |\Gamma|$.* □

An important step in proving Theorem 11.4.2 is to demonstrate the following theorem by Tutte. Although we do not prove Theorem 11.4.2, we still prove Theorem 11.4.3 and then use it below. The group $\mathcal{Z}_k$ is the additive group of integers modulo k.

Theorem 11.4.3 *[877] Let $G = (V, E)$ be an undirected multigraph and $k \geq 1$ an integer. Then G has a k-flow if and only if G has a $\mathcal{Z}_k$-flow.*

Proof: If (D, x) is a k-flow in G, then $x(a) \in [k-1]$ for each $a \in A$ and

$$\sum_{uv \in A(D)} x(uv) - \sum_{vw \in A(D)} x(vw) = 0 \equiv 0 \qquad \text{(modulo } k\text{)}.$$

Hence (D, x) is also a $\mathcal{Z}_k$-flow in G.

Suppose now that (D', x') is a $\mathcal{Z}_k$-flow in G. Since all calculations are modulo k, we may assume that $x'(a) \in [k-1]$ for each $a \in A$. By the definition of a $\mathcal{Z}_k$-flow we also have

$$\sum_{uv \in A(D')} x'(uv) - \sum_{vw \in A(D')} x'(vw) \equiv 0 \text{ (modulo } k\text{)}.$$

For a given $\mathcal{Z}_k$-flow $(D = (V, A), x)$, we let the balance vector b_x be defined as in (4.5), that is,

$$b_x(v) = \sum_{vw \in A(D)} x(vw) - \sum_{uv \in A(D)} x(uv).$$

Now assume that (D', x') is chosen among all $\mathcal{Z}_k$-flows in G such that the sum

$$\phi(D', x') = \sum_{v \in V(D')} |b_{x'}(v)| \tag{11.2}$$

is minimized. We show that $\phi(D', x') = 0$, implying that (D', x') is a k-flow in G. Suppose this is not the case. Then let

$$P = \{v \in V : b_{x'}(v) > 0\}, M = \{v \in V : b_{x'}(v) < 0\}.$$

It follows from standard flow considerations (compare with Section 4.1) that $P, M \neq \emptyset$. By Theorem 4.3.1, we conclude that there is a path Q from P to M in D'. Let (D'', x'') be obtained by reversing all arcs of Q and changing the flow of each arc $a \in A(Q)$ to $k - x'(a)$ while leaving the flow on all arcs not on Q unchanged. It is easy to see that (D'', x'') is a $\mathcal{Z}_k$-flow in G and that $\phi(D'', x'') = \phi(D', x') - 2k$ (which is still at least zero since every vertex in

P (M) contributes a positive (negative) multiple of k to the balance vector). This contradicts the choice of (D', x') and hence we must have $\phi(D', x') = 0$ implying that (D', x') is a k-flow. □

The usefulness of Theorem 11.4.2 is illustrated several times below. The point is that, as we shall see below, it is sometimes considerably easier to establish that a multigraph has a Γ-flow than it is to prove directly that it has a $|\Gamma|$-flow.

A multigraph is **cubic** if every vertex has degree 3.

Proposition 11.4.4 *A cubic multigraph G has a 3-flow if and only if G is bipartite.*

Proof: Suppose first that G is cubic and bipartite with bipartition (X, Y). Let D be the orientation obtained by orienting all edges from X to Y. Let $x \equiv 1$, then (D, x) is a $\mathcal{Z}_3$-flow in G. By Theorem 11.4.3, G has a 3-flow (D', x').

Suppose now that G is cubic and has a 3-flow (D, x). Since the only values of x are 1 and 2, it is easy to see that taking X (Y) as those vertices which are the tail (head) of an arc whose x-value is 2, we obtain a partition of $V(G)$ into two independent sets. Thus G is bipartite with bipartition (X, Y). □

A multigraph G is ***r*-edge-colourable** if one can assign each edge a number from the set $[r]$ in such a way that all edges incident to the same vertex receive different numbers. Such an assignment is also called an ***r*-edge-colouring** of G. By Exercise 4.58, every cubic bipartite multigraph is 3-edge-colourable. For general 3-edge-colourable cubic multigraphs it may not be possible to find a 3-flow (see Exercise 11.26), but one can always find a 4-flow as the next result shows.

Theorem 11.4.5 *A cubic multigraph G has a 4-flow if and only if G is 3-edge-colourable.*

Proof: By Theorem 11.4.2, G has a 4-flow if and only if it has a $\mathcal{Z}_2 \times \mathcal{Z}_2$-flow[10]. Observe that the non-zero elements of $\mathcal{Z}_2 \times \mathcal{Z}_2$ are their own inverses. Furthermore these three elements sum up to the zero element in $\mathcal{Z}_2 \times \mathcal{Z}_2$. This shows that at every vertex of G precisely one edge has flow equal to $(1,0), (0,1)$ and $(1,1)$, respectively. Thus if (D, x) is a $\mathcal{Z}_2 \times \mathcal{Z}_2$-flow in G, then we can consider the elements $(0,1), (1,0), (1,1)$ as edge colours and we obtain that G is 3-edge-colourable. This argument works the other way also and hence the claim is proved. □

Theorem 11.4.6 *A multigraph G has a 4-flow if and only if it contains two eulerian subgraphs G_1, G_2 such that $E(G) = E(G_1) \cup E(G_2)$.*

[10] The additive group $(\mathcal{Z}_2 \times \mathcal{Z}_2, +)$ has elements $\{(0,0), (1,0), (0,1), (1,1)\}$ and addition is coordinate-wise.

Proof: Exercise 11.25. □

Theorem 11.4.7 *[560] Every 4-edge-connected multigraph G has a 4-flow.*

Proof: Let $G = (V, E)$ be 4-edge-connected. By Theorem 9.4.2, G has two edge-disjoint spanning trees T_1, T_2. Every edge $e \in E - E(T_1)$ forms a unique cycle C_e with $E(T_1)$. Let E_1 be the modulo 2 sum of the edge sets of all cycles of the form C_e, $e \in E - E(T_1)$. Then the subgraph G_1 of G induced by E_1 is eulerian and contains all edges of $E - E(T_1)$. Similarly there is an eulerian subgraph G_2 which contains all edges of $E - E(T_2)$. Hence $E(G) = E(G_1) \cup E(G_2)$, because T_1 and T_2 are edge-disjoint, and the claim follows from Theorem 11.4.6. □

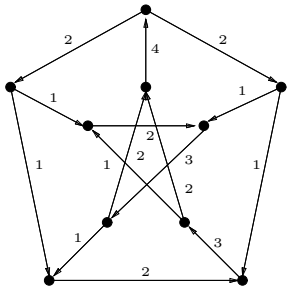


Figure 11.8 The Petersen graph with a 5-flow (D, x) indicated. Notice that the value 4 is only used once.

By Theorem 11.4.5 and the existence of 2-edge-connected cubic multigraphs which are not 3-edge-colourable (the most famous example being the Petersen graph, see Figure 11.8 for an orientation of the Petersen graph) we conclude that not all 2-edge-connected multigraphs have a 4-flow. However, Tutte conjectured that 4 can be replaced by 5.

Conjecture 11.4.8 (Tutte's 5-flow conjecture) *[877] Every multigraph which is 2-edge-connected has a 5-flow.*

The next lemma (described as a folklore result by Seymour in [809]) shows that it is sufficient to prove the conjecture for multigraphs which are cubic and 3-connected.

Lemma 11.4.9 *If $k \geq 3$ and $G = (V, E)$ is a 2-edge-connected multigraph which does not have a k-flow, but every 2-edge-connected multigraph $H = (V'', E'')$ with $|V''| + |E''| < |V| + |E|$ has a k-flow, then G is cubic and 3-connected.*

Proof: Suppose first that G has a cut-vertex z such that $V - z$ has connected components $H_1, \ldots, H_p$, $p \geq 2$. By the minimality of G, each of the multigraphs $H_i + z$, $i \in [p]$, has a k flow and using these we easily obtain a k-flow for G. Hence we may assume that G is 2-connected.

Suppose $\{e, e'\}$ is a 2-edge-cut in G. Let $e = st$ and let $U' \cup W'$ be a bipartition of V such that $s \in U'$, $t \in W'$ and there is no edge between U' and W' in $G - \{e, e'\}$. Let $U = U' - s$ and $W = W' - t$. By the definition of U, W and the fact that G has no cut-vertex there is precisely one edge between U and W in G, namely, e'. Now let the multigraph $G' = (V', E')$ be obtained from G by contracting e into one vertex v_e and deleting the loop created this way. Since $|V'|+|E'| < |V|+|E|$ and contraction cannot decrease edge-connectivity, it follows from the assumption on G that there is a k-flow (D', x') in G'.

In D' we may assume without loss of generality that e' is oriented as an arc a' from W to U. Let $r = x'(a')$. Since x' is a circulation the following must hold:

$$\sum_{w \in W} x'(v_e w) - \sum_{w' \in W} x'(w' v_e) = r,$$
$$\sum_{u \in U} x'(v_e u) - \sum_{u' \in U} x'(u' v_e) = -r.$$

In $G - e$ the vertex s (t) is adjacent only to vertices in U (W). Let D'' be the orientation obtained by using the orientations prescribed by D' on the edges of G and orienting the edge st from s to t. Define x'' by $x''(a) = x'(a)$ for all arcs except st where we take $x''(st) = r$. Then (D'', x') is a k-flow in G, contradicting the assumption. Hence it follows that G is 3-edge-connected.

If G has a vertex s of degree at least 4, then it follows from a result of Fleischner [319] (see Exercise 11.35) that s has neighbours u, v so that replacing the edges su, sv by the edge uv we obtain a 2-edge-connected multigraph G^*[11]. By the minimal choice of G, there is a k-flow (D^*, x^*) in G^* and it is easy to obtain a k-flow in G from this (just replace the arc between u and v in D^* by a path of length 2 via s in G, using the two edges su, sv and send the appropriate amount of flow along that path). This contradicts the choice of G and hence we conclude that G is cubic. It follows from Exercise 11.20 that G is 3-connected. □

A major breakthrough on Tutte's 5-flow conjecture came when Jaeger [560] proved that every 2-edge-connected multigraph has an 8-flow. His proof was surprisingly short and elegant. The reader is asked to give a proof of Jaeger's result in Exercise 11.28.

The strongest result so far is due to Seymour.

[11] In the language of Section 14.1 the result says that there is a feasible splitting su, sv (with respect to 2-edge-connectivity) for some pair of neighbours u, v of s.

Theorem 11.4.10 *[809, Seymour] Every 2-edge-connected multigraph has a nowhere-zero 6-flow.* □

Since the proof is based on arguments that do not involve directed graphs, we will not give the proof in detail here (see Seymour's original paper [809] or the books by Diestel [258] and Fleischner [321]). It follows from Lemma 11.4.9 that it suffices to prove the result for 3-connected cubic multigraphs. Seymour proves that the edge set of such a multigraph G can be covered by two multigraphs G_1, G_2 such that G_1 is eulerian and G_2 has a 3-flow x'. It follows from Proposition 11.4.1 that G_1 has a 2-flow. Since $E(G) = E(G_1) \cup E(G_2)$ it is easy to obtain a $\mathcal{Z}_2 \times \mathcal{Z}_3$-flow in G using x, x' and hence, by Theorem 11.4.2, G has a 6-flow.

An algorithmic version of Seymour's proof, leading to a polynomial algorithm for finding a 6-flow in any 2-edge-connected multigraph, was given by Younger [923].

Bienia, Goddyn, Gvozdjak, Sebő and Tarsi proved the following interesting result. The case when $k \geq 5$ is an obvious consequence of Theorem 11.4.10.

Theorem 11.4.11 *[159] If G has a nowhere-zero flow with at most $k-1$ distinct values, then G has a k-flow.* □

For much more information on nowhere-zero flows we refer the reader to the books by Fleischner [321] and Jensen and Toft [564], the papers [561, 562] by Jaeger as well as [810] by Seymour. In particular, Chapter 13 in the book by Jensen and Toft [564] contains a lot of useful information about the subject and the important open problems.

11.5 Orientations Achieving High Arc-Strong Connectivity

Let us recall that an orientation D of a multigraph $G = (V, E)$ is obtained by assigning one of the two possible orientations to each edge of G (in particular, two parallel edges may receive opposite orientations).

By Robbins' theorem, a multigraph $G = (V, E)$ has a strongly connected orientation if and only if G is 2-edge-connected. Below we describe two generalizations of Robbins' theorem, due to Nash-Williams, both of which are much deeper than Robbins' theorem, especially the one in Theorem 11.5.4.

11.5.1 k-Arc-Strong Orientations

In order to illustrate to usefulness of the splitting technique which is discussed in Chapter 14, we prove Theorem 11.5.3 below using a splitting result for undirected graphs. This theorem, due to Lovász, is analogous to Theorem

14.1.2. The reader is asked to prove this theorem in Exercise 11.34. Analogously to the directed case, we denote by $\lambda(x, y)$ the maximum number of edge-disjoint xy-paths in G and we say that a graph $G = (V + s, E)$ with a special vertex s is k-edge-connected in V if $\lambda(x, y) \geq k$ holds for all $x, y \in V$.

Theorem 11.5.1 (Lovász's splitting theorem) *[655] Let $G = (V + s, E)$ be a multigraph with a designated vertex s of even degree and suppose that G is k-edge-connected in V, for some $k \geq 2$. Then for every edge st there exists an edge su such that after splitting off the pair (st, su) the new graph is still k-edge-connected* [12] *in V.* □

An undirected multigraph $G = (V, E)$ is **minimally k-edge-connected** if G is k-edge-connected ($\lambda(G) = k$), but $\lambda(G - e) = k - 1$ for every edge $e \in E$. The following theorem by Mader is analogous to Theorem 5.6.4. The proof is left to the reader as Exercise 11.33.

Theorem 11.5.2 *[664] Every minimally k-edge-connected multigraph has a vertex of degree k.* □

Now we can prove the following famous result of Nash-Williams:

Theorem 11.5.3 (Nash-Williams' orientation theorem) *[716] An undirected multigraph $G = (V, E)$ has a k-arc-strong orientation D if and only if G is $2k$-edge-connected.*

Proof: The proof idea used below is due to Lovász [655]. Suppose G has a k-arc-strong orientation D. Thus for every non-empty proper subset X of V we have $d_D^+(X), d_D^-(X) \geq k$. This implies that in G we have $d(X) \geq 2k$ and hence, G is $2k$-edge-connected.

To prove the other direction, we proceed by induction on the number of edges in G. Let $G = (V, E)$ be $2k$-edge-connected. If $|E| = 2k$, then G is just two vertices x, y joined by $2k$ copies of the edge xy. Clearly this multigraph has a k-arc-strong orientation. Thus we may proceed to the induction step. Since adding arcs to a directed multigraph cannot decrease its arc-strong connectivity, it suffices to consider the case when G is minimally $2k$-edge-connected.

By Theorem 11.5.2, G contains a vertex s such that $d_G(s) = 2k$. Apply Lovász's splitting theorem to G with s as the special vertex and conclude that we can pair off the $2k$ edges incident to s in G in k pairs $(su_1, sv_1), \ldots, (su_k, sv_k)$ in such a way that deleting s and adding the edges $u_1v_1, \ldots, u_kv_k$ to $G - s$ results in a $2k$-edge-connected graph H. Since H has fewer edges than G it follows by induction that H has an orientation D' which is k strong.

By Exercise 14.7, we can obtain a k-arc-strong orientation of G by adding the arcs $u_1s, u_2s, \ldots, u_ks$ and the arcs $sv_1, sv_2, \ldots, sv_k$ to H. □

[12] As for directed graphs (see Section 14.1), **splitting off** the pair (su, sv) means that we replace the edges su, sv by a new edge uv (or a copy of that edge if it already exists).

11.5.2 Well-Balanced and Best-Balanced Orientations

Following Nash-Williams, let us say that an orientation D of a multigraph G is **well-balanced** if $\lambda_D(x,y) \geq \lfloor \frac{\lambda_G(x,y)}{2} \rfloor$ for every ordered pair of vertices $x, y \in V(G)$.

Following Király and Szigeti [595], we say that an orientation of a multigraph is **smooth** if the in-degree and out-degree of every vertex differ by at most one and a smooth well-balanced orientation of a multigraph G is called a **best-balanced** orientation of G. Nash-Williams proved the following much stronger result which clearly contains Theorem 11.5.3 as a special case.

Theorem 11.5.4 (Nash-Williams' strong orientation theorem) *[716] Every multigraph G has a best-balanced orientation D.* □

It is beyond the scope of this book to give a complete proof here. The original proof by Nash-Williams [716] is quite complicated and so are alternative proofs by Mader (using a local edge-connectivity version of Theorem 11.5.1 [668]) and Frank [343]. It remains a real challenge to find a short and transparent proof for this important theorem.

We will outline the main idea of Nash-Williams' proof (the two other proofs use the same approach). The first observation is that if G is eulerian, then the statement is easy to prove (Exercise 11.31). So we may assume that G is not eulerian. We can make it eulerian by adding any matching on the odd degree vertices. Such a matching will be called a **pairing** of G. If we could find a pairing M so that after orienting $G + M$ as an eulerian digraph D' and then removing the arcs corresponding to M, we still have

$$\lambda_D(x,y) = \lfloor \lambda_G(x,y)/2 \rfloor \qquad \text{for all } x, y \in V, \tag{11.3}$$

where[13] $D = D' - M$, then we would obtain the desired orientation since D is clearly smooth.

Let us see which conditions the pairing M should satisfy in order to give rise to the desired orientation D as above. Following Frank [343], we use the notation $\tilde{f} = 2\lfloor f/2 \rfloor$ whenever f is an integer-valued function. Let R be defined as follows: $R(\emptyset) = R(V) = 0$ and for every $\emptyset \neq X \neq V$ we let $R(X) = \max\{\lambda_G(x,y) : x \in X, y \in V - X\}$. We call R the **requirement function** for G. Let $b_G(X) = d_G(X) - \tilde{R}_G(X)$ for all $X \subseteq V$. By Menger's Theorem for undirected edge-connectivity (11.3) is equivalent to requiring that

$$d^-_D(X) \geq \tilde{R}_G(X)/2 \qquad \forall X \subset V. \tag{11.4}$$

A pairing M is **feasible** if

$$d_M(X) \leq b_G(X) \qquad \forall X \subset V. \tag{11.5}$$

[13] By this we mean the oriented graph obtained from D by removing the arcs corresponding to M.

Here $d_M(X)$ denotes the number of edges from M with precisely one end in X. Suppose M is a feasible pairing for G. Let D' be an eulerian orientation of $G + M$ and let $D = D' - M$. Then we have

$$\begin{aligned} d_D^-(X) &\geq d_{D'}^-(X) - d_M(X) \\ &= (d_G(X) + d_M(X))/2 - d_M(X) \\ &= (d_G(X) - d_M(X))/2 \\ &\geq \tilde{R}_G(X)/2, \end{aligned} \tag{11.6}$$

implying that (11.4) and hence (11.3) holds.

Clearly D is a smooth orientation of G. Thus if we can find a feasible pairing, then we get the desired orientation easily. The main point then is to prove the next theorem.

Theorem 11.5.5 *[716, Nash-Williams] Every undirected multigraph has a feasible pairing.* □

Let M be a pairing of G. An orientation O of M is **good** if

$$d_O^-(X) - d_O^+(X) \leq b_G(X) \qquad \forall X \subset V. \tag{11.7}$$

M is **well-orientable** if there exists a good orientation of M and M is **strong** if every orientation of M is good.

Proposition 11.5.6 *A pairing M of G is strong if and only if M is feasible.*

Proof: Exercise 11.37. □

Thus Theorem 11.5.5 is equivalent to Theorem 11.5.7.

Theorem 11.5.7 *[716, Nash-Williams] Every graph has a strong pairing.* □

11.5.3 Simultaneous Best-Balanced Orientations

Nash-Williams proved the following extension of Theorem 11.5.4.

Theorem 11.5.8 *[716] Every subgraph H of a multigraph G has a best-balanced orientation that can be extended to a best-balanced orientation of G.* □

Király and Szigeti refined this result as follows.

Theorem 11.5.9 *[595] Let $G = (V, E)$ be a multigraph, $\{E_1, E_2, \ldots, E_k\}$ an arbitrary partition of E and $G_i = (V, E_i)$, $1 \leq i \leq k$. Then G has a best-balanced orientation D so that the induced orientations D_i, $i = 1, 2, \ldots, k$, of each G_i are also best-balanced.* □

An easy consequence of Euler's theorem is that the edge set of every multigraph G with $2k$ vertices of odd degree can be decomposed into k paths and some cycles such that the end-vertices of each path are vertices of odd degree in G. The following thus is a direct consequence of Theorem 11.5.9.

Corollary 11.5.10 *[595] Let $P_1, \ldots, P_r, C_1, \ldots, C_s$ be a decomposition of the edge set of $G = (V, E)$ such that each P_i is a path and each C_j is a cycle. There is a best-balanced orientation of G for which each P_i is a directed path and each C_j is a directed cycle.* □

Corollary 11.5.11 *[595] For every partition $\{X_1, X_2, \ldots, X_r\}$ of $V(G)$, there exists a best-balanced orientation D of the multigraph G so that $D\langle X_i\rangle$ is also best-balanced for all $i \in [r]$.* □

11.5.4 Best-Balanced Orientations of Eulerian Multigraphs

Not surprisingly, one can obtain stronger results for eulerian multigraphs. The following results are all due to Király and Szigeti.

For an eulerian multigraph G, an **edge-pairing** at vertex v is an arbitrary partition $P(v) = \{[u_1v, u_2v], [u_3v, u_4v], \ldots, [u_{d(v)-1}v, u_{d(v)}v]\}$ of the edges incident to v into $\frac{d(v)}{2}$ pairs. Suppose we are given a collection of edge pairings $\mathcal{P}(G) = \{P(v) : v \in V\}$, one for each vertex of G. An eulerian orientation D of G is **admissible** with respect to $\mathcal{P}(G)$ if at each vertex v every pair $[u_{2i-1}v, u_{2i}v]$ forms a directed path in D.

Proposition 11.5.12 *[595] Let G be an eulerian multigraph and let an arbitrary edge-pairing $\mathcal{P}(G) = \{P(v) : v \in V\}$ be given. Then there exists an eulerian orientation of G which is admissible with respect to $\mathcal{P}(G)$.*

Proof: Exercise 11.38. □

Theorem 11.5.13 *[595] Every eulerian multigraph $G = (V, E)$ has a best-balanced orientation D such that $D - v$ is a best-balanced orientation of $G - v$ for all $v \in V$.*

Proof: Let G be an eulerian multigraph and define an edge-pairing at vertex $v \in V$ as follows. Take a maximum number of pairs of parallel edges incident to v. Since G is eulerian, the remaining edges incident to v are incident to vertices of odd degree in $G - v$. By Theorem 11.5.7, there exists a strong pairing M_v of $G - v$; M_v induces a pairing of the remaining edges incident to v (those not paired with parallel edges). Thus, taking the pairs from the edge-pairing of parallel edges and those induced from M_v, we obtain an edge-pairing at v. Let $\mathcal{P}(G)$ consist of these edge-pairings for each $v \in V(G)$. By Proposition 11.5.12, there is an eulerian orientation D of G which is admissible with respect to $\mathcal{P}(G)$. For $v \in V$ let $O(M_v)$ be the orientation of M_v defined by D (each edge in M_v inherits the orientation of the pair at v

to which it corresponds). Then $D - v + O(M_v)$ is an eulerian orientation of $G - v$. This implies that $D - v$ is smooth for all v so it remains to prove that it is also well-balanced. Let $O'(M_v)$ denote the converse of $O(M_v)$ for every $v \in V$. Note that $O'(M_v)$ is good because M_v is strong and furthermore $d^-_{D-v}(X) - d^+_{D-v}(X) = d^-_{O'(M_v)}(X) - d^+_{O'(M_v)}(X)$ since $D - v + O(M_v)$ is eulerian. Now, using that $O'(M_v)$ is good, we obtain $d^-_{D-v}(X) - d^+_{D-v}(X) \leq b_{G-v}(X)$ for every X, implying that $D - v$ is well-balanced by Exercise 11.39. □

Corollary 11.5.14 *[595] An eulerian multigraph $G = (V, E)$ has an eulerian orientation D so that $D - v$ is k-arc-strong for all $v \in V$ if and only if $G - v$ is $2k$-edge-connected for all $v \in V$.* □

Corollary 11.5.15 *[595] An eulerian multigraph $G = (V, E)$ has a k-arc-strong orientation D so that $D - v$ is $(k-1)$-arc-strong for all $v \in V$ if and only if G is $2k$-edge-connected and $G - v$ is $(2k-2)$-edge-connected for all $v \in V$.* □

11.6 *k*-Strong Orientations

As indicated in the last section, a substantial amount of results exists on orientations preserving high arc-strong connectivity. On the contrary, not much is known about orientations that achieve high vertex-strong connectivity. The following conjecture, due to Jackson and Thomassen, is a special case of Conjecture 11.10.2.

Conjecture 11.6.1 *[866] Every $2k$-connected graph has a k-strong orientation.*

Very little progress has been made on this conjecture and even the existence of a function $g(k)$ so that every $g(k)$-connected graph has a k-strong orientation is open for $k \geq 3$. For $k = 2$ the result below by Jordán (Theorem 11.6.4) shows that $g(2) \leq 18$.

To see that the bound $2k$ would be best possible, let G be the kth power of an undirected cycle $C = v_1 v_2 \dots v_{2r} v_1$ on $2r > 2k$ vertices. It is not difficult to prove that G is $2k$-connected and that the only separating sets of size $2k$ in G are those obtained by taking two sets of k consecutive vertices on C, each separated by at least one vertex on both sides. From this it follows that if we add the diagonals $v_1 v_{r+1}, v_2 v_{r+2}, \dots, v_r v_{2r}$, then we obtain a $(2k+1)$-connected graph H and it is clear that H cannot have a $(k+1)$-strong orientation, since H is not $2(k+1)$-edge-connected, a condition we know is trivially necessary just to have a $(k+1)$-arc-strong orientation. See Figure 11.9 and Exercise 11.57.

It is also easy to show that a graph which is not $2k$-connected may still have a k-strong orientation. Take for example two copies X, Y of K_{2k+1}, fix

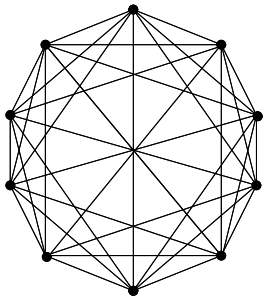


Figure 11.9 A 7-connected 7-regular graph obtained from the third power of a 10-cycle by adding longest diagonals.

a set $\{x_1, x_2, \ldots, x_k\}$ of k vertices in X and similarly $\{y_1, y_2, \ldots, y_k\}$ in Y and add to $X \cup Y$ the edges of the cycle $C = x_1y_1x_2y_2x_3 \ldots x_ky_kx_1$. The resulting graph G has a k-strong orientation by orienting X and Y as k-strong tournaments and C as a directed cycle. Note that G is k-connected and $2k$-edge-connected, but G is not $k+1$-connected as $\{x_1, x_2, \ldots, x_k\}$ is a separating set of size k.

The following conjecture by Frank is still open even for $k = 2$. It attempts a full characterization which would imply Conjecture 11.6.1. Note that for $k = 1$ the conjecture follows from Robbins' theorem. Compare also with Section 11.10.

Conjecture 11.6.2 *[346, Frank] A graph $G = (V, E)$ has a k-strong orientation if and only if $G - X$ is $2(k - j)$-edge-connected for every set X of j vertices $(0 \leq j \leq k)$.*

Corollary 11.5.14 implies that Conjecture 11.6.2 holds for eulerian multigraphs when $k = 2$. Jordán [578] used this result (also obtained be he and Berg in [142]) to obtain a sufficient condition for an arbitrary graph to have a 2-strong orientation (Theorem 11.6.4). His proof also uses the following result for undirected graphs.

Theorem 11.6.3 *[578] Every $6k$-connected graph contains k edge-disjoint 2-connected spanning subgraphs.* □

Theorem 11.6.4 *[578] Every 18-connected graph has a 2-strong orientation.*

Proof: Let $G = (V, E)$ be an arbitrary 18-connected graph. By Theorem 11.6.3, G contains three edge-disjoint 2-connected spanning subgraphs $H_i = (V, E_i)$, $i = 1, 2, 3$. Observe that $H' = (V, E_1 \cup E_2)$ is 4-edge-connected and $H' - v$ is 2-edge-connected for every $v \in V$. This follows easily from the fact that each of H_1, H_2 is 2-connected. Since H_3 is a connected spanning subgraph of G we may use a subset $X \subset E_3$ of edges from H_3 to connect possible vertices of odd degree in H' without introducing new vertices of odd

degree in $H = (V, E_1 \cup E_2 \cup X)$ (Exercise 11.52). Hence H is eulerian and $H - v$ is 2-edge-connected for every $v \in V$. Now it follows from Corollary 11.5.14 that H has a 2-strong orientation. Clearly this implies that G also has one. □

The bound 18 above is not sharp as we do not really need that H_3 is 2-connected, any connected graph would suffice here.

11.7 Orientations Respecting Degree Constraints

In this section we first consider orientations of multigraphs which satisfy prescribed constraints on their semi-degrees. Then we consider the more general case when we have restrictions on certain subsets of the vertices (possibly all proper subsets of the vertex set). A set function f on a ground-set S is **supermodular** if $f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$ holds for every choice of sets $X, Y \subseteq S$. Recall that f is submodular on S if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ holds for every choice of sets $X, Y \subseteq S$. The function f is **modular** if it is both submodular and supermodular[14]. Note that when we are dealing with a function f defined on the vertex set V of a (di)graph H, then we can always extend f to a modular function on 2^V by letting $f(X) = \sum_{v \in X} f(v)$ for every $X \subseteq V$.

11.7.1 Orientations with Prescribed Degree Sequences

We saw in Section 4.11.3 that given a directed multigraph $D = (V, A)$ and numbers $a_1, a_2, \ldots, a_n$ such that $\sum_{i=1}^{n} a_i \leq |A|$, we can use algorithms for maximum flows to decide whether D has a spanning subdigraph D' such that $d^-_{D'}(v_i) = a_i$ for every $i \in [n]$.

We start by showing that we can also solve a similar orientation problem using flows. Namely, given an undirected multigraph $G = (V, E)$, $V = [n]$, and numbers $a_1, a_2, \ldots, a_n$ such that $\sum_{i=1}^{n} a_i = |E|$, does G have an orientation D for which $d^-_D(i) = a_i$, for all $i \in [n]$?

First, form the reference orientation $H = (V, A)$ of G by orienting an edge ij from i to j whenever $i < j$. Form the network $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ by giving each arc of A capacity one and lower bound zero. Let us interpret a feasible integer flow x in $\mathcal{N}$ as an orientation $D' = (V, A')$ of G as follows. If $x_{ij} = 1$, then A' contains the arc ij and otherwise it contains the arc ji. Then for a given flow x we see that for each $i \in [n]$, the vertex i will satisfy

$$d^-_{D'}(i) = \sum_{ji \in A} x_{ji} + (d^+_H(i) - \sum_{ij \in A} x_{ij}).$$

Since we want D' to have in-degree a_i at vertex i, for all $i \in [n]$, we obtain the following restriction on the balance vector b_x of x:

[14] Note that a modular function f with $f(\emptyset) = 0$ satisfies $f(X) = \sum_{x \in X} f(x)$.

$$d_H^+(i) - a_i = \sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji} = b_x(i) \quad \forall i \in [n]. \tag{11.8}$$

Thus we have reduced the orientation problem to that of deciding whether there exists a feasible flow x in $\mathcal{N}$ which has balance vector b_x as in (11.8). Hence, by Lemma 4.2.2, we can use any polynomial algorithm for maximum flow to solve the orientation problem and find the desired orientation if one exists.

Based on the reduction above and the feasibility theorem for flows (Theorem 4.8.4) one may derive necessary and sufficient conditions for the existence of an orientation with a prescribed in-degree sequence (or equivalently, out-degree sequence). One such feasibility theorem which is particularly well-known is for orientations of complete graphs as tournaments. The **score** of a vertex in a tournament is its out-degree. Landau proved the following characterization for score sequences of tournaments (the reader is asked to give a proof in Exercise 11.41):

Theorem 11.7.1 (Landau's theorem) *[634] A sequence $(s_1, s_2, \ldots, s_n)$ of integers satisfying $0 \leq s_1 \leq s_2 \leq \ldots \leq s_n$ is the score sequence of some tournament on n vertices if and only if*

$$\sum_{i=1}^{k} s_i \geq \binom{k}{2} \quad \forall k \in [n],$$

with equality when $k = n$. □

For a very nice collection of different proofs of Landau's theorem we refer the reader to the survey paper [774] by Reid.

Harary and Moser [501] characterized score sequences of strong tournaments.

Theorem 11.7.2 *[501] A sequence $s_1 \leq s_2 \leq \ldots \leq s_n$ of non-negative integers with $n \geq 3$ is the out-degree sequence of some strong tournament if and only if for each j, $1 \leq j \leq n-1$,*

$$\sum_{i=1}^{j} s_i > \binom{j}{2}$$

and

$$\sum_{i=1}^{n} s_i = \binom{n}{2}.$$

□

Below we denote, for an undirected graph $G = (V, E)$ and a subset X of V, the number of edges of E with at least one end (both ends) in X by $e_G(X)$ ($i_G(X)$). Furthermore we denote by $c(G)$ the number of connected components of G. Frank and Gyárfás proved the following theorem which deals with bounds on the in-degrees of an orientation:

Theorem 11.7.3 *[351] Let $G = (V, E)$ be an undirected graph. Let $f : V \to \mathbb{Z}_0$ and $g : V \to \mathbb{Z}_+ \cup \{\infty\}$ be functions on V such that $f \leq g$. Then the following holds:*

(a) There exists an orientation D of G such that

$$d_D^-(v) \geq f(v) \qquad \text{for all } v \in V \tag{11.9}$$

if and only if

$$e_G(X) \geq f(X) \qquad \text{for all } X \subseteq V. \tag{11.10}$$

(b) There exists an orientation D' of G such that

$$d_{D'}^-(v) \leq g(v) \qquad \text{for all } v \in V \tag{11.11}$$

if and only if

$$i_G(X) \leq g(X) \qquad \text{for all } X \subseteq V. \tag{11.12}$$

(c) There exists an orientation D^* of G satisfying both (11.9) and (11.11) if and only if there is one satisfying (11.9) and one satisfying (11.11)[15].

Proof: We consider (a) first. If D satisfies (11.9), then (11.10) follows easily from the following calculation:

$$\begin{aligned} f(X) = \sum_{v \in X} f(v) &\leq \sum_{v \in X} d_D^-(v) \\ &= e_G(X) - d^+(X) \leq e_G(X). \end{aligned} \tag{11.13}$$

Suppose now that (11.10) holds but there is no orientation which satisfies (11.9). Choose D among all possible orientations of G as one which minimizes

$$\sum_{\{v \in V : f(v) > d_D^-(v)\}} (f(v) - d_D^-(v)). \tag{11.14}$$

Let x be a vertex for which $f(x) > d_D^-(x)$. Let X consist of those vertices $u \in V$ for which there is a directed (x, u)-path in D. Note that by the definition of X we have $d_D^+(X) = 0$ or $X = V$. Since $f(X) \leq e_G(X)$, it is easy to see (using that $x \in X$) that there is some vertex $u \in X$ such

[15] Frank calls the phenomenon formulated in part (c) of the theorem the **linking principle** [343, 347].

that $d^-(u) > f(u)$. Let P be any (x,u)-path in D. Let D' be obtained from D by reversing the orientation of every arc on P. Now it is easy to see that D' either satisfies (11.9) or achieves a smaller count for (11.14). This contradiction completes the proof that (11.9) holds.

To prove (b) we proceed as follows. Let $g'(v) = \min\{d_G(v), g(v)\}$ for every $v \in V$. It is easy to see that G has an orientation D satisfying $d_D^-(v) \leq g'(v)$ for all $v \in V$ if and only if it has one satisfying $d_D^-(v) \leq g(v)$ for all $v \in V$. On the other hand, G has an orientation satisfying (11.11) with respect to g' if and only if it has an orientation satisfying (11.9) with respect to $f(v) = d_G(v) - g'(v)$, $v \in V$ (just consider the converse of such an orientation). By (a), such an orientation exists if and only if $e_G(X) \geq f(X)$ for each $X \subseteq V$. Using that $\sum_{x \in X} d_G(x) = e_G(X) + i_G(X)$, we conclude that $e_G(X) \geq f(X)$ if and only if $i_G(X) \leq g'(X)$. This proves (b).

To prove that (c) holds, we choose among all orientations satisfying (11.11) an orientation D which minimizes (11.14). If the sum for this D is zero, then we are done. Otherwise observe that the only vertex whose in-degree is increased by reversing the path P (as in the proof of (a)) is the vertex x for which we have $d_D^-(x) < f(v) \leq g(v)$ and hence we still have $d_{D'}^-(x) \leq g(v)$ and get the same contradiction as in the proof of (a). □

The non-constructive proof above can easily be turned into a polynomial algorithm which finds the desired orientations or a proof that none exists (Exercise 11.42).

We also point out that using the approach from the beginning of this subsection, Theorem 11.7.3 can be proved using flows (Exercise 11.43).

Although Theorem 11.7.3 is fairly simple to prove, it has several consequences. One of these is the marriage theorem which characterizes the existence of a perfect matching in a bipartite graph (Corollary 4.11.4). To see that Theorem 11.7.3 implies the marriage theorem, it suffices to see that a bipartite graph $B = (U, V, E)$ has a perfect matching if and only if it has an orientation D in which every vertex in U has in-degree one and every vertex in $v \in V$ has in-degree $d_B(v) - 1$. We leave the details to the reader as Exercise 11.44. The next result, due to Ford and Fulkerson, can also be derived from Theorem 11.7.3. The proof of this is left as Exercise 11.40.

Corollary 11.7.4 *[331] Let $M = (V, A, E)$ be a mixed graph. Let $G = (V, E)$ be the undirected part and let $D = (V, A)$ be the directed part of M. The edges from G can be oriented so that the resulting directed multigraph[16] is eulerian if and only if $d_G(v) + d_D^-(v) + d_D^+(v)$ is even for each $v \in V$ and the following holds:*

$$d_G(X) \geq d^-(X) - d^+(X) \qquad \text{for all } X \subseteq V. \tag{11.15}$$

□

[16] Recall that a mixed graph may have an edge and an arc with the same end-vertices.

The following common generalization of Robbins' theorem (Theorem 1.6.1) and Theorem 11.7.3 was obtained by Frank and Gyárfás in [351].

Theorem 11.7.5 *[351] Let $G = (V, E)$ be an undirected graph which is 2-edge-connected. Let $f : V \to \mathcal{Z}_0$ and $g : V \to \mathcal{Z}_+ \cup \{\infty\}$ be functions on V such that $f \leq g$. Then the following holds:*

(a) There exists a strong orientation D of G such that

$$d_D^-(v) \geq f(v) \qquad \text{for all } v \in V \tag{11.16}$$

if and only if

$$e_G(X) \geq f(X) + c(G - X) \qquad \text{for all } \emptyset \neq X \subseteq V. \tag{11.17}$$

(b) There exists a strong orientation D' of G such that

$$d_{D'}^-(v) \leq g(v) \qquad \text{for all } v \in V \tag{11.18}$$

if and only if

$$i_G(X) + c(G - X) \leq g(X) \qquad \text{for all } \emptyset \neq X \subseteq V. \tag{11.19}$$

(c) There exists a strong orientation D^* of G satisfying both (11.16) and (11.18) if and only if there is one satisfying (11.16) and one satisfying (11.18). □

11.7.2 Restrictions on Subsets of Vertices

The purpose of this subsection is to study more general problems on orientations with degree conditions on subsets of vertices rather than just the vertices themselves.

Let $G = (V, E)$ be an undirected graph and let $h : 2^V \to \mathbb{Z}_+ \cup \{0\}$ satisfy $h(\emptyset) = h(V) = 0$. The function h is **fully G-supermodular**[17] if

$$h(X) + h(Y) \leq h(X \cap Y) + h(X \cup Y) + d_G(X, Y) \tag{11.20}$$

holds for all pairs of subsets of V (recall that $d_G(X, Y)$ denotes the number of edges in G with one end in $X - Y$ and the other in $Y - X$). If (11.20) is required to hold only for intersecting (crossing) sets, then we say that h is **intersecting (crossing) G-supermodular**. A set function h on G is **symmetric** if $h(X) = h(V - X)$ for every $X \subset V$. The following quite general theorem was proved in [335]. It allows one to find conditions for the existence of k-arc-strong orientations satisfying certain degree constraints on the vertices (see, e.g., [343, page 98]).

[17] This strange-looking definition will be easier to understand when one considers the relation between orientations of mixed graphs and submodular flows in Section 11.9. In particular, see (11.38).

Theorem 11.7.6 (Frank's orientation theorem) *[335] Let G be an undirected graph and let h be a non-negative crossing G-supermodular function on subsets of V. There exists an orientation D of G which satisfies*

$$d_D^-(X) \geq h(X) \qquad \text{for all } X \subset V \tag{11.21}$$

if and only if both

$$e_{\mathcal{F}} \geq \sum_{V_i \in \mathcal{F}} h(V_i) \tag{11.22}$$

and

$$e_{\mathcal{F}} \geq \sum_{V_i \in \mathcal{F}} h(V - V_i) \tag{11.23}$$

hold for every partition $\mathcal{F} = \{V_1, V_2, \ldots, V_t\}$ of V, where $e_{\mathcal{F}}$ denotes the number of edges connecting different V_i's. If h is intersecting G-supermodular, then (11.22) alone is necessary and sufficient. If h is fully G-supermodular, or h is symmetric and crossing supermodular, then it suffices to require (11.22) and (11.23) only for partitions of V into two sets. □

It is an easy exercise to show that Frank's orientation theorem implies Nash-Williams' orientation theorem (Exercise 11.51).

Frank shows in [343] how to derive Theorem 11.7.6 from the theory of submodular flows discussed in Section 11.8. See also Exercise 11.69.

11.8 Submodular Flows

In all of this section we consider set functions which are integer valued and zero on the empty set. The purpose of this section is to introduce a very powerful generalization of flows, due to Edmonds and Giles [286], and to show how many important theorems in graph theory and combinatorial optimization are special cases of this theory.

Let $D = (V, A)$ be a directed multigraph and let $r : A \to \mathbb{R}$ be a function on A. We use the notation

$$r^+(U) = \sum_{a \in (U, \bar{U})} r(a), \qquad r^-(U) = \sum_{a \in (\bar{U}, U)} r(a). \tag{11.24}$$

That is, $r^+(U)$ ($r^-(U)$) is the sum of the r values on arcs leaving (entering) U and $\overline{U} = V - U$[18].

In Chapter 4 it is shown that every feasible flow in a network $\mathcal{N} = (V, A, l, u, b)$ can be modelled as a circulation in an augmented network. Recall that for a circulation x in a network $\mathcal{N}$ we require that for every vertex

[18] Note that the function r^+ is a generalization of d^+ for any directed multigraph D, since taking $r \equiv 1$ we obtain d^+.

v, the flow into v equals the flow out of v. This easily translates to non-empty proper subsets of the vertex set V, i.e., for every circulation x and every non-empty proper subset U of V, $x^-(U) = x^+(U)$. The flows we will consider below do not in general satisfy this property, but there is a bound $b(U)$ on the difference between the flow into U and the flow out of U.

Let $\mathcal{F}$ be a family of subsets of V closed under union and intersection and let $b : \mathcal{F} \to \mathbb{Z} \cup \{\infty\}$ be a function defined on $\mathcal{F}$. The function b is **fully submodular** on $\mathcal{F}$ if the inequality

$$b(X) + b(Y) \geq b(X \cap Y) + b(X \cup Y) \tag{11.25}$$

holds for every choice of members X, Y of $\mathcal{F}$. If (11.25) is only required to hold for intersecting (crossing) members of $\mathcal{F}$, then b is **intersecting (crossing)** submodular on $\mathcal{F}$. By an intersecting (crossing) **pair** $(\mathcal{F}, b)$ we mean a family $\mathcal{F}$ which is intersecting (crossing) and a function b which is submodular on intersecting (crossing) subsets of $\mathcal{F}$.

11.8.1 Submodular Flow Models

Let $f : A \to \mathbb{Z} \cup \{-\infty\}$ and $g : A \to \mathbb{Z} \cup \{\infty\}$ be functions on the arc set of a directed multigraph $D = (V, A)$. Let $\mathcal{F}$ be a family of subsets of V such that $\emptyset, V \in \mathcal{F}$ and let $b : \mathcal{F} \to \mathbb{Z} \cup \{\infty\}$ be fully submodular on $\mathcal{F}$. A function $x : A \to \mathbb{R}$ is a **submodular flow** with respect to $\mathcal{F}$ if it satisfies

$$x^-(U) - x^+(U) \leq b(U) \qquad \text{for all } U \in \mathcal{F}. \tag{11.26}$$

A submodular flow x is **feasible** with respect to f, g if $f(a) \leq x(a) \leq g(a)$ holds for all $a \in A$. The set of feasible submodular flows (with respect to given f, g and $(\mathcal{F}, b)$) form a polyhedron called the **submodular flow polyhedron** $Q(f, g; (\mathcal{F}, b))$ [343].

Submodular flows were introduced by Edmonds and Giles in [286]. In that paper it was only required that the function b is crossing submodular on a crossing family $\mathcal{F}$, something which gives much more flexibility in applications (see Subsection 11.8.4). However, as remarked by Frank in [343] the crossing submodular functions define the same class of polyhedra as do fully submodular functions.

Submodular flow polyhedra have very nice properties which make submodular flows a very powerful tool in combinatorial optimization (see, e.g., Subsection 11.8.4).

Theorem 11.8.1 (Edmonds-Giles theorem) *[286] Let $D = (V, A)$ be a directed multigraph. Let $\mathcal{F}$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}$, let $b : \mathcal{F} \to \mathbb{Z} \cup \{\infty\}$ be crossing submodular on $\mathcal{F}$ with $b(\emptyset) = b(V) = 0$, and let $f \leq g$ be functions on A such that $f : A \to \mathbb{Z} \cup \{-\infty\}$ and $g : A \to \mathbb{Z} \cup \{\infty\}$. The linear system*

$$\{f \le x \le g \text{ and } x^-(U) - x^+(U) \le b(U) \qquad \text{for all } U \in \mathcal{F}\} \tag{11.27}$$

is totally dual integral. That is, if f, g, b are all integer valued, then the linear program $\min\{c^T x : x$ *satisfies* (11.27)$\}$ *has an integer optimum solution (provided it has a solution). Furthermore, if c is integer valued, then the dual linear program has an integer-valued optimum solution (provided it has a solution).* □

In the definition of a submodular flow, we have followed Frank [338, 339, 343, 347, 348, 357] and Schrijver [798]. Sometimes the definition of a submodular flow is slightly different (see, e.g., the original paper by Edmonds and Giles [286] or the book by Fujishige [364]), namely, x is required to satisfy

$$f \le x \le g \text{ and } x^+(U) - x^-(U) \le b(U) \qquad \text{for all } U \in \mathcal{F}. \tag{11.28}$$

There is really no difference in these two definitions, since we see that if x satisfies (11.27), then $-x$ satisfies (11.28) with respect to the same submodular function b and the bounds $-g \le -f$.

One can also use supermodular functions in the definition as shown in the next lemma. Hence there are several models to choose from when one wants to model a problem as a submodular flow problem. Depending on the problem at hand, one model may be easier to use than another. For an illustration of this see Sections 11.8.4 and 13.1, where we use several different definitions.

Lemma 11.8.2 *Let $D = (V, A)$ be a directed multigraph and let $\mathcal{F}$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}$. If p is a crossing supermodular function on $\mathcal{F}$ with $p(\emptyset) = p(V) = 0$, then any $x : A \to \mathbb{R}$ which satisfies*

$$x^-(U) - x^+(U) \ge p(U) \qquad \text{for all } U \in \mathcal{F} \tag{11.29}$$

is a submodular flow.

Proof: To see this, observe that the function $b(U) = -p(\overline{U})$ is crossing submodular on the crossing family $\overline{\mathcal{F}}$ defined as the complements of sets in $\mathcal{F}$. Furthermore, by (11.24), (11.29) is equivalent to $x^-(\overline{U}) - x^+(\overline{U}) \le -p(U) = b(\overline{U})$ for all $\overline{U} \in \overline{\mathcal{F}}$. □

11.8.2 Existence of Feasible Submodular Flows

The following theorem, due to Frank, characterizes when a feasible submodular flow exists with respect to functions f, g and b.

Theorem 11.8.3 (Feasibility theorem for fully submodular flows) *[338] Let $D = (V, A)$ be a directed multigraph, let $f \le g$ be modular functions on A such that $f : A \to \mathbb{Z} \cup \{-\infty\}$ and $g : A \to \mathbb{Z} \cup \{\infty\}$ and let b be a fully submodular function on 2^V. There exists an integer-valued feasible submodular flow if and only if*

$$f^-(U) - g^+(U) \leq b(U) \qquad \text{for all } U \subseteq V. \tag{11.30}$$

In particular there exists a feasible integer-valued submodular flow if and only if there exists any feasible submodular flow.

Proof: We follow the proof by Frank in [343]. Suppose first that there exists a feasible submodular flow x. Then we have $f^-(U) - g^+(U) \leq x^-(U) - x^+(U) \leq b(U)$, showing that (11.30) holds.

Suppose now that (11.30) holds. Define the set function p as follows:

$$p(U) = f^-(U) - g^+(U). \tag{11.31}$$

Claim: The function p is fully supermodular, that is, $p(U) + p(W) \leq p(U \cap W) + p(U \cup W)$ for all $U, W \subseteq V$. Furthermore, equality only holds if $f(a) = g(a)$ for all arcs with one end in $U - W$ and the other in $W - U$.
Proof of Claim: Since f and g are modular as set functions, we get, by considering the contribution of each arc in A:

$$\begin{aligned} p(U) + p(W) &= (f^-(U) - g^+(U)) + (f^-(W) - g^+(W)) \\ &= (f^-(U) + f^-(W)) - (g^+(U) + g^+(W)) \\ &= (f^-(U \cap W) + f^-(U \cup W) + f(U, W)) \\ &\quad -(g^+(U \cap W) + g^+(U \cup W) + g(U, W)) \\ &= (p(U \cap W) + p(U \cup W)) - (g(U, W) - f(U, W)), \end{aligned}$$

where $f(U, W)$ counts the f values on arcs with one end in $U - W$ and the other in $W - U$[19].

From this it follows that p is supermodular (since $f \leq g$) and that equality only holds if $f(a) = g(a)$ for all arcs with one end in $U - W$ and the other in $W - U$. This completes the proof of the claim.

An arc $a \in A$ is **tight** if $f(a) = g(a)$ and a subset $U \subset V$ is **tight** if $p(U) = b(U)$. Suppose that there is no feasible flow with respect to f, g and b in D and that f, g are chosen so that the number of tight arcs plus the number of tight sets is maximum.

If every arc $a \in A$ is tight, then take $x(a) = f(a) = g(a)$ for every $a \in A$. Now we have $x^-(U) - x^+(U) = f^-(U) - g^+(U) \leq b(U)$ and hence x is a feasible submodular flow in D, a contradiction.

Hence we may assume that there is some arc a_0 such that $f(a_0) < g(a_0)$. Suppose that there is no tight set which is entered by a_0. Then we can increase $f(a_0)$, until either the new value $f'(a_0)$ equals $g(a_0)$, or we find a tight set U (with respect to f', g) which is entered by a_0. It follows that the new functions f', g have a higher count of tight arcs plus tight sets. Hence, by the choice of f, g, there exists a feasible submodular flow x with respect to f', g. Obviously

[19] Again this definition generalizes the corresponding definition of $d(X, Y)$ in Chapter 5.

x is also feasible with respect to f, g, contradicting the assumption. Hence the arc a_0 must enter a tight set U.

Similarly we can prove (by lowering g otherwise) that the arc a_0 must also leave some tight set W. Now we have, using the claim, (11.30) and the fact that $p(U) = b(U), p(W) = b(W)$:

$$\begin{aligned} p(U \cap W) + p(U \cup W) &\geq p(U) + p(W) \\ &= b(U) + b(W) \\ &\geq b(U \cap W) + b(U \cup W) \\ &\geq p(U \cap W) + p(U \cup W), \end{aligned}$$

implying that equality holds everywhere above. However, this contradicts the second part of the claim since $f(a_0) < g(a_0)$ and we have argued that the arc a_0 leaves U and enters W. This contradiction completes the proof. □

Note that the special case of Theorem 11.8.3 when $b \equiv 0$ says that $x^-(U) - x^+(U) = 0$ for all subsets $U \subseteq V$. In particular $x^-(v) = x^+(v)$ for all $v \in V$. That is, every feasible submodular flow with respect to f, g and $b \equiv 0$ is circulation and conversely. It is easy to see that the characterization in Theorem 11.8.3 in the case $b \equiv 0$ is exactly the condition in Hoffman's circulation theorem (Theorem 4.8.2).

In fact, the proof of Theorem 11.8.3 in some sense resembles that of Theorem 4.8.2. Thus it is natural to ask how easy it is to find a feasible solution, or detect that none exists. This can be read out of the proof above: the essential step is to decide whether an arc enters or leaves a tight set (or both). This requires that we can find $\min\{b(U) - p(U) : a \in (U, \overline{U})\}$ and $\min\{b(U) - p(U) : a \in (\overline{U}, U)\}$ for every arc a of the directed multigraph D. This is a special case of the problem of minimizing a submodular function, that is, finding the minimum value of the submodular function in question over a prescribed family of sets. This can be done in polynomial time for arbitrary submodular functions using the ellipsoid method as shown by Grötschel, Lovász and Schrijver [429]. However, the ellipsoid method, though polynomial, is not of practical use, since it is highly inefficient.

It was an open problem for several decades whether there exists a polynomial combinatorial algorithm for minimizing a submodular function b over a family $\mathcal{F}$, that is, to find $\min\{b(U) : U \in \mathcal{F}\}$. Schrijver [802] solved the problem completely by describing a strongly polynomial time algorithm for minimizing an arbitrary submodular function given by a value-giving oracle. Schrijver's algorithm does not use the ellipsoid method or any other linear programming method. A similar result was obtained independently by Iwata, Fleischer and Fujishige [554][20].

As mentioned earlier, one can also define submodular flows for functions b that are intersecting, respectively crossing, submodular functions (defined

[20] In fact, both Schrijver and Iwata, Fleischer and Fujishige were awarded the 2003 Fulkerson price for their papers [554, 802].

on a family of subsets of the directed multigraph D which is intersecting, respectively crossing). In the case of intersecting and in particular for crossing submodular flows the feasibility theorem is much more complicated. A collection $U_1, U_2, \ldots, U_k$ of subsets of a ground set S are **co-disjoint** if their complements are pairwise disjoint (that is, $U_i \cup U_j = S$ for all $i \neq j$). Frank proved the following feasibility theorem crossing submodular flows:

Theorem 11.8.4 (Existence of a crossing submodular flow) *[339] Let $D = (V, A)$ be a directed multigraph and let f, g be real-valued modular functions such that $f \leq g$. Let $\mathcal{F}''$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}''$ and let b'' be a crossing submodular function on $\mathcal{F}''$. Then there exists a feasible submodular flow with respect to f, g and b'' if and only if*

$$f^-(\bigcup_{i=1}^{t} X_i) - g^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} \sum_{j=1}^{q_i} b''(X_{ij}) \tag{11.32}$$

holds for every subpartition $\{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint members $X_{i1}, X_{i2}, \ldots, X_{iq_i}$ of $\mathcal{F}''$. Furthermore, if f, g, b'' are all integer-valued functions and (11.32) holds, then there exists a feasible integer-valued submodular flow with respect to f, g and b''. □

Finding a feasible submodular flow or a configuration which shows that none exists is much more difficult than finding a feasible circulation in a network (recall Section 4.8). Frank [339] gave a combinatorial algorithm for finding a feasible integer-valued submodular flow with respect to bounds f, g and a pair $(\mathcal{F}, b)$ which is either intersecting or crossing submodular. The algorithm is polynomial provided one has an algorithm for minimizing the involved submodular functions. For this task we can apply the algorithms of Schrijver and Iwata, Fleischer and Fujishige which we mentioned above.

11.8.3 Minimum Cost Submodular Flows

Let $D = (V, A)$ be a directed multigraph and let $f : A \to \mathbb{Z} \cup \{-\infty\}$, $g : A \to \mathbb{Z} \cup \{\infty\}$ be functions on the arc set of D. Let $c : A \to \mathbb{R}$ be a cost function on the arcs of D. Let $\mathcal{B} \subseteq 2^V$ be a crossing family with $\emptyset, V \in \mathcal{B}$. Let $b : 2^V \to \mathbb{Z} \cup \{\infty\}$ be crossing submodular on $\mathcal{B}$ with $b(\emptyset) = b(V) = 0$. Denote the network defined by D and these functions by $\mathcal{N}_S = (V, A, f, g, (\mathcal{B}, b), c)$. The MINIMUM COST SUBMODULAR FLOW PROBLEM is as follows:

$$\begin{aligned} &\text{Minimize} && \sum_{a \in A} c(a)x(a) \\ &\text{subject to} \\ &&& x^-(U) - x^+(U) \leq b(U) && \text{for all } U \in \mathcal{B} \\ &&& f(a) \leq x(a) \leq g(a) && \text{for all } a \in A. \end{aligned}$$

A feasible submodular flow with respect to f, g and b which achieves this minimum is called an **optimal submodular flow** in $\mathcal{N}_S$.

This problem, which again generalizes the minimum cost circulation problem from Chapter 4, is very interesting because it forms a common extension of many problems on (di)graphs as well as problems from other areas (see, e.g., the book [364] by Fujishige). Recall also Theorem 11.8.1.

Fujishige proved the following (see also the papers [234] by Cunningham and Frank and Frank's paper [338]):

Theorem 11.8.5 *[365] The minimum cost submodular flow problem can be solved in polynomial time provided a polynomial algorithm for minimizing the relevant submodular functions is available.* □

11.8.4 Applications of Submodular Flows

In this section we will illustrate the usefulness of submodular flows as a tool to obtain short proofs of important results as well as algorithms for various connectivity problems. See also Section 13.1.

We start with Nash-Williams' orientation theorem (Theorem 11.5.3). The approach taken is due to Frank [340] (the same idea was used by Jackson [556]). Let $G = (V, E)$ be an undirected graph. Let D be an arbitrary orientation of G. Clearly G has a k-arc-strong orientation if and only if it is possible to reorient some arcs of D so as to get a k-arc-strong directed multigraph. Suppose we interpret the function $x : A \to \{0, 1\}$ as follows: $x(a) = 1$ means that we reorient a in D and $x(a) = 0$ means that we leave the orientation of a as it is in D. Then G has a k-arc-strong orientation if and only if we can choose x so that the following holds:

$$d_D^-(U) + x^+(U) - x^-(U) \geq k \qquad \forall\, \emptyset \neq U \subset V. \tag{11.33}$$

This is equivalent to

$$x^-(U) - x^+(U) \leq (d_D^-(U) - k) = b(U) \qquad \forall\, U \subset V, U \neq \emptyset, V, \tag{11.34}$$

$$b(\emptyset) = b(V) = 0. \tag{11.35}$$

Observe that the function b is crossing submodular on $\mathcal{F} = 2^V$ (it is not fully submodular in general, since we have taken $b(\emptyset) = b(V) = 0$). Thus we have shown that G has a k-arc-strong orientation if and only if there exists a feasible integer-valued submodular flow in D with respect to the functions $f \equiv 0, g \equiv 1$ and b.

Suppose now that G is $2k$-edge-connected, that is, $d_G(X) \geq 2k$ for all proper non-empty subsets of V. We claim that $x \equiv \frac{1}{2}$ is a feasible submodular flow. This follows from the following calculation:

$$\begin{aligned} d_D^-(U) + x^+(U) - x^-(U) &= d_D^-(U) + \frac{1}{2}d_D^+(U) - \frac{1}{2}d_D^-(U) \\ &= \frac{1}{2}d_D^-(U) + \frac{1}{2}d_D^+(U) \\ &\geq \frac{1}{2}(2k - d_D^+(U)) + \frac{1}{2}d_D^+(U) \\ &= k. \end{aligned}$$

Hence it follows from the integrality statement of Theorem 11.8.4 and the equivalence between (11.33) and (11.34) that there is a feasible integer-valued submodular flow x in D with respect to f, g and b. As described above this implies that G has a k-arc-strong orientation where the values of x prescribe which arcs to reverse in order to obtain such an orientation from D.

Notice that by formulating the problem as a minimum cost submodular flow problem, we can also solve the weighted version where the two possible orientations of an edge may have different costs and the goal is to find the cheapest k-arc-strong orientation of the graph (Exercise 11.66). By Theorem 11.8.5, the optimal (minimum cost feasible) submodular flow in D with respect to the functions $f \equiv 0, g \equiv 1$ and b (as defined in (11.34)) can be found in polynomial time (see Exercise 11.65).

The following useful result, mentioned by Frank in [338], follows from the discussion above and Theorem 11.8.5.

Corollary 11.8.6 *[338] There is a polynomial algorithm for finding the minimum number of arcs to reverse in a directed multigraph D in order to obtain a k-arc-strong reorientation of D.* □

Similarly, combining the discussion above with Frank's algorithm for finding a feasible submodular flow (or deciding that none exists) with respect to a crossing submodular function, we obtain the following result[21]:

Corollary 11.8.7 *[338] There is a polynomial algorithm for finding a k-arc-strong orientation of a given undirected multigraph G or verify that G has no such orientation.* □

The following theorem by Frank can also be derived from the formulation of the k-arc-strong orientation as a submodular flow problem (see Fujishige's book [364, Section 3.3]).

Theorem 11.8.8 *[337] If D and D' are k-arc-strong orientations of an undirected graph G, then there exists a sequence of k-arc-strong orientations $D = D_0, D_1, \ldots, D_r = D'$ of G such that for each $i \in [r]$, D_i is obtained from D_{i-1} by reversing all arcs in a directed path or a directed cycle.* □

[21] See Exercise 11.36 for a different proof based on Lovász's splitting theorem.

Frank [337] gives a direct and short proof of this without using submodular flows, but his proof uses submodular arguments (see Exercises 11.47-11.50).

In [358] Frank and Tardos showed how to reduce the following problem to a submodular flow problem. Given a directed graph $D = (V, A)$ and a special vertex s, find a minimum set of new arcs to add to D such that the resulting directed multigraph contains k internally disjoint paths from s to v for every $v \in V - s$. The similar problem where we only want arc-disjoint (s, v)-paths is solvable via matroid intersection algorithms (see Exercise 9.5).

For much more material on submodular flows the reader is referred to the books by Fujishige [364] and Schrijver [804], the papers [338, 339, 343, 347, 348] by Frank, [357] by Frank and Tardos and the paper [798] by Schrijver. In particular [357] and [364, 804] give a lot of interesting results on the structure of submodular flows and the relation between submodular flows and other models such as independent flows and polymatroidal flows. Finally Schrijver's paper [798] is a very useful overview of the various models and their interrelations.

11.9 Orientations of Mixed Multigraphs

In this section a mixed graph may contain multiple edges and/or arcs. Also recall that when we speak of orienting a mixed (multi)graph this means that we assign an orientation to every edge and leave the original arcs unchanged (implying that the result may not be an oriented graph).

Orientation problems for mixed graphs are generally much harder than for undirected graphs. One illustration of this is displayed in Figure 11.10. This example, due to Tardos (see [347]), shows that the linking principle for strong connectivity orientations does not hold for general mixed graphs (compare this with Theorem 11.7.5).

Not every $2k$-arc-strong mixed graph has a k-arc-strong orientation (Exercise 11.55) but Jackson proved the following extension of Theorem 11.5.3. The proof is left to the reader as Exercise 11.54.

Theorem 11.9.1 *[556] Let $M = (V, A, E)$ be a mixed graph. Let $G = (V, E)$ and $D = (V, A)$ denote the undirected, respectively the directed part of M, and define k by*

$$k = \min\{\lfloor \frac{1}{2} d_G(X) \rfloor + d_D^+(X) : X \text{ is a proper subset of } V\}.$$

Then the edges of E can be oriented in such a way that the resulting directed multigraph is k-arc-strong. □

It is not difficult to see that one can formulate the problem of orienting a mixed graph so as to get a k-arc-strong directed multigraph as a submodular

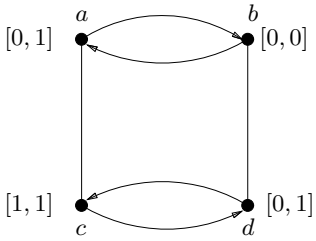


Figure 11.10 A mixed graph M with prescribed lower and upper bounds on the desired in-degrees in the directed multigraph induced by the arc between a and c and the arc between b and d in an orientation D of M. It is easy to see that by orienting the edges ac, bd as $a \to c, d \to b$ we obtain a strong orientation satisfying the lower bounds on the directed multigraph induced by the newly oriented arcs. Similarly if we orient the same edges as $c \to a, b \to d$ we obtain a strong orientation which satisfies the upper bounds on the directed multigraph induced by the newly oriented arcs. However, there is no strong orientation which satisfies the lower and upper bounds simultaneously on the directed multigraph induced by the newly oriented arcs.

flow problem. We can use the same approach as in Subsection 11.8.4. The only change is that we insist that $x(a) = 0$ for original arcs (Exercise 11.54).

Jackson [556] conjectured that Theorem 11.9.1 could be extended to local connectivities and hence providing a generalization of Nash-Williams' strong orientation theorem (Theorem 11.5.4). However, examples by Enni [295] show that this conjecture is false. In the case when the directed part $D = (V, A)$ of $M = (V, A, E)$ is eulerian such an extension is indeed possible. In [295] Enni shows how to extend Theorem 11.5.5 to the case of mixed graphs when the directed part $D = (V, A)$ is eulerian.

We remark that there seems to be no easy way of formulating orientation problems concerning local connectivities as submodular flow problems.

When we consider orientation problems where the input is a mixed graph $M = (V, A, E)$ which we wish to orient so as to satisfy a certain lower bound $h(X)$ on the in-degree of every subset X of vertices, then we cannot in general apply a theorem like Frank's orientation theorem (Theorem 11.7.6). The reason for this is that even if the function $h(X)$ 'behaves nicely', we have to take into account the arcs in A because these will contribute to the in-degree of the final oriented graph D'. To give an example, consider a mixed graph $M = (V, A, E)$ and let $h(X) = k$ for all proper subsets of V and $h(\emptyset) = h(V) = 0$. That is, we are looking for a k-arc-strong orientation of M. When we want to apply a theorem like Theorem 11.7.6 we have to consider the revised in-degree lower bound h' given by $h'(X) = k - d_D^-(X)$, where $D = (V, A)$ is the directed graph induced by the arcs already oriented in M. The function h' is easily seen to be crossing G-supermodular, where $G = (V, E)$ is the undirected part of M (Exercise 11.64). However,

h' is typically negative on certain sets and hence Theorem 11.7.6 cannot be applied.

As we mentioned above, for the particular lower bound $h(X) = k$, whenever $\emptyset \neq X \neq V$, the problem can be formulated as a submodular flow problem. This is no coincidence as we show below.

Let $G = (V, E)$ be an undirected graph. Let $h : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be crossing G-supermodular with $h(\emptyset) = h(V) = 0$. Let $D = (V, A)$ be an arbitrary but fixed orientation of G. Let $x : A(D) \to \{0, 1\}$ be a vector and define an orientation $D' = (V, A')$ of G by taking $A' = \{a : a \in A, x(a) = 0\} \cup \{\overleftarrow{a} : a \in A, x(a) = 1\}$. Here $\overleftarrow{a}$ denotes the opposite orientation of the arc a (compare with Section 11.8.4). Then D' will satisfy

$$d^-_{D'}(U) \geq h(U) \qquad \text{for all } U \subset V \tag{11.36}$$

if and only if $d^-_D(U) - x^-(U) + x^+(U) \geq h(U)$ for all $U \subset V$, or equivalently

$$x^-(U) - x^+(U) \leq d^-_D(U) - h(U) = b''(U) \qquad \text{for all } U \subset V. \tag{11.37}$$

Since d^-_D satisfies (5.2) and h is crossing G-supermodular[22], we conclude that whenever U, W are crossing sets the following holds:

$$\begin{aligned} b''(U) + b''(W) &= (d^-_D(U) - h(U)) + (d^-_D(W) - h(W)) \\ &= d^-_D(U \cap W) + d^-_D(U \cup W) + d_G(U, W) - (h(U) + h(W)) \\ &\geq d^-_D(U \cap W) + d^-_D(U \cup W) + d_G(U, W) - (h(U \cap W) \\ &\quad + h(U \cup W) + d_G(U, W)) \\ &= b''(U \cap W) + b''(U \cup W). \end{aligned} \tag{11.38}$$

Thus the function b'' is crossing submodular on $\mathcal{F}'' = 2^V - \{\emptyset, V\}$ and the equivalence of (11.36) and (11.37) shows that there is a one-to-one correspondence between orientations satisfying (11.36) and integer-valued solutions to the submodular flow problem defined by (11.37) and $0 \leq x \leq 1$. This shows that we can use submodular flow algorithms to solve the orientation problem. We can also derive a characterization of the existence of an orientation satisfying (11.36) from Theorem 11.8.4. We do this below as an illustration of how to use the feasibility theorem for crossing submodular flows (Theorem 11.8.4).

Suppose there exists an integer-valued feasible submodular flow with respect to the crossing submodular function b'' defined above. By (11.32) this means that

$$f^-(\bigcup_{i=1}^{t} X_i) - g^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} \sum_{j=1}^{q_i} b''(X_{ij}) \tag{11.39}$$

[22] Note how we use the definition of a crossing G-supermodular function here to get rid of the contribution from edges with one end in $X - Y$ and the other in $Y - X$.

holds for every subpartition $\mathcal{P} = \{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint subsets $X_{i1}, X_{i2}, \ldots, X_{iq_i}$ of V.

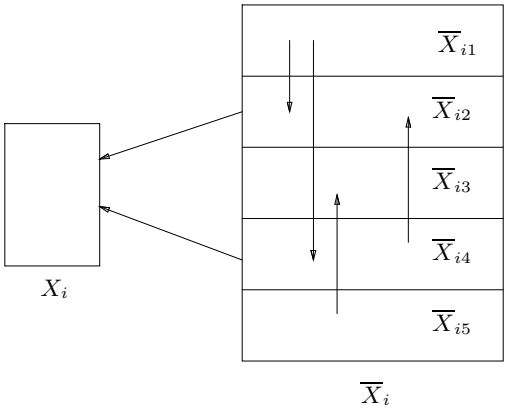


Figure 11.11 The situation when deriving Theorem 11.9.2 from Theorem 11.8.4. The set X_i is part of a subpartition $\mathcal{P}$ of V and X_i is the intersection of the five co-disjoint sets $X_{i1}, \ldots, X_{i5}$ whose complements (which form a partition of $\overline{X}_i$) are indicated in the figure. The arcs shown are those between different sets $\overline{X}_{ij}$, $\overline{X}_{ir}$ (which are the same as those arcs that go between different X_{ij}'s!) and those arcs that enter X_i.

We derive an expression that relates only to G and h using (11.39). To do so, it is helpful to study Figure 11.11.

Using that $f \equiv 0$ and $g \equiv 1$ and the definition of b'' we see that (11.39) is equivalent to

$$-d_D^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t}\sum_{j=1}^{q_i}(d_D^-(X_{ij}) - h(X_{ij})). \tag{11.40}$$

For fixed i the sum $\sum_{j=1}^{q_i} d_D^-(X_{ij})$ counts the following arcs:

(1) those arcs which enter X_i (the common intersection of all X_{ij}'s) from its complement, plus
(2) those arcs which go between different X_{ij}'s (which is the same as arcs that go from some $\overline{X}_{ij}$ to some other $\overline{X}_{ir}$). This is the same as the number of edges in G that go between two X_{ij}'s. Denote the total number of edges of this kind in G by e_i.

Using this observation we conclude that (11.40) is equivalent to

$$d_D^+(\cup_{i=1}^{t} X_i) + \sum_{i=1}^{t} d_D^-(X_i) \geq \sum_{i=1}^{t}(\sum_{j=1}^{q_i} h(X_{ij}) - e_i). \tag{11.41}$$

Finally, observe that the left-hand side of (11.41) counts precisely those edges of G which enter some $X_i \in \mathcal{P}$. Now we have proved the following orientation theorem due do Frank:

Theorem 11.9.2 (Frank's general orientation theorem) *[343] Let $G = (V, E)$ be an undirected graph. Let $h : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be crossing G-supermodular with $h(\emptyset) = h(V) = 0$. There exists an orientation D of G satisfying*

$$d_D^-(X) \geq h(X) \qquad \text{for all } X \subset V \tag{11.42}$$

if and only if

$$e_{\mathcal{P}} \geq \sum_{i=1}^{t} (\sum_{j=1}^{q_i} h(X_{ij}) - e_i) \tag{11.43}$$

holds for every subpartition $\mathcal{P} = \{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint sets $X_{i1}, X_{i2}, \ldots, X_{iq_i}$. Here $e_{\mathcal{P}}$ counts the number of edges which enter some member of $\mathcal{P}$ and e_i counts the number of edges which go between different sets X_{ij}, X_{ir}. □

By our previous remark on the function $k - d_D^-$, Theorem 11.9.2 can be used to derive a necessary and sufficient condition for the existence of a k-arc-strong orientation of a mixed graph. This is left to the reader as Exercise 11.58.

One might ask whether such a complicated condition involving partitions and copartitions is really necessary in Theorem 11.9.2. The following example due to Frank [347] shows that one cannot have a condition which only involves partitions or subpartitions.

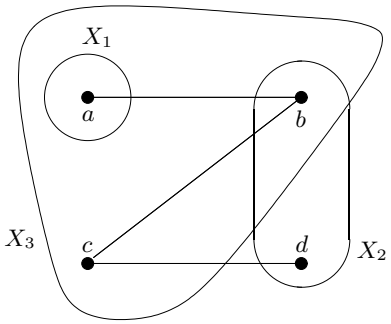


Figure 11.12 Frank's example showing that no (sub)partition type condition for the existence of an orientation satisfying (11.42) exists.

Let $G = (V, E)$ be the graph in Figure 11.12 and let the sets X_1, X_2, X_3 be as defined there. Define h by $h(\emptyset) = h(V) = 0$, $h(X_1) = h(X_3) = 1$, $h(X_2) = 2$ and $h(X) = -\infty$ for all other subsets of V. Then h is crossing

G-supermodular since no two crossing sets X, Y have $h(X), h(Y) > -\infty$. It is easy to check that G has no orientation satisfying (11.42) with respect to h. On the other hand, if we decrease $h(X_i)$ by one for any $i = 1, 2, 3$, then there exists a feasible orientation with respect to the new h_i. This shows that every certificate for the non-existence of an orientation with respect to h must include all the sets X_1, X_2, X_3. It is easy to see that these three sets neither form a subpartition nor do they form a co-partition.

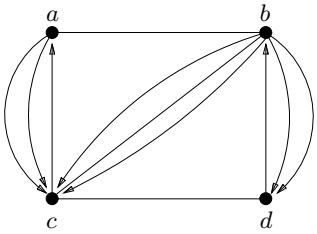


Figure 11.13 A mixed graph which has no 2-arc-strong orientation and for which every certificate for the non-existence of such an orientation must involve the three sets $\{a\}, \{b, d\}, \{a, b, c\}$ [347, Figure 2.3].

The example from Figure 11.12 also shows that there is no 2-arc-strong orientation of the mixed graph in Figure 11.13. Hence even for orientations of mixed graphs to obtain a uniform degree of arc-strong connectivity we cannot hope for a much simpler condition.

Since we derived Theorem 11.9.2 from Theorem 11.8.4, it is possible to get a simpler characterization if one can find such a characterization of feasibility of submodular flows with respect to a crossing pair $(\mathcal{F}'', b'')$. This was done by Frank in [347] where a somewhat simpler (but still far from easy) characterization was found.

11.10 *k*-(Arc)-Strong Orientations of Digraphs

We saw in Corollary 5.3.9 that every strong digraph without a bridge has a strong orientation. In this section we investigate how much of the degree of arc-strong or vertex-strong connectivity of a digraph D comes from its 2-cycles. More precisely, suppose we must delete one arc of every 2-cycle (thus obtaining an orientation of D), can we always maintain a high arc-strong, respectively vertex-strong, connectivity if the starting digraph has high arc-strong, respectively vertex-strong, connectivity? It is not difficult to see that we may not be able to preserve the same degree of arc/vertex-strong connectivity, even if D is semicomplete. See Figure 11.14 for an example. So the question is whether there exist functions $f(k), g(k)$ with the property that

every $f(k)$-strong (($g(k)$-arc-strong) digraph contains a spanning k-strong (k-arc-strong) subgraph without cycles of length 2.

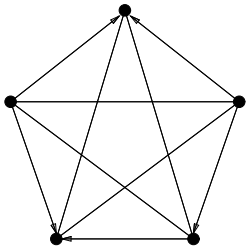


Figure 11.14 A 2-strong semicomplete digraph which has no 2-arc-strong spanning subtournament. Undirected edges correspond to directed 2-cycles.

Let us first consider arc-strong connectivity. Note that every k-arc-strong oriented graph D must have $UG(D)$ $2k$-edge-connected. In particular, if G is an undirected graph with edge-connectivity $\lambda(G) = 2k-1$ and $\overleftrightarrow{G}$ is the complete biorientation of G, then D does not contain a spanning k-arc-strong subgraph. Hence the following result due to Jackson and Thomassen implies that $g(k) = 2k$ and this is the best possible by the remark above.

Theorem 11.10.1 *[556, 866] Every $2k$-arc-strong digraph has a k-arc-strong orientation.* □

Since we may convert a digraph to a mixed graph by replacing each 2-cycle with an undirected edge, Theorem 11.10.1 follows from Theorem 11.9.1.

In the vertex-strong connectivity case the problem becomes much harder. Jackson and Thomassen posed the following conjecture (see [866]):

Conjecture 11.10.2 *[866]) Every $2k$-strong digraph has a k-strong orientation.*

In Section 11.6 we noted that if an oriented graph D is k-strong, then $UG(D)$ is k-connected and $2k$-edge-connected. The following example, due to Alon and Ziegler [866, page 406], shows that $UG(D)$ may be k-connected and $2k$-edge-connected and still D has no k-strong orientation: take the complete biorientation of H, where H is the graph constructed by taking two large complete graphs G_1, G_2 sharing just one vertex v and adding $k-1$ independent edges with one end in $V(G_1) - v$ and the other in $V(G_2) - v$.

On the other hand, the example above does not satisfy the obvious necessary condition that $D - X$ has a $(k-j)$-arc-strong orientation for every set X of j vertices ($0 \leq j \leq k$). The semicomplete digraph in Figure 11.14 shows that this condition is still not sufficient and the example can be generalized

to an arbitrary odd number of vertices by taking the second power on an odd cycle C and orienting the original edges as in Figure 11.14. This shows that Conjecture 11.6.2 can be extended neither to mixed graphs nor to digraphs.

Conjecture 11.10.2 clearly generalizes Conjecture 11.6.1 and it is open even for $k = 2$ whether there there is some function $f(k)$ so that every $f(k)$-strong digraph has a k-strong orientation. Jordán's proof that every 18-connected graph has a 2-strong orientation does not extend to orientations of digraphs as it explicitly uses that the starting point is an undirected graph. Below we will describe some results on special classes of digraphs.

Guo [435] and Huang [540] considered orientations of locally semicomplete digraphs.

Theorem 11.10.3 *[540] Every round decomposable k-strong locally semicomplete digraph can be oriented as a k-strong local tournament*[23]*.* □

Bang-Jensen and Thomassen [66] proved that for semicomplete digraphs the function $f(k)$ indeed exists. The value of this function was later improved by Guo.

Theorem 11.10.4 *[435] For every natural number k, every $(3k-2)$-strong locally semicomplete digraph has an orientation as a k-strong local tournament digraph.*

We will not prove the bound $3k-2$ here, but instead give the proof by Bang-Jensen and Thomassen that $f(k) \leq 5k$ for semicomplete digraphs. That proof illustrates the main ideas and Guo's proof is a refinement of the proof we give. Note that by Theorem 11.10.3 it is enough to consider semicomplete digraphs.

We prove by induction on k that every $5k$-strong semicomplete digraph D contains a spanning k-strong tournament. The case $k = 1$ is easy, since by Theorem 1.5.1, every strong semicomplete digraph has a Hamilton cycle. Let C be a Hamilton cycle in D. For every 2-cycle of D delete an arbitrary arc of that 2-cycle, unless one of its arcs is used by C. In the latter case we delete one arc of the 2-cycle so as to preserve C. We obtain a spanning strong tournament T of D. Note that the case $k = 1$ also follows easily from Corollary 5.3.9.

Suppose we have proved the statement for all $r \leq k-1$, that is, every $5r$-strong semicomplete digraph contains a spanning r-strong tournament. Let D be a $5k$-strong semicomplete digraph and suppose D does not contain a spanning k-strong tournament. We derive a contradiction to this assumption. First observe that we must have $|V(D)| \geq 5k+2$ since otherwise D is the complete digraph on $5k+1$ vertices and this clearly contains a k-connected spanning tournament.

By the induction hypothesis, D contains a $(k-1)$-strong spanning tournament. Let T be chosen among all $(k-1)$-strong spanning tournaments of D such that the following holds:

[23] The paper [540] claims that in fact all k-strong non-semicomplete locally semicomplete digraphs have such an orientation. This, however, is not true.

(i) The number s of separating sets of size $k-1$ in T is minimum over all $k-1$-strong spanning subtournaments of D.
(ii) T has a separating set S of size $k-1$ such that the number m of strong components of $T-S$ is minimum taken over all separating sets of size $k-1$ of T.

Let S be some separating set of T such that $T-S$ has precisely m strong components $T_1, \ldots, T_m$ (written in the unique acyclic order). Let $U = V(T_1) \cup \ldots \cup V(T_{m-1})$ and $W = V(T_m)$. Since D is $5k$-strong, it follows easily from Menger's theorem (Corollary 5.4.2) that in D there are $5k$ internally disjoint paths from W to U (see Exercise 5.17). At most $k-1$ of these can pass through S. Thus in $D-S$ there are at least $4k+1$ arcs from W to U. Let $U' \subset U$ ($W' \subset W$) be those vertices v of U (W) for which some arc in D from W to U has v as its head (tail). Since $D-S$ has at least $4k+3$ vertices, either U or W has size at least $2k+2$. Using this and the fact that $D-S$ has $4k+1$-internally disjoint (w,u)-paths for every choice of $u \in U, w \in W$, we get from Corollary 5.4.2 that either $|U'| \geq 2k+1$ or $|W'| \geq 2k+1$. By considering the converse of D if necessary, we may assume $|U'| \geq 2k+1$.

The digraph $T\langle U' \rangle$ is a tournament on at least $2k+1$ vertices and hence it has a vertex x with at least k out-neighbours in U'. Let y be a vertex in W' such that yx is an arc of D (y exists since $x \in U'$). In T we have the arc xy (since every vertex in U dominates every vertex in W) and since x has out-degree at least k in $T\langle U' \rangle$, there are at least k (x,y)-paths of length 2 in T. Let T' be the spanning tournament in D that we obtain from T by replacing the arc xy by the arc yx. Applying Lemma 14.4.6, we get that T' has no more than s minimum separating sets. However, it is easy to see that $T'-S$ is either strong (if $x \in V(T_1)$), or it has fewer strong components than $T-S$ and hence we obtain a contradiction to the choice of T according to (i), (ii). □

It can be seen by inspecting Guo's proof in [435] that $(3k-2)$-strong connectivity is the best bound one can prove using his approach. However, at least for $k=2$ this is not sharp when we have more than $2k$ vertices:

Proposition 11.10.5 *[108] Every 3-strong semicomplete digraph on at least 5 vertices contains a spanning 2-strong tournament.* □

As an illustration of the usefulness of the structural characterization of quasi-transitive digraphs in Theorem 2.7.5 we show how Theorem 11.10.4 implies the same statement for quasi-transitive digraphs.

Corollary 11.10.6 *For every natural number k, every $(3k-2)$-strong quasi-transitive digraph has an orientation as a k-strong quasi-transitive digraph.*

Proof: Let D be a $(3k-2)$-strong quasi-transitive digraph and let $D = Q[W_1, \ldots, W_q]$, $q = |Q|$, be a decomposition of D according to Theorem 2.7.5. By Corollary 5.8.2, the digraph D_0 obtained from D by deleting all

arcs inside each W_i is also $(3k-2)$-strong. By Theorem 2.7.5, if Q contains a 2-cycle $q_iq_jq_i$, then each of W_i, W_j has size one. Now let H be a semicomplete digraph obtained from D_0 by adding an arbitrary arc between every pair of vertices inside each $V(W_i)$. Clearly H is (at least) $(3k-2)$-strong and hence, by Theorem 11.10.4, it contains a spanning k-connected tournament T (which is obtained from H by deleting one arc from every 2-cycle, that is, T is an orientation of H). By the way we constructed H, we have $T = Q'[T_1, \ldots, T_q]$ for some choice of tournaments $T_1, \ldots, T_{|Q|}$ on $|W_1|, \ldots, |W_q|$ vertices, respectively. Here Q' is a spanning tournament in Q. Applying Corollary 5.8.2 to $T = Q'[T_1, \ldots, T_q]$, we get that the quasi-transitive digraph $D' = Q'[\overline{K}_{|W_1|}, \ldots, \overline{K}_{|W_q|}]$ is k-strong and by the remark above on 2-cycles in Q we see that D' is a spanning subdigraph of D. It is easy to see that if we delete an arc from every 2-cycle of a quasi-transitive digraph, then the result is a quasi-transitive digraph (Exercise 11.68). Let W_i' be obtained from W_i by deleting one arc from every 2-cycle in W_i for $i = 1, 2, \ldots, q$. Now we see that $D'' = Q'[W_1', W_2', \ldots, W_q']$ is the desired k-strong orientation of D. □

Note that it also follows from the proof above that every $(3k-2)$-strong quasi-transitive digraph contains a spanning k-strong extended tournament.

11.11 Miscellaneous Topics

11.11.1 Another Measure of Well-Balancedness

Instead of trying to find well-balanced orientations where $\lambda_D(x,y)$ and $\lambda_D(y,x)$ are as close as possible for all pairs of vertices, one may also look for different measures for the quality of an orientation. Pekéc (private communication, October 1997) posed the following problem:

Problem 11.11.1 *Let G be a multigraph and define M_{opt} as*

$$M_{opt} = \max\{\sum_{x,y \in V(D)} \lambda_D(x,y) : D \text{ is an orientation of } G\}.$$

Is there a nice characterization for M_{opt}? In particular, can M_{opt} be calculated in polynomial time?

11.11.2 Orienting to Preserve Reachability for Prescribed Pairs

In [492] Hakimi, Schmeichel and Young considered the problem of orienting a connected graph in such a way as to maximize or minimize the number of pairs of vertices x, y for which the orientation has an (x,y)-path. Let G be an undirected graph, D be an orientation of G and denote by $reach(D)$

the number of pairs (x, y) so that D has an (x, y)-path. Denote by $reach(G)$ ($Reach(G)$) the minimum (maximum) of $reach(D)$ over all orientations of G. By Robbin's theorem $Reach(G) = |V(G)|(|V(G)| - 1)$ if and only if it is 2-edge-connected. It is shown in [492] that $Reach(G)$ can be found in polynomial time.

Problem 11.11.2 *[492] Determine the complexity of the maximum reachability orientation problem for mixed graphs.*

Let us call an orientation D with $reach(D) = reach(G)$ **minimal**. It is easy to see that every minimal orientation is acyclic (Exercise 11.70). Using this observation it is shown in [492] that $reach(G)$ is equal to $|E(G)|$ plus the number of edges one needs to add to G in order to obtain a comparability graph. They then show that determining this number is NP-hard and hence so is determining $reach(G)$.

In [46] Arkin and Hassin considered orientations of mixed graphs which preserve directed paths between prescribed ordered pairs of vertices. They show that the following is an NP-hard problem: Given mixed graph M and a collection $P = \{(s_j, t_j) : j \in [r]\}$ of ordered pairs of vertices in M. Does M have an orientation D which contains a directed (s_j, t_j)-path for each $j \in [r]$ (such an orientation is called a P**-orientation** of M and M is called P**-connected** if it has an (s_i, t_i)-path for all $i \in [r]$)?

When M is some undirected graph it is easy to show that the desired orientation exists if and only if there is no edge e whose removal disconnects the graph into components X and Y so that for some $i \neq j$ $s_i, t_j \in X$ and s_j, t_i in Y (Exercise 11.71). Such an edge is called a ***P*-bridge**. In the case of mixed graphs we must also take into account the directed edges and it is easy to see that the absence of P-bridges is not sufficient to guarantee the existence of a P-orientation even when $|P| = 2$. An edge e of a mixed graph M is called P**-essential** if none of the two orientations of e preserve P-connectedness of the resulting mixed graph. When $|P| = 2$ the existence of P-essential edges is the only thing that can prevent a P-connected mixed graph from having a P-orientation. For $|P| > 2$ the absence of P-essential edges is no longer sufficient [46].

Theorem 11.11.3 *[46] A mixed graph M has a P-orientation for $P = \{(s_1, t_1), (s_2, t_2)\}$ if and only if M is P-connected and has no essential edges.* □

By Theorem 11.5.4, an undirected graph G with $s, t \in V(G)$ has an orientation D so that $\min\{\lambda_D(s, t), \lambda_D(t, s)\} \geq k$ if and only if $\lambda_G(s, t) \geq 2k$. Motivated by this the following problem was posed in [46].

Problem 11.11.4 *Given an undirected multigraph $G = (V, E)$, vertices $s_1, s_2, t_1, t_2 \in V$ and a positive integer k. Does G have an orientation D such that $\lambda_D(s_i, t_i) \geq k$ for $i = 1, 2$?*

It is shown by a simple example in [46] that already for $k = 2$ it is not sufficient to require that every edge-cut separating V into X, Y so that $s_1, t_2 \in X$ and $s_2, t_1 \in Y$ has at least $2k$ edges.

11.12 Exercises

11.1. Prove Proposition 11.1.16.

11.2. Show that if a locally semicomplete digraph D contains a 2-cycle xyx, then the edge xy is balanced in $UG(D)$.

11.3. **(+) Lexicographic 2-colouring gives a transitive orientation of comparability graphs**. Prove Theorem 11.1.4.

11.4. Prove that if G is a reduced proper circular arc graph, then, up to reversing the orientation of all arcs, G has a unique orientation as a round local tournament.

11.5. **(+) Linear algorithm for recognizing round local tournaments**. Prove that there is an $O(n + m)$ algorithm which either finds a round labelling of an oriented graph D or decides that D is not a round local tournament (Huang [539]).

11.6. Prove Theorem 11.1.8.

11.7. Using the same approach as in the proof of Proposition 11.1.15 formulate the instance of 2-SAT which corresponds to the oriented graph D in Figure 11.15. Show that $UG(D)$ has no orientation as a locally in-tournament digraph.

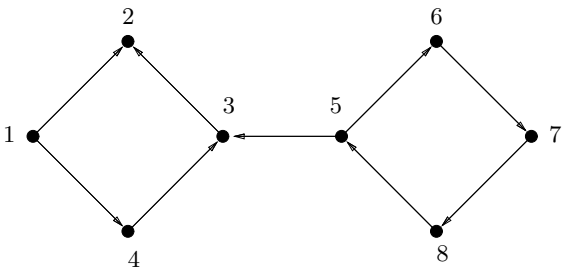


Figure 11.15 An oriented graph D.

11.8. **An orientation-based characterization of proper interval graphs**. A **straight ordering** of an oriented graph is a vertex ordering $v_1, v_2, \ldots, v_n$ such that for each i the vertex v_i is dominated by $v_{i-d^-(v_i)}, v_{i-d^-(v_i)+1}, \ldots, v_{i-1}$ and dominates $v_{i+1}, v_{i+2}, \ldots, v_{i+d^+(v_i)}$. Here indices are **not** modulo n, that is, $1 \leq i - d^-(v_i)$ and $i + d^+(v_i) \leq n$ for each $i \in [n]$. A digraph is **straight** if it has a straight ordering (Deng, Hell and Huang [257]). A graph is a **proper interval graph** if it is the intersection graph of an inclusion-free family of intervals on the real line.

(a) Prove that if D has a straight ordering, then D is an acyclic round local tournament digraph.
(b) Prove that an undirected graph G is a proper interval graph if and only if it has a straight orientation. Hint: compare this with Theorem 11.1.6.

11.9. **Recognizing non-strong locally semicomplete digraphs in linear time**. Give a simple linear algorithm to recognize non-strong locally semicomplete digraphs based on Theorem 2.10.6 (Bang-Jensen, Hell and Huang [102]).

11.10. Derive Theorem 11.3.1 from Theorem 11.3.7.

11.11. **(+) Acyclic orientations such that every vertex is on an (s,t)-path**. Let $G = (V, E)$ be an undirected graph. Let s, t be special vertices and assume that if G has a cut-vertex, then every cut-vertex v of G separates $G - v$ into two connected components, one containing s and one containing t. Prove that G has an acyclic orientation D such that every vertex of D is on an (s,t)-path (Gerards and Shepherd [400]).

11.12. **Strong orientations of the Petersen graph contain an even cycle**. Prove that every strongly connected orientation of the Petersen graph has an even cycle.

11.13. **Strong orientations of odd-K_4's and odd necklaces contain even cycles**. Prove Lemma 11.2.2.

11.14. **Undirected graphs without even cycles.** Describe the structure of those connected undirected graphs that have no even cycle.

11.15. **Graphs with strong orientations without even cycles and with the maximum number of vertices**. Prove that the graph L_n defined in Section 11.2 has a strong orientation without even cycles.

11.16. **(−)** Prove that Theorem 11.3.7 implies Camion's theorem (Corollary 1.5.2) that every strong tournament has a hamiltonian cycle.

11.17. **3-colouring the Petersen graph**. Find an orientation of the Petersen graph which has no directed path of length 3. Use this to find a 3-colouring of the Petersen graph by colouring as in the proof of Theorem 11.3.1.

11.18. Figure 11.16 shows a graph G known as the Grötzsch graph. Prove that every orientation of G has a path of length 3. Find an orientation D of G such that $\mathrm{lp}(D) = 3$. Finally, show that if e is any edge of G, then we can find an orientation of $G - e$ with no path of length 3.

11.19. Prove that if a connected graph G has a k-flow (D, x) for some k, then D is strongly connected.

11.20. Prove that a cubic graph is 3-edge-connected if and only if it is 3-connected.

11.21. **(+)** Prove that the Petersen graph has no 4-flow.

11.22. **Hamiltonian graphs have a 4-flow**. Prove that every hamiltonian graph has a 4-flow. Hint: use Theorem 11.4.6.

11.23. Find a 4-flow in the cubic graph in Figure 11.17.

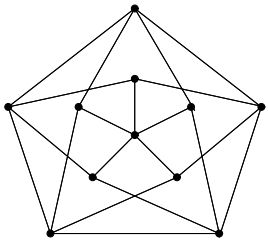


Figure 11.16 The Grötzsch graph.

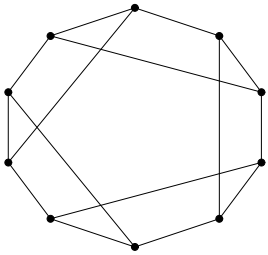


Figure 11.17 A hamiltonian cubic graph.

11.24. **Converting a $\mathbb{Z}_k$-flow to a k-flow.** The proof of Theorem 11.4.3 gives rise to a polynomial algorithm to convert a given $\mathbb{Z}_k$-flow to a k-flow. Describe such an algorithm and illustrate it by converting the $\mathbb{Z}_5$-flow in the Petersen graph in Figure 11.18 to a 5-flow.

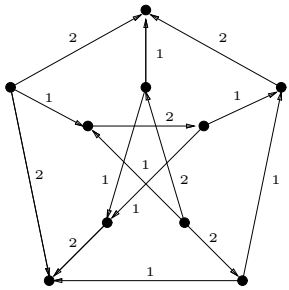


Figure 11.18 A $\mathbb{Z}_5$-flow in the Petersen graph.

11.25. (+) Prove Theorem 11.4.6. Hint: define a $\mathbb{Z}_2 \times \mathbb{Z}_2$-flow from G_1, G_2 and vice versa.

11.26. Show that the complete graph on 4 vertices is 3-edge-colourable and has no 3-flow.

11.27. **(+) Three spanning trees with no common edges in graphs which are 3-edge-connected**. Prove that every 3-edge-connected graph has three spanning trees T_1, T_2, T_3 with the property that $E(T_1) \cap E(T_2) \cap E(T_3) = \emptyset$. Hint: use Theorem 9.4.2.

11.28. **(+) Jaeger's 8-flow theorem**. Prove, without using Theorem 11.4.10, that every 2-edge-connected graph G has an 8-flow. Hint: first observe that it suffices to prove the statement for 3-edge-connected graphs. By Exercise 11.27, G has three spanning trees such that no edge lies in all of these. Use this to construct a $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$-flow in G (compare this with the proof of Theorem 11.4.7).

11.29. **A minimum counterexample to Tutte's 5-flow conjecture has no 3-cycle**. Show that if G is cubic 3-edge-connected and C is a 3-cycle of G, then the graph H obtained by contracting C to one vertex v in G and deleting the loops created is also cubic and 3-edge-connected. Show that every 5-flow in H can be extended to a 5-flow in G.

11.30. Show by an example that the idea of Exercise 11.29 does not always work for cycles longer than 5.

11.31. **(−) Nash-Williams' strong orientation theorem for eulerian multigraphs**. Prove Theorem 11.5.4 for eulerian graphs. Hint: consider an eulerian tour.

11.32. **Smooth k-arc-connected orientations**. Prove the following slight extension of Nash-Williams' orientation theorem. If $G = (V, E)$ is $2k$-connected, then it has a k-arc-strong orientation D such that $\max\{|d_D^+(x) - d_D^-(x)| : x \in V(D)\} \leq 1$. Hint: follow the proof of Theorem 11.5.3 and change it appropriately when needed.

11.33. **(+) Vertices of degree k in minimally k-edge-connected graphs**. Prove that every minimally k-edge-connected graph contains a vertex of degree k. Hint: use the results analogous to Proposition 5.1.1 for undirected graphs.

11.34. **(+) Lovász's splitting theorem for undirected edge-connectivity**. Prove Theorem 11.5.1. Hint: define a set of vertices X not containing the special vertex s to be **k-dangerous** if $d(X) \leq k + 1$. Clearly a splitting (su, sv) preserves k-edge-connectivity unless there is some k-dangerous set $X \subset V$ with $u, v \in X$. Observe that the degree function of an undirected graph has properties analogous to Proposition 5.1.1. Use this to show that there are at most two distinct maximal k-dangerous sets X, Y which contain a given neighbour t of s. Let X, Y be distinct maximal k-dangerous sets containing t but not s if such sets exist. Otherwise, either let X be the unique maximal k-dangerous set containing t but not s and $Y = \emptyset$ or, if no k-dangerous sets containing t but not s exist, then take $X = Y = \emptyset$. Conclude that s has a neighbour t' in $V - (X \cup Y)$ and show that (st, st') is an admissible splitting [655].

11.35. **(+) Splittings that do not create cut-edges.** Prove the following result due to Fleischner [319]. If G is a 2-edge-connected undirected graph and s is a vertex of degree at least 4, then there exist neighbours u, v of s such that replacing the edges su, sv by one edge uv results in a graph which is 2-edge-connected. Hint: this follows from Theorem 11.5.1 if $d_G(s)$ is even. If

$d_G(s)$ is odd, then study maximal 2-dangerous sets containing neighbours of s (see also the hints for Exercise 11.34).

11.36. **(+) A polynomial algorithm for finding a k-arc-strong orientation of a $2k$-edge-connected multigraph**. Convert the proof of Theorem 11.5.3 into a polynomial algorithm which finds a k-arc-strong orientation of an arbitrary input multigraph, or outputs a proof that none exists.

11.37. Prove Proposition 11.5.6.

11.38. Prove Proposition 11.5.12.

11.39. Prove that an orientation D of a multigraph G is well-balanced if and only if

$$d_D^-(X) - d_D^+(X) \leq b_G(X) \qquad \forall X \subseteq V. \tag{11.44}$$

11.40. Prove Corollary 11.7.4.

11.41. (+) Show how to derive Theorem 11.7.1 from Theorem 4.8.4.

11.42. Show how to convert the proof of Theorem 11.7.3 into a polynomial algorithm which either finds an orientation with the desired property, or a set violating the corresponding necessary condition.

11.43. Show how to derive Theorem 11.7.3 using the approach taken in the beginning of Subsection 11.7.1 and Exercise 4.32.

11.44. Prove that Theorem 11.7.3 implies the marriage theorem (Corollary 4.11.4).

11.45. Prove that the condition in Conjecture 11.6.2 is necessary in order to have a k-strong orientation.

11.46. **Reversing the orientation of a cycle preserves arc-strong connectivity**. Prove that if D is k-arc-strong and C is a cycle in D, then the digraph obtained by reversing the orientation of all arcs on C is also k-arc-strong.

11.47. **(+) Converting one k-arc-strong orientation into another via reversal of cycles**. Suppose that D and D' are k-arc-strong orientations of $G = (V, E)$ and that $d_D^-(v) = d_{D'}^-(v)$ for every $v \in V$. Prove that one can obtain D' from D by successive reversals of the orientation of a cycle in the current digraph.

11.48. **Reversal of a path while preserving k-arc-strong connectivity**. Suppose that D and D' are k-arc-strong orientations of a graph G and that there exists a vertex u such that $d_D^-(u) < d_{D'}^-(u)$. Show that D' contains a vertex v such that $d_D^-(v) > d_{D'}^-(v)$ and a (u, v)-path P. Under what conditions can we obtain a new k-arc-strong orientation of G by reversing the arcs of P?

11.49. **(+) Finding a good path to reverse.** Suppose that D and D' are k-arc-strong orientations of a graph G and that there exists a vertex u such that $d_D^-(u) < d_{D'}^-(u)$. Prove that there is always a vertex v such that $d_D^-(v) > d_{D'}^-(v)$ and a (u, v)-path P such that one can reverse all arcs of P without destroying the k-arc-strong connectivity. Hint: use your observation in Exercise 11.48. Assume that all paths are bad. Use submodularity of d_D^- to show that the maximal sets $X_1, X_2, \ldots, X_h$ containing v but not u and which have in-degree k in D are pairwise disjoint. Count those arcs

that have at most one end-vertex in $\cup_{i=1}^{h} X_i$ in both D and D' and obtain a contradiction (Frank [337]).

11.50. **Proof of Theorem 11.8.8**. Combine your observations in Exercises 11.47, 11.48 and 11.49 into a proof of Theorem 11.8.8.

11.51. Show that Theorem 11.5.3 is a special case of Theorem 11.7.6.

11.52. Let $G = (V, E)$ be an arbitrary connected graph containing two edge-disjoint connected spanning graphs $H = (V, E')$ and $T = (V, E'')$. Suppose H is not eulerian and that $\mathcal{O} = \{v_1, v_2, \ldots, v_{2k}\}$ is the set of odd degree vertices in H. Show that one can use a subset of the edges from E'' to form a collection of k edge-disjoint paths whose end-vertices are precisely the set $\mathcal{O}$.

11.53. Let $D = (V, A)$ be a digraph and $x : A \to \mathbb{R}$ a function on the arc set of D. Show that the function $x^-(U) - x^+(U)$ is a modular function.

11.54. (+) Prove Theorem 11.9.1. Hint: use a similar approach to that used in Section 11.8.4 to prove Theorem 11.5.3 via submodular flows.

11.55. Construct $2k$-arc-strong mixed graphs with no k-arc-strong orientation. Hint: they must violate the condition in Theorem 11.9.1.

11.56. Prove directly that the condition (11.43) is necessary for the existence of an orientation satisfying (11.42). Hint: assume that D is an orientation which satisfies (11.42) and study which edges are counted by the sum $\sum_{j=1}^{q_i} d_D^-(X_{ij})$.

11.57. **Constructing k-(strongly)-connected k-regular (di)graphs**. Prove that the rth power of an undirected cycle is $(2r)$-connected. Prove that if n is even and G is obtained from an even cycle $v_1v_2 \ldots v_{2k}v_1$ by taking the rth power and then adding longest diagonals (v_1v_{k+1}, v_2v_{k+2}, etc.), then G is $(2r+1)$-connected. These graphs are due to Harary [498], see also the book [872, pages 202-205] by Thulasiraman and Swamy.

11.58. (+) **Orienting a mixed graph to be k-arc-strong**. Use Theorem 11.9.2 to derive a necessary and sufficient condition for a mixed graph $M = (V, A, E)$ to have a k-arc-strong orientation (Frank [343, 347]).

11.59. (+) **Orientations containing k-arc-disjoint out-branchings from a given root.** Let $G = (V, E)$ be an undirected graph with a special vertex $s \in V$ and let k be a natural number. Prove without using Theorem 11.7.6 that G has an orientation such that $d^-(X) \geq k$ for every $X \subseteq V - s$ if and only if (9.5) holds (Frank [344]).

11.60. (+) **Orienting a mixed graph in order to obtain many arc-disjoint branchings.** Consider the problem of finding an orientation of a mixed graph $M = (V, A, E)$ so that it has k arc-disjoint out-branchings rooted at a specified vertex s or concluding that no such orientation exists. Show how to reduce this problem to a submodular flow problem. Argue that you can also solve the minimum cost version where there may be different costs on the two possible orientations of an edge $e \in E$.

11.61. (+) **Arc-disjoint in- and out-branchings with a fixed root in orientations of graphs.** Describe an algorithm to decide whether a given undirected graph $G = (V, E)$ has an orientation D such that there exist arc-disjoint in- and out-branchings F_v^+, F_u^- where $u, v \in V$ are specified (not

necessarily distinct) vertices of V. Prove that the corresponding problem for mixed graphs is $\mathcal{NP}$-complete. Hint: use Theorem 9.6.1.

11.62. (−) Characterize when an undirected graph $G = (V, E)$ has an orientation so that x, y are in the same strong component for specified distinct vertices $x, y \in V$.

11.63. **Orienting a mixed graph so as to get a closed trail containing two specified vertices.** Show that the following problem is $\mathcal{NP}$-complete: Given a mixed graph $M = (V, A, E)$ and two distinct vertices s, t. Decide if M has an orientation that contains are-disjoint (s, t)-, (t, s)-paths.

11.64. Let $M = (V, A, E)$ be a mixed graph and let $D = (V, A)$ be the directed part of M. Prove that for every k the function $k - d_D^-$ is crossing G-supermodular. Hint: use the fact that d_D^- is submodular.

11.65. Show how to minimize the submodular function b defined by (11.34) and (11.35) over a given collection of subsets in polynomial time. Hint: use flows to determine the in-degrees of the relevant sets.

11.66. Let k be a natural number and let $G = (V, E)$ be a graph with a cost function c that for every edge $e \in E$ assigns a cost to each of the two possible orientations of e. Show how to formulate the problem of finding a k-arc-strong orientation of G of minimum cost with respect to c as a minimum cost submodular flow problem.

11.67. **Reversing arcs in order to get many arc-disjoint out-branchings from a fixed root.** Show how to solve the following problem using submodular flows. Given a directed multigraph $D = (V, A)$, a vertex $s \in V$ and a natural number k. Determine whether it is possible to reverse the orientation of some arcs in A such that the resulting directed multigraph has k arc-disjoint out-branchings rooted at s. Argue that one can also solve the minimum cost version of the problem in polynomial time.

11.68. **Orientations of quasi-transitive digraphs are quasi-transitive**. Prove this claim. Hint: use Theorem 2.7.5.

11.69. Derive Theorem 11.7.6 from the feasibility theorem for crossing submodular flows (Theorem 11.8.4).

11.70. Consider the reachability orientation problem in Section 11.11.2. Prove that every minimal orientation is acyclic.

11.71. Consider the P-orientation problem (Subsection 11.11.2) for an undirected graph G and show that the desired orientation exists if and only if G has no P-bridge.

12. Sparse Subdigraphs with Prescribed Connectivity

In this chapter we treat a topic that is of practical interest and at the same time illustrates important applications of several concepts from other chapters, including Chapters 2, 4, 5 and 6.

A spanning k-(arc)-strong subdigraph D' of a directed multigraph D is called a **certificate** for the k-(arc)-strong connectivity of D. One of the central questions treated in the chapter is the following optimization problem. Let $D = (V, A)$ be a k-(arc)-strong directed multigraph and let c be a cost function on A (possibly $c(a) = 1$ for all $a \in A$). What is the minimum cost of a k-(arc)-strong spanning subdigraph D' of D? An **optimal certificate** for k-(arc)-strong connectivity in D is a spanning k-(arc)-strong subdigraph D' of minimum cost. Finding such an optimal certificate is a hard problem already when $k = 1$ and $c \equiv 1$. This follows from the fact that the optimal certificate for the strong connectivity of D has n arcs if and only if D has a Hamilton cycle. The fact that the Hamilton cycle problem is a special case of the problem of finding an optimal certificate for strong connectivity makes it interesting to consider classes of digraphs for which we know that the Hamilton cycle problem is polynomially solvable and to see what we can say about the complexity of finding the optimal certificate for strong connectivity (when $c \equiv 1$). This topic is treated in Section 12.2 for some classes of generalizations of tournaments. Section 12.3 deals with approximating the size of an optimal certificate for strong connectivity and in Section 12.5 we briefly deal with the case when c is arbitrary and we are looking for a minimum cost certificate for strong connectivity. In Section 12.4 we give a number of results on small certificates for k-(arc)-strong connectivity. In Section 12.6 we consider digraph analogues of the graph Steiner problem and finally in Section 12.7 we discuss a number of miscellaneous topics.

Definition 12.0.1 *Let $D = (V, A)$ be a directed multigraph. We denote by $\delta_{\geq k}(D)$ the minimum number of arcs in a spanning subdigraph D' of D which has $\delta^0(D') \geq k$. If $\delta^0(D) < k$, we let $\delta_{\geq k}(D) = \infty$.*

Proposition 12.0.2 *There exists a polynomial algorithm $\mathcal{A}$ which, given a directed multigraph $D = (V, A)$ with minimum semi-degree $\delta^0(D) \geq k$, returns a subset $A' \subseteq A$ of cardinality $\delta_{\geq k}(D)$ such that the directed multigraph $D' = (V, A')$ has $\delta^0(D') \geq k$.*

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_12,

Proof: Exercise 12.9. □

The following result is an easy consequence of the proof of Theorem 5.6.1.

Proposition 12.0.3 *Every k-arc-strong digraph contains a spanning k-arc-strong subdigraph with at most $2k(n-1)$ arcs.* □

The proof of Theorem 9.3.1 can be turned into a polynomial algorithm for finding, in a given directed multigraph D with a specified vertex r, either k arc-disjoint out-branchings from r or a proof that no such set of branchings exist (see Exercise 9.7). As every k-(arc)-strong directed multigraph D has minimum semi-degree at least k, every certificate for the k-(arc)-strong connectivity of a D on n vertices must have at least $\delta_{\geq k}(D) \geq kn$ arcs. Hence we have the following corollary:

Corollary 12.0.4 *There exists a polynomial algorithm to find, in a given k-arc-strong digraph D, a spanning k-arc-strong subdigraph with at most twice the number of arcs in an optimal certificate.* □

12.1 Minimum Strong Spanning Subdigraphs

The central topic of the first three sections in this chapter is the problem of finding an optimal certificate for strong connectivity ($c \equiv 1$). We call this the MSSS PROBLEM (MSSS stands for Minimum Spanning Strong Subdigraph). Note that multiple arcs will never be part of an optimal solution to the MSSS problem, so below we consider digraphs rather than directed multigraphs.

The MSSS problem is a special case[1] of the so-called MINIMUM EQUIVALENT SUBDIGRAPH PROBLEM of a digraph. Here, for a given D, we wish to find a spanning subdigraph D' with as few arcs as possible such that, for every choice of $x, y \in V(D)$, D' contains an (x,y)-path if and only if D has one. This problem, which has many practical applications, has been considered several times in the literature, see, e.g., [10, 406, 537, 592, 593, 685, 779, 821].

A digraph $D = (V, A)$ is strong if and only if it contains an out-branching and an in-branching rooted at some vertex v. Furthermore, for every choice of in- and out-branchings B_v^-, B_v^+ rooted at v, the digraph $H = (V, A(B_v^-) \cup A(B_v^+))$ forms a strong spanning subdigraph of D on $2n-2-k$ arcs, where $k = |A(B_v^-) \cap A(B_v^+)|$. Thus we have the following easy but important observation.

Proposition 12.1.1 *Let $D = (V, A)$ be a strong digraph on n vertices, let $v \in V$ be arbitrary and let $k \leq n-2$ be a natural number. There exists a*

[1] It is, in fact, the most important ingredient since once we know the best subdigraph inside each strong component, we can contract each strong component to a vertex and consider the problem of finding a minimum equivalent subdigraph of an acyclic directed multigraph. That problem is solvable in polynomial time by Proposition 2.3.5.

strong spanning subdigraph of D with at most $2n-2-k$ arcs if and only if D contains an in-branching B_v^- and an out-branching B_v^+ with the same root so that $|A(B_v^-) \cap A(B_v^+)| \geq k$. □

Hence the MSSS problem is equivalent to finding, for an arbitrary vertex $v \in V(D)$, an out-branching B_v^+ and an in-branching B_v^- maximizing $|A(B_v^+) \cap A(B_v^-)|$.

Note that if we fix an out-branching B_v^+, then we can find an in-branching B_v^- maximizing $|A(B_v^-) \cap A(B_v^+)|$ in polynomial time using any polynomial algorithm for finding a minimum cost in-branching in a digraph (see Section 9.2).

Recall that $\mathrm{pcc}(D)$, the path-cycle covering number of D, is the smallest (positive) number of paths in a k-path-cycle factor of D. Define, for every digraph D, the number $\mathrm{pcc}^*(D)$ by

$$\mathrm{pcc}^*(D) = \begin{cases} 0 & \text{if } D \text{ has a cycle factor} \\ \mathrm{pcc}(D) & \text{otherwise.} \end{cases}$$

Proposition 12.1.2 *For every strongly connected digraph $D = (V, A)$ of order n, we have $\delta_{\geq 1}(D) = n + \mathrm{pcc}^*(D)$. Hence every spanning strong subdigraph of D has at least $n + \mathrm{pcc}^*(D)$ arcs.*

Proof: To see that $\delta_{\geq 1}(D) \leq n + \mathrm{pcc}^*(D)$ it suffices to observe that given any k-path-cycle factor $\mathcal{F}$ of a strong digraph D we can obtain a spanning digraph D' with minimum in- and out-degree one by adding at most $2k$ arcs from D to the arcs of $\mathcal{F}$ such that k of these arcs enter the initial vertex of each of the k paths in $\mathcal{F}$ and the other k leave the terminal vertices of the k paths. This D' will have at most $n+k$ arcs. To prove the other direction let D' be a spanning digraph of D with $\delta(D') \geq 1$ and $\delta_{\geq 1}(D)$ arcs. We claim that we can extract a $(\delta_{\geq 1}(D) - n)$-path-cycle factor from D'. To see this, it suffices to consider a connected component of $UG(D')$ and show that the desired inequality holds here. Below we assume that $UG(D')$ is connected.

Let $C_1, C_2, \ldots, C_r$ be any maximal collection of vertex-disjoint cycles in D' which we can obtain by extracting the vertices of each cycle one by one from D' and deleting the vertices of each cycle as we go along. After deleting these vertices from D' there is no cycle left and it is not difficult to show, using that $\delta(D') \geq 1$, that we can cover the remaining vertices of D' (those not on any of the cycles) by $(\delta_{\geq 1}(D) - n)$ paths. Details are left to the reader as Exercise 12.1. □

We prove in Section 12.2.1 that the inequality of Proposition 12.1.2 is, in fact, an equality for extended semicomplete digraphs. It was shown in [119] that this is also the case for semicomplete bipartite digraphs. However, already for quasi-transitive digraphs and semicomplete multipartite digraphs the inequality of Proposition 12.1.2 is not always an equality as such digraphs can have a cycle factor and still not be hamiltonian (see Sections 6.6 and 6.7).

12.1.1 Digraphs with High Minimum Degree

Theorem 12.1.3 below is due to Bang-Jensen, Huang and Yeo.

Theorem 12.1.3 *[107] Let D be a strong digraph, let R denote the complement of $UG(D)$ and let $c \geq 0$ be an integer. Suppose we have*[2]

$$\sum_{\{u \in V(D): d_R(u) > c\}} [d_R(u) - c] \leq q. \tag{12.1}$$

Then, in polynomial time, we can find a strong spanning subdigraph H of D such that $|A(H)| \leq n + c + \sqrt{2q}$.

Proof: We only prove the existence here and leave the complexity as Exercise 12.2. Recall the definition of a 1-maximal cycle from Section 8.1.5. Let z_1 be an arbitrary vertex, let $D_1 = D$ and let C_1 be a 1-maximal cycle containing z_1 in D_1. If C_1 is a hamiltonian cycle, then this can play the role of H. Otherwise contract C_1 into one vertex z_2 and let D_2 denote the resulting digraph (we delete multiple arcs if some are created as well as the loop created at z_2). Let C_2 be a 1-maximal cycle containing z_2 in D_2. If C_2 is a hamiltonian cycle in D_2, then stop. Otherwise contract C_2 into one vertex z_3 and let D_3 denote the resulting digraph. Continue this way until the current 1-maximal cycle C_j is a hamiltonian cycle in D_j.

Denote by H_i the subdigraph of D induced by the arcs of $C_1, C_2, \ldots, C_i$. It is easy to see that H_i has $|V(H_i)| + i - 1$ arcs and that $|V(H_i)| \geq i + 1$. Hence it suffices to show that the process above can continue for at most $c + \sqrt{2q}$ contraction steps.

Below we denote by $d_R(x, Y)$ the number of edges in R between the vertex x and the vertices in $Y \subset V - x$. Suppose x is a vertex in D_i which does not belong to H_i. We claim that $|(x, V(H_i))_R| \geq i$. The claim holds for $i = 1$ by Lemma 8.1.28. Suppose the claim does not hold for some step i above and let this i be chosen as small as possible. Then we have $d_R(x, V(H_{i-1})) = i - 1 = d_R(x, V(H_i))$. Since $|V(H_{i-1})| \geq i$, x has an arc to or from some vertex of $V(H_{i-1})$ in D and thus x is adjacent to z_i in D_i. Furthermore, x must be adjacent to every $y \in V(H_i) - V(H_{i-1})$ since we have $d_R(x, V(H_{i-1})) = d_R(x, V(H_i))$. However, this means that x is adjacent to every vertex of C_i, contradicting Lemma 8.1.28 since C_i is a 1-maximal cycle in the strong digraph D_i. This shows that $d_R(x, V(H_i)) \geq i$ for every step i above and the claim is proved.

Let j be chosen such that C_j is a hamiltonian cycle in D_j. By our remark above, H_j is a spanning strong subdigraph of D with $n + j - 1$ arcs. Hence we may assume that $j > c$. Let $p \geq 0$ be chosen such that $j = c + p + 1$. For each $i = c+1, \ldots, c+p+1$ we choose a vertex $x_i \in V(H_i) - V(H_{i-1})$. Using the definition of q and the fact that $d_R(x_i) \geq i - 1$ (since x_i has at least $i - 1$ non-neighbours in H_{i-1}), we get

[2] Here $d_R(u)$ is the degree of vertex u in the complement of $UG(D)$.

$$q \geq \sum_{i=c+1}^{c+p+1} [d_R(x_i) - c] \geq \sum_{i=0}^{p} i = \frac{p(p+1)}{2}.$$

Thus we have $p(p+1) \leq 2q$ which implies that $p \leq \sqrt{2q}$ (with equality only if $q = 0$). This implies that H_{c+p+1} is a spanning strong subdigraph of D with at most $n + c + \sqrt{2q}$ arcs. □

Corollary 12.1.4 *If D is a strong digraph on n vertices so that no vertex of D has more than c non-neighbours, then D contains a strong spanning subdigraph with at most $n + c$ arcs.* □

It is easy to derive the following consequence from Theorem 12.1.3 (simply suppress the subdividing vertices and consider D).

Corollary 12.1.5 *[107] Suppose $D = (V, A)$ satisfies the hypothesis of Theorem 12.1.3 and let $\tilde{D}$ be obtained from D by subdividing some arcs. Then $\tilde{D}$ contains a strong subdigraph $\tilde{H} = (\tilde{V}, \tilde{A})$ such that $V \subseteq \tilde{V}$ and $|\tilde{A}| \leq |\tilde{V}| + c + \sqrt{2q}$.* □

12.2 Polynomially Solvable Cases of the MSSS Problem

Since the MSSS problem is $\mathcal{NP}$-hard, it is natural to study the problem under certain extra assumptions. In order to find classes of digraphs for which we can solve the MSSS problem in polynomial time, we have to consider classes of digraphs for which we can solve the hamiltonian cycle problem in polynomial time.

Khuller, Raghavachari and Young considered the restriction when the digraph in question has no cycle with more than r arcs. Then the problem is known under the name SCCS_r [592]. In [593] it is shown that if one only considers digraphs with no cycle longer than 3, then the optimal certificate can be found in polynomial time. The algorithm is based on the following result.

Theorem 12.2.1 *[593] The $SCCS_3$ problem reduces in time $O(n^2)$ to the problem of finding a minimum edge-cover[3] in a bipartite graph.* □

This gives an $O(n^2 + m\sqrt{n})$ time algorithm for the SCCS_3 problem, since the problem of finding a minimum edge cover in a bipartite graph is equivalent to the problem of finding a maximum matching in such a graph [618]. The latter problem can be solved in time $O(\sqrt{n}m)$ (see Theorem 4.11.1). Although SCCS_3 is a very restricted case of the MSSS we shall see in Section 12.4.3

[3] An **edge-cover** of an undirected graph $G = (V, E)$ is a set of edges $E' \subset E$ such that every $v \in V$ is incident with at least one edge from E'.

that the result is actually useful in obtaining an approximation algorithm for a more general problem for directed multigraphs.

Already the SCCS_5 problem is $\mathcal{NP}$-hard and the SCCS_{17} is even MAX $\mathcal{SNP}$-hard, implying that there cannot exist a polynomial time approximation scheme for this problem, unless $\mathcal{P} = \mathcal{NP}$ [592].

12.2.1 The MSSS Problem for Extended Semicomplete Digraphs

The next result by Bang-Jensen and Yeo shows that the inequality in Proposition 12.1.2 is actually an equality for digraphs that are extensions of a semicomplete digraph. The main tool in the proof below is the characterization of the longest cycle in an extended semicomplete digraph given in Theorem 6.6.8.

Theorem 12.2.2 *[119] Let $D = (V, A)$ be a strong extended semicomplete digraph and let $\tilde{D} = (V, \tilde{A})$ be a minimum strong spanning subdigraph of D. Then $|\tilde{A}| = n + \mathrm{pcc}^*(D) = \delta_{\geq 1}(D)$.*

Proof: (Sketch) Let $D = S[H_1, H_2, \ldots, H_s]$, $s = |V(S)|$, be a strong extended semicomplete digraph, where the decomposition is such that S is semicomplete. For each $i = 1, 2, \ldots, s$ we let m_i denote the maximum number of vertices from H_i that can be covered by any cycle subdigraph of D. Let C be a longest cycle of D. By Theorem 6.6.8, C contains precisely m_i vertices from H_i for each $i = 1, 2, \ldots, s$. If D is hamiltonian, then $\mathrm{pcc}^*(D) = 0$ and there is nothing to prove. Hence we may assume below that $\mathrm{pcc}^*(D) > 0$. By Corollary 6.6.19, the extended semicomplete digraph $D' = D - C$ is acyclic. Let $k = \alpha(D')$. By Lemma 13.5.9, D' has a path-factor $P_1 \cup P_2 \cup \ldots \cup P_k$ where P_1 is a longest path in D', P_2 is a longest path in $D' - P_1$ and so on.

Start by letting $H := (V(C), A(C))$. Since P_1 is a longest path in D', its initial (terminal) vertex x (y) has no arc entering (going out) in D'. Thus, since D is strong, there exist arcs ux, yv such that u, v are vertices of H. Change H by adding the vertices of P and all arcs of P along with the arcs ux, yv to H. Now consider the path P_2 in $D' - P_1$. Using that P_2 is a longest path in $D' - P_1$, we again conclude that there must exist an arc from $V(H)$ to the initial vertex of P_2 and an arc from the terminal vertex of P_2 to H. Now it is easy to see how to continue and end up with a subdigraph H which is strong, spanning and has $n + k$ arcs.

It remains to prove that this is optimal. By the remark above $\mathrm{pcc}^*(D) > 0$, so by Proposition 12.1.2, it suffices to prove that $k = \mathrm{pcc}(D)$. Let $p = \mathrm{pcc}(D)$ and let $R_1, R_2, \ldots, R_p, \mathcal{Q}$ be an arbitrary p-path-cycle factor of D, where $\mathcal{Q}$ consists of one or more cycles and R_i is a path for $i = 1, 2, \ldots, p$. If some R_i contains two vertices from the same H_i, then we can replace it with a new path R_i' and a cycle C_i (Exercise 12.4). Doing this for all the paths $R_1, R_2, \ldots, R_p$ until none of these contains two independent vertices we end

up with a collection of paths $R'_1, R'_2, \ldots, R'_p$, where R'_i is the result of removing zero or more cycles from $D\langle R_i\rangle$[4]. Now consider the cycle subdigraph $\mathcal{Q}'$ we obtain by taking $\mathcal{Q}$ and all the cycles we extracted above. By the definition of m_i, $\mathcal{Q}'$ contains at most m_i vertices from H_i. Thus $\alpha(D - V(\mathcal{Q}')) \geq k$ and since no R'_i contains two independent vertices, it follows that $p \geq k$ must hold. □

Corollary 12.2.3 *[119] A minimum spanning strong subdigraph of a strong extended semicomplete digraph can be found in time* $O(n^{\frac{5}{2}})$.

Proof: Exercise 12.5. □

12.2.2 The MSSS Problem for Quasi-Transitive Digraphs

We first give a lower bound for the number of arcs in any minimum spanning strong subdigraph of an arbitrary given strong quasi-transitive digraph. This bound can be calculated in polynomial time using Gutin's algorithm for finding a hamiltonian cycle in a quasi-transitive digraph (Theorem 6.7.4) as well as the algorithm of Theorem 6.7.5. We prove that this lower bound is also attainable for quasi-transitive digraphs. The proof of this uses Theorem 6.6.8.

Definition 12.2.4 *Let* D *be a strong quasi-transitive digraph and define* $\mathrm{pc}^*(D)$ *by* $\mathrm{pc}^*(D) = 0$ *if* D *is hamiltonian and* $\mathrm{pc}^*(D) = \mathrm{pc}(D)$ *otherwise.*

Lemma 12.2.5 *For every strongly connected quasi-transitive digraph* D*, every spanning strong subdigraph of* D *has at least* $n + \mathrm{pc}^*(D)$ *arcs.*

Proof: Exercise 12.7. □

In fact Lemma 12.2.5 holds for arbitrary digraphs. This is not in contradiction with Theorem 12.2.2 since $\mathrm{pcc}^*(D) = \mathrm{pc}^*(D)$ for every strong extended semicomplete multipartite digraph by Theorems 6.6.2 and 6.6.5. Below we characterize the optimal solution to the MSSS problem for quasi-transitive digraphs and show that the problem is polynomially solvable.

Theorem 12.2.6 *[106] Every minimum spanning strong subdigraph of a quasi-transitive digraph has precisely* $n + \mathrm{pc}^*(D)$ *arcs. Furthermore, we can find a minimum spanning strong subdigraph in time* $O(n^4)$.

Proof: Let $D = S[W_1, W_2, \ldots, W_s]$, $s = |S| \geq 2$, be the decomposition of a strong quasi-transitive digraph D according to Theorem 2.7.5. Using the algorithm of Theorem 6.7.4 we can check whether D is hamiltonian and find a hamiltonian cycle if one exists. If D is hamiltonian, then any hamiltonian

[4] Observe that by the definition of p, no R'_i is empty.

cycle is the optimal spanning strong subdigraph. Suppose below that D is not hamiltonian. Then in particular we have $\mathrm{pc}^*(D) = \mathrm{pc}(D)$ by Definition 12.2.4.

Let $D_0 = S[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_s}]$ be the extended semicomplete digraph one obtains by deleting all arcs inside each W_i (that is, $n_i = |V(W_i)|$ for all $i \in [s]$).

For each $i = 1, 2, \ldots, s$, let m_i denote the maximum number of vertices which can be covered in H_i by any cycle subdigraph of D_0. According to Theorem 6.6.8, every longest cycle C in D_0 contains exactly m_i vertices from H_i, $i = 1, 2, \ldots, s$. By Theorem 6.6.8, we can find C in time $O(n^3)$. Let

$$k = \max\{\mathrm{pc}(W_i) - m_i : i = 1, 2, \ldots, s\}. \tag{12.2}$$

Note that by Theorem 6.7.3, $k \geq 1$ since D has no hamiltonian cycle. Let $m_i^* = \max\{\mathrm{pc}(W_i), m_i\}$, $i = 1, 2, \ldots, s$, and define the extended semicomplete subdigraph D^* of D by $D^* = S[H_1^*, H_2^*, \ldots, H_s^*]$, where H_i^* is an independent set containing m_i^* vertices for $i = 1, 2 \ldots, s$. Since vertices inside an independent set of D have the same in- and out-neighbours, we may think of C as a longest cycle in D^* (i.e., C contains precisely m_i vertices from H_i^*, $i = 1, 2, \ldots, s$). By Corollary 6.6.19, $D^* - C$ is acyclic and by Lemma 13.5.9, $D^* - C$ can be covered by k paths $P_1^*, P_2^*, \ldots, P_k^*$ such that P_i^* is a longest path in $D^* - (V(P^*) \cup \ldots \cup V(P_{i-1}^*))$ for $i = 1, 2, \ldots, k$.

It follows from the proof of Theorem 12.2.2 that we can glue P_1^* onto C and then P_2^* onto the resulting graph etc., until we obtain a spanning strong subdigraph D^{**} of D^* with $|V^*| + k$ arcs.

Now we obtain a spanning strong subdigraph of the quasi-transitive digraph D as follows. Since $m_i^* \geq \mathrm{pc}(W_i)$ for $i = 1, 2, \ldots, s$, each W_i contains a collection of $t_i = m_i^*$ paths $P_{i1}, P_{i2}, \ldots, P_{it_i}$ such that these paths cover all vertices of W_i. Such a collection of paths can easily be constructed from a given collection of $\mathrm{pc}(W_i)$ paths which cover $V(W_i)$. Let $x_{i1}, x_{i2}, \ldots, x_{it_i}$ be the vertex set of H_i^*, $i = 1, 2, \ldots, s$. Replace x_{ij} in D^{**} by the path P_{ij} for each $j = 1, 2, \ldots, t_i$, $i = 1, 2, \ldots, s$. We obtain a spanning strong subdigraph D' of D. The number of arcs in D' is

$$\begin{aligned} A(D') &= \sum_{i=1}^{s} (|W_i| - m_i^*) + (|V^*| + k) \\ &= (n - |V^*|) + (|V^*| + k) \\ &= n + k. \end{aligned} \tag{12.3}$$

It remains to argue that D' is the smallest possible. By Lemma 12.2.5, it suffices to prove that $\mathrm{pc}^*(D) \geq k$.

Since this part is similar to the proof of Theorem 12.2.2 we only sketch how to prove it. Let $P_1, P_2, \ldots, P_r$ be an optimal path cover of D. Path-contract all subpaths that lie inside some W_i and let $P_1', \ldots, P_r'$ denote the resulting paths. Delete all arcs that still remain inside each W_i after this

contraction. That way we obtain a path cover of an extended semicomplete digraph which we may consider as a subdigraph of D_0.

As in the proof of Theorem 12.2.2, we can continue replacing paths in the current collection by a cycle or a path until every path in the current collection contains at most one vertex from H_i. Let $P_1'', P_2'' \ldots, P_r''$ be the final collection after removing all such cycles. Using an argument analogous to the last part of the proof of Theorem 12.2.2, we now conclude that $r \geq k$, implying that the subdigraph D' is optimal. □

12.3 Approximation Algorithms for the MSSS Problem

By Proposition 12.0.3 there is a 2-approximation for the MSSS problem. It also follows from Lemma 5.3.5 and Corollary 5.3.7 that, in linear time, we can find a strong spanning subdigraph on at most $2n-2$ arcs in any strong digraph on n vertices. The purpose of this section is to discuss better approximation algorithms. We start with a simple but useful observation.

Lemma 12.3.1 *It is sufficient to consider digraphs with no cut vertex.*

Proof: Suppose D contains a cut vertex v and that $UG(D)-v$ has connected components $X_1, X_2, \ldots, X_k$, $k \geq 2$. Then the arc set of every spanning strong subdigraph of D is the disjoint union of the arc sets of spanning strong subdigraphs of $D_1, D_2, \ldots, D_k$, where $D_i = D\langle X_i + v\rangle$. Hence we can decompose the problem into k problems, one for each D_i. Combining α-approximations for each D_i, $i \in [k]$, we obtain an α-approximation for D. □

12.3.1 A Simple $\frac{7}{4}$-Approximation Algorithm

Theorem 12.3.2 *In time $O(n^2)$ we can find a $\frac{7}{4}$-approximation for the MSSS problem in a strong digraph with no cut vertex.*

Proof: We only prove the existence and leave the complexity to the reader as Exercise 12.8. Given a strong digraph D on n vertices having no cut vertex we proceed as follows. If $n = 2$, we return D as the optimal solution so we may assume $n \geq 3$. Since D has no cut vertex it contains a cycle of length at least 3. Let P_0 be such a cycle and add to it a maximal sequence $P_1, P_2, \ldots, P_s$ of ears of size at least three (as defined in Definition 5.3.1, but without ears of size two or one added). Let $X = V(P_0) \cup \ldots \cup V(P_s)$.

We claim that $Y = V(D) - X$ is an independent set. Assume this is not true and let uv be an arc of Y so that v dominates some vertex in X. Since we cannot add any new ear of size at least three to X, it follows that every (X, u)-path must pass through v (in fact, v is the first vertex after X on any such path) and thus uv is contained in a strong component W of $D\langle Y\rangle$. Now it follows from the fact that v is not a cut vertex that there is an edge e in

$UG(D)$ with precisely one vertex p in W and $p \neq v$. Let q be the other vertex of e. If $q \in X$, then it is easy to see that we can add another ear of size at least three to X, contradicting the maximality of the sequence $P_1, P_2, \ldots, P_s$. So $q \in Y$. But now it follows from the fact that D is strong and $q \notin W$ that if $pq \in A(D)$, then there is a directed path from q to X which avoids W and if $qp \in A(D)$, then there is a directed path from X to q which avoids W and again we find a new ear of size at least three. Thus in both cases we obtain a contradiction to the maximality of $P_1, P_2, \ldots, P_s$. Therefore Y is independent.

Let D'_X be the strong spanning subdigraph of $D\langle X\rangle$ consisting of all arcs from $P_0, P_1, \ldots, P_s$. Since each ear P_i, $i \leq s$, adds at least two new vertices to X, we have $s \leq \frac{|X|-3}{2}$. Hence we get

$$|A(D'_X)| = |X| + s \leq |X| + \frac{|X|-3}{2} < \frac{3}{2}|X|.$$

By adding each vertex of Y to D'_X as an ear of size two we obtain a strong spanning subdigraph D' of D with $m' = |A(D'_X)| + 2(n - |X|)$ arcs. By the calculation above,

$$m' < \frac{3}{2}|X| + 2(n - |X|) \leq 2n - \frac{|X|}{2}. \tag{12.4}$$

If $|X| \geq \frac{n}{2}$, then we get from (12.4) that $m' \leq \frac{7}{4}n$ and we are done. So assume $|X| < \frac{n}{2}$. It follows from the fact that Y is independent that every strong spanning subdigraph of D must contain at least $2(n - |X|)$ arcs (one in and out from every vertex of Y). Thus the approximation ratio α which we obtain by returning D' as our solution is no worse than $\frac{m'}{2(n-|X|)} \leq \frac{2n-\frac{|X|}{2}}{2(n-|X|)}$. This number is strictly increasing in $|X|$ and as $|X| < \frac{n}{2}$ we have $\alpha \leq \frac{7}{4}$. □

12.3.2 Better Approximation Algorithms

Khuller, Raghavachari and Young [592] gave an approximation algorithm with approximation guarantee 1.65 for the MSSS problem. The idea in the algorithm from [592] is to find a long cycle, contract it and continue recursively. As a slight improvement the authors were able to show in [593] that this approach can be performed in such a way that one obtains a solution in polynomial time with no more than about 1.62 times the size of an optimum solution.

Extending the idea of contracting cycles Vetta [886] improved this bound to the following, which is currently the best approximation guarantee for general digraphs.

Theorem 12.3.3 *[886] There exists a polynomial algorithm which given a strong digraph D returns a strong spanning subdigraph H of D with at most $\frac{3}{2}$ times the size of a minimum spanning strong subdigraph of D.* □

The proof of Theorem 12.3.3 uses several tricks to perform cycle contraction in an appropriate way so as to guarantee the desired approximation ratio. Vetta's algorithm also requires the calculation of a spanning subdigraph D' with $\delta^0(D') \geq 1$ and hence is not linear.

In [932] Zhao, Nagamochi and Ibaraki showed that by contracting appropriately chosen cycles one can obtain a linear time $\frac{5}{3}$-approximation algorithm for the MSSS problem.

The following corollary follows from the proof of Theorem 12.3.2. To see this it suffices to note that when we create D' from D'_X above by adding two arcs from every vertex incident to each vertex in Y, we may do so without creating a 2-cycle.

Corollary 12.3.4 *Every strong digraph D without a cut vertex contains a minimum spanning strong subdigraph without 2-cycles.* □

By Corollary 12.3.4, if we can prove that an arc xy of a 2-cycle must be contained in every optimal solution to the MSSS problem for D, then we can reduce the problem to a digraph with fewer arcs by considering $D - yx$.

We finish this section with a couple of remarks on the MSSS problem for digraphs in which every vertex has high degree. Bessy and Thomassé proved that one can bound the size of a minimum spanning strong subdigraph of D in terms of the independence number $\alpha(D)$ of D.

Theorem 12.3.5 *[157] Every strong digraph D has a spanning strong subdigraph with at most $|V(D)| + 2\alpha(D) - 2$ arcs.* □

By Corollary 12.1.4 the following holds:

Theorem 12.3.6 *Every strong digraph D with $\Delta(\overline{UG(D)}) \leq \frac{n}{r}$ contains a spanning strong digraph with at most $(1 + \frac{1}{r})n$ arcs. Furthermore, such a subdigraph can be found in polynomial time.* □

12.4 Small Certificates for k-(Arc)-Strong Connectivity

In this section we present some results by Cheriyan and Thurimella [209] which show that we can approximate the size of a smallest k-(arc)-strong spanning subdigraph better, the higher k is. We point out that the results for k-arc-strong connectivity are only valid for digraphs[5] and consider the case of directed multigraphs in Section 12.4.3.

[5] As usual, for vertex-strong connectivity, parallel arcs play no role so in the case of k-strong connectivity we may assume that we are dealing with a digraph.

12.4.1 Small Certificates for k-Strong Connectivity

Cheriyan and Thurimella gave an approximation algorithm with a very good approximation guarantee by combining some fairly elementary results on subgraphs of (di)graphs with Mader's powerful result on anti-directed trails and k-critical arcs (Theorem 5.6.11). We start with the two subgraph results and then describe the simple algorithm from [210].

Proposition 12.4.1 *[210] Let $B = (V, E)$ be a bipartite graph with minimum degree k. Let $E' \subset E$ be a minimum cardinality subset of E with the property that $B' = (V, E')$ has minimum degree $k-1$. Then $|E'| \leq |E| - |V|/2$ and this bound is best possible.* □

Theorem 12.4.2 *[210] There exists a polynomial algorithm which, given a k-strong digraph $D = (V, A)$, returns a spanning k-strong spanning subdigraph $D'' = (V, A'')$ of D such that $|A''| \leq (1 + \frac{1}{k})|A^*_{opt}|$, where $D^* = (V, A^*_{opt})$ is any optimal certificate for the k-strong connectivity of D.*

Proof: Let $\mathcal{B}$ be the following algorithm:

Input: A directed graph $D = (V, A)$ and a number k such that D is k-strong.
Output: A small certificate $\tilde{D} = (V, \tilde{A})$ for k-strong connectivity of D.

1. Use the algorithm $\mathcal{A}$ of Proposition 12.0.2 to find a subset $A' \subset A$ of size $\delta_{\geq k-1}(D)$ such that the digraph $D' = (V, A')$ has $\delta^0(D') \geq k - 1$;
2. Find an inclusion-wise minimal subset $A'' \subset A - A'$ with the property that $\tilde{D} = (V, A' \cup A'')$ is k-strong;
3. Return $\tilde{D}$.

Clearly $\tilde{D} = (V, A' \cup A'')$ is k-strong, so we can concentrate on the approximation factor and the running time.

To see that the approximation factor is as claimed, let $D^* = (V, A^*_{opt})$ denote an arbitrary optimal certificate for k-strong connectivity of D. Clearly we have

$$|A'| \leq |A^*_{opt}|. \tag{12.5}$$

To bound the size of A'' we use Theorem 5.6.11. We claim that $D'' = (V, A'')$ has no anti-directed trail. Suppose T is an anti-directed trail in D''. Note that T is a subdigraph of $\tilde{D}$. Hence we can apply Theorem 5.6.11 to $\tilde{D}$. Now it follows from the fact that every arc of A'' is k-critical in $\tilde{D}$ that only (b) or (c) can hold in Theorem 5.6.11 when applied to $\tilde{D}$. However, by the choice of A', neither (b) nor (c) can hold in $\tilde{D}$ since every source (sink) of T has out-degree (in-degree) at least $k + 1$ in D. Thus T cannot exist and D'' has no anti-directed trail. From this it follows, by considering the bipartite representation $BG(D'')$, that

$$|A''| \leq 2|V| - 1. \tag{12.6}$$

We leave the proof of this as Exercise 12.10 (recall the definition of $BG(D)$ in Chapter 1).

Combining (12.5) and (12.6), it is easy to see that the approximation guarantee of $\mathcal{B}$ is at least as good as $(1 + \frac{2}{k})$. However, using Proposition 12.4.1, we can do a little better. Let A^{**} be a minimum cardinality subset of A^*_{opt} so that the spanning subdigraph $D^{**} = (V, A^{**})$ has $\delta^0(D^{**}) \geq k - 1$. Consider $BG(D^*)$ and the edge sets E^*, E^{**} corresponding to A^*_{opt} and A^{**}. By Proposition 12.4.1

$$\begin{aligned} |A^{**}| = |E^{**}| &\leq |E^*| - |V(BG(D^*))|/2 \\ &= |A^*_{opt}| - |V|. \end{aligned} \tag{12.7}$$

By the choice of A' we have $|A'| \leq |A^{**}|$ and combining (12.6) and (12.7) gives

$$\begin{aligned} \frac{|\tilde{A}|}{|A^*_{opt}|} &\leq \frac{|A^*_{opt}| - |V| + (2|V| - 1)}{|A^*_{opt}|} \\ &\leq 1 + \frac{1}{k}, \end{aligned} \tag{12.8}$$

since clearly $|A^*_{opt}| \geq k|V|$.

It remains to prove that $\mathcal{B}$ can actually be performed in polynomial time. Step 1 is performed by the polynomial algorithm $\mathcal{A}$ whose existence is proved in Exercise 12.9. Step 3 can be implemented by starting from D and deleting arcs of $\bar{A}$ one by one until every remaining arc from $\bar{A}$ is k-critical. Clearly this part can be done in polynomial time, using any algorithm for checking whether a digraph is k-strong. □

The authors claimed in [210] that the running time of the algorithm can be made $O(k|A|^2)$.

Bang-Jensen posed the problem whether a similar result as Theorem 12.4.7 holds for optimal certificates for k-strong connectivity in tournaments.

Problem 12.4.3 *[76] Does there exist a function $g = g(k)$ such that every k-strong tournament contains a spanning k-strong subdigraph with at most $kn + g(k)$ arcs?*

12.4.2 Small Certificates for k-Arc-Strong Connectivity

Cheriyan and Thurimella showed that for arc-strong connectivity one can also approximate the size of an optimal certificate better the higher the arc-strong connectivity is.

Theorem 12.4.4 *[210] There exists a polynomial algorithm which given a k-arc-strong digraph $D = (V, A)$ returns a spanning k-arc-strong subdigraph $D' = (V, A')$ of D such that $|A'| \leq (1+4/\sqrt{k})|A_{opt}|$, where $|A_{opt}|$ denotes the number of arcs in an optimal certificate for k-arc-strong connectivity. The running time of the algorithm is $O(k^3|V|^3 + |A|^{1.5}\log^2(|V|))$.*

Proof: The idea is similar to the vertex-strong connectivity case so we will only sketch the proof here. Let $D = (V, A)$ be k-arc-strong. First find, using the algorithm $\mathcal{A}$ of Proposition 12.0.2, a minimum cardinality subset $U \subset A$ such that $H = (V, U)$ has $\delta^0(H) \geq k$. Then find an inclusion-wise minimal subset $U' \subset A - U$ such that $\tilde{H} = (V, U \cup U')$ is k-arc-strong. As in the proof of Theorem 12.4.2, the key step is to estimate the size of U', since $|U|$ is clearly at most the size of an optimal solution.

To estimate $|U'|$ we use the following definition. An arc uv of a k-arc-strong digraph W is **special** if $W - uv$ is not k-arc-strong and furthermore $d^+_W(u), d^-_W(v) \geq k+1$. Clearly each arc in the set U' is special in the digraph $\tilde{H}$. Hence we can apply the following estimate.

Theorem 12.4.5 *[210] Let $k \geq 1$ be an integer and let $W = (V, A)$ be a k-arc-strong digraph. The number of special arcs in W is at most $4\sqrt{k}|V|$.* □

Combining this with the fact that $|A'|$ is no more than the size of an optimal certificate the theorem follows. For the complexity bound we refer to [210]. □

Every integer $k \geq 0$ determines two unique integers τ_k, ω_k with $0 \leq \omega_k \leq \tau_k$ such that $k = \tau_k(\tau_k + 1) + \omega_k$. Now let $\sigma(k) = \tau_k + \frac{\omega_k}{\tau_k+1}$. Gabow [376] studied the structure of special arcs and was able to improve the bound in Theorem 12.4.5 as follows.

Theorem 12.4.6 *[376] Let $k \geq 1$ be an integer and let $W = (V, A)$ be a k-arc-strong digraph. For $k \geq 15$ the number of special arcs in W is at most $\sigma(k)|V|$ and this bound is tight. For $5 < k < 15$ there are at most $5|V|$ special arcs.* □

The remainder of this section deals with the case when D is a tournament. The results described are due to Bang-Jensen, Huang and Yeo. Let $n \geq 3$ and $k \geq 1$ be two integers. We define $f(n, k)$ to be the smallest integer such that every k-arc-strong tournament on n vertices contains a k-arc-strong spanning subdigraph with at most $f(n, k)$ arcs. By Proposition 12.0.3 and the fact that every vertex in a k-arc-strong digraph has out-degree at least k, we have

$$nk \leq f(n, k) \leq 2k(n-1). \tag{12.9}$$

Since every strong tournament is hamiltonian we have $f(n, 1) = n$.

For all $k \geq 1$ and $n \geq 5k+2$ we define $\mathcal{T}_{n,k}$ as the class of tournaments on n vertices that can be obtained from a transitive tournament A on k vertices

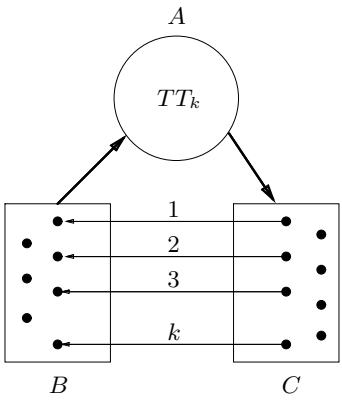


Figure 12.1 The structure of the tournaments in $\mathcal{T}_{n,k}$. The tournament A is the transitive tournament on k vertices, B and C are arbitrary k-arc-strong tournaments. The bold arcs $B \rightarrow A, A \rightarrow C$ indicate that all possible arcs are present in that direction. There are exactly k arcs from C to B and all other arcs go from B to C.

and two k-arc-strong tournaments B, C as shown in Figure 12.1. It is not difficult to show that each tournament in $\mathcal{T}_{n,k}$ is k-arc-strong (Exercise 12.3).

Let T be any member of $\mathcal{T}_{n,k}$. Observe that every k-arc-strong subdigraph D of T must contain at least $k(k+1)/2$ arcs from B to A and exactly k arcs from C to B (there are no more). Hence we have $[\sum_{x \in B} d_D^+(x)] - [\sum_{x \in B} d_D^-(x)] \geq k(k+1)/2 - k$, implying that $\sum_{x \in B} d_D^+(x) \geq k|B| + k(k-1)/2$. This implies that D has at least $nk + k(k-1)/2$ arcs. Thus the tournaments in $\mathcal{T}_{n,k}$ show that $f(n,k) \geq nk + ck^2$ for some constant $c > 0$ and hence the following result is best possible in terms of the exponent on k.

Theorem 12.4.7 *[107] For every $n \geq 3$ and $k \geq 1$, every k-arc-strong tournament T on n vertices contains a spanning k-arc-strong subdigraph D' with at most $nk + 136\ k^2$ arcs.* □

The following result shows that for tournaments, $\delta_{\geq k}$ can be bounded nicely.

Theorem 12.4.8 *[107] For every tournament with $\delta^0(T) \geq k$ we have $\delta_{\geq k}(T) \leq nk + k(k+1)/2$. Furthermore, if T is also k-arc-strong, then we have $\delta_{\geq k}(T) \leq nk + k(k-1)/2$.*

Proof: Let $T = (V, A)$ be a tournament on n vertices with $\delta^0(T) \geq k$. Form a flow network $\mathcal{N}$ with vertex set $U = V_S \cup V_T \cup \{p, q\}$, where $V_S = \{v_S : v \in V\}$, $V_T = \{v_T : v \in V\}$, and arc set $\{pv_S : v \in V\} \cup \{v_S w_T : vw \in A\} \cup \{v_T q : v \in V\}$. Every arc of the kind $v_S w_T$ has capacity 1 and lower bound 0 and each arc a in the set $\{pv_S : v \in V\} \cup \{v_T q : v \in V\}$ has capacity n and a lower bound $l(a) = k$. Observe that if x is a feasible integer-valued (p, q)-flow in $\mathcal{N}$ of value M (the flow out of p) then the spanning

subdigraph D' of T that we obtain by taking those arcs $vw \in A$ for which $x(v_S w_T) = 1$ has precisely M arcs and $\delta^0(D') \geq k$. The other direction holds as well. Thus the minimum number of arcs in a spanning subdigraph D' of T with $\delta^0(D') \geq k$ is equal to the minimum value of a feasible integer valued (p, q)-flow in $\mathcal{N}$. By Theorem 4.9.1, this value is equal to the maximum of $l(R, \bar{R}) - u(\bar{R}, R)$ over all partitions of U into two sets $R, \bar{R}$ where $p \in R$ and $q \in \bar{R}$.

Let $(R, \bar{R})$ be an arbitrary partition as above and denote by X, Y, Z, W the following sets:

- $X = \{v \in V : v_S \in \bar{R} \text{ and } v_T \in R\}$, $Y = \{v \in V : v_S \in \bar{R} \text{ and } v_T \in \bar{R}\}$,
- $Z = \{v \in V : v_S \in R \text{ and } v_T \in \bar{R}\}$, $W = \{v \in V : v_S \in R \text{ and } v_T \in R\}$.

Then $U = X \cup Y \cup Z \cup W$ and it is easy to check that we have $l(R, \bar{R}) \leq k(n + |X| - |Z|)$ and $u(\bar{R}, R) \geq |X|(|X| - 1)/2$ (every arc in X contributes one to $u(\bar{R}, R)$). Thus we have

$$l(R, \bar{R}) - u(\bar{R}, R) \leq kn + k|X| - k|Z| - |X|(|X| - 1)/2. \tag{12.10}$$

This implies that $l(R, \bar{R}) - u(\bar{R}, R) \leq kn + k(k+1)/2$ with equality only if $|Z| = 0$. Furthermore, if T is k-arc-strong then it is easy to see that either $|Z| \geq 1$ or there are at least k arcs from Y to $X \cup W$ in T. In both cases we conclude that $l(R, \bar{R}) - u(\bar{R}, R) \leq kn + k(k-1)/2$. Since $(R, \bar{R})$ was chosen arbitrarily the result now follows. □

Bang-Jensen, Huang and Yeo also made the following conjecture, which if true would imply that $f(n, k) = nk + k(k-1)/2$. Furthermore, since a minimum spanning subdigraph D with $\delta^0(D) \geq k$ can be found in polynomial time, the truth of the conjecture would imply that one can find an optimal certificate for k-arc-strong connectivity in polynomial time for every k-arc-strong tournament T.

Conjecture 12.4.9 *[107] For each $k \geq 1$ and for every k-arc-strong tournament T every optimal certificate for the k-arc-strong connectivity of T has precisely $\delta_{\geq k}(T)$ arcs, that is, the minimum possible.*

Conjecture 12.4.10 *[107] There exists a polynomial algorithm for finding a minimum k-arc-strong spanning subdigraph of a given k-arc-strong tournament T.*

12.4.3 Certificates for Directed Multigraphs

In this section, for a given k-arc-strong directed multigraph D, we denote by $OPT_k(D)$ the size of a minimum spanning k-arc-strong directed multigraph of D.

The following example, due to Gabow [375], shows that the approach in Section 12.4.2 does not always work when the input is a directed multigraph. Recall that the approach for a given digraph $D = (V, A)$ was to find a minimum cardinality set of arcs $A' \subseteq A$ so that $\delta^0(H) \geq k$, where $H = (V, A')$, and then find an arbitrary inclusion-wise minimal set of arcs $A'' \subseteq A - A'$ so that adding these arcs to H results in a k-arc-strong subdigraph of D. Now consider the k-arc-strong directed multigraph W_k which we obtain from a complete biorientation of an undirected n-cycle, n even, by replacing each arc by k-copies of the arc. Clearly W_k is k-arc-strong and if we label its vertices $0, 1, 2, \ldots, n-1$ around the original cycle, we can obtain the following valid choices for A' and A'': A' is the set of all arcs between vertex $2i$ and $2i+1$, $i = 0, 1, \ldots, \frac{n}{2} - 1$ and A'' is the set of all arcs between vertex $2i-1$ and $2i$, $i = 1, \ldots, \frac{n}{2} - 1$. Then $|A' \cup A''| = 2k(n-1)$, but by just taking k copies of the directed k-cycle we obtain an optimal k-arc-strong subdigraph of W_k with kn arcs. Thus $|A' \cup A''|/OPT_k(D) \rightarrow 2$ as $n \rightarrow \infty$.

The following lemma is due to Gabow [376].

Lemma 12.4.11 *[376] Let $D = (V, A)$ be a directed multigraph which contains K arc-disjoint out-branchings rooted at some vertex $s \in V$. Suppose that $H = (V, \tilde{A})$ is a spanning subdigraph of D and that H has L arc-disjoint out-branchings rooted in s. For every integer $0 \leq r \leq K - L$ there exists a set of arcs $A_r \subseteq A - \tilde{A}$ so that the directed multigraph $H + A_r$ has $L + r$ arc-disjoint out-branchings rooted in s and no vertex of V is the head of more than r arcs from A_r.*

Proof: Exercise 12.12. □

Lemma 12.4.12 *[376] If D is a k-arc-strong directed multigraph with no cycle of length more than 3, then $OPT_k(D) \geq kOPT_1(D)$.* □

The following result, due to Gabow, shows that it is indeed possible to obtain a better approximation guarantee than 2 for directed multigraphs. Note that contrary to the case of directed graphs the approximation guarantee becomes worse the higher k is.

Theorem 12.4.13 *[376] There exists a polynomial $(2 - \frac{1}{3k})$-approximation algorithm for approximating the smallest spanning k-arc-strong directed multigraph of a k-arc-strong directed multigraph D.*

Proof: The algorithm is based on the cycle contraction idea of Khuller et al. for finding an approximate solution to the MSSS problem (see Section 12.3). There are two phases in the algorithm. First we contract cycles of length at least four as long as possible. Let H denote the resulting directed multigraph. Apply the polynomial algorithm of Theorem 12.2.1 to find an optimal solution to the MSSS problem for H and let A_H denote the set of arcs in this solution. Now let $W = (V, A')$ be the strong spanning subdigraph

of D that we obtain by taking the arcs of all the contracted cycles plus the arcs corresponding to A_H back in[6] D. In the second phase of the algorithm we fix an arbitrary vertex $s \in V$ and apply Lemma 12.4.11 and its obvious analogue for in-branchings to W, s with $r = k-1$ to get a set of arcs A'' so that $W + A''$ has k arc-disjoint out-branchings rooted in s and a set of arcs A''' so that $W + A'''$ has k arc-disjoint in-branchings rooted in s. Clearly $W + A'' + A'''$ is k-arc-strong, so it remains to bound the number of arcs in $A' \cup A'' \cup A'''$. Let c be the total number of arcs in the cycles of length at least four which were contracted in the first phase, let h be the number of vertices in H and let $a_H = |A_H|$. Then $|A'| = c + a_H$ by construction and it is easy to see that $n \geq \frac{3c}{4} + h$ and $a_H \leq 2h$ hold. Hence, as $OPT_k(D) \geq nk$ we have

$$OPT_k(D) \geq k(\frac{3c}{4} + \frac{a_H}{2}). \tag{12.11}$$

Since contraction of cycles does not increase the size of a smallest k-arc-strong spanning subdigraph we have $OPT_k(D) \geq OPT_k(H)$. By Lemma 12.4.12 this means that

$$OPT_k(D) \geq ka_H. \tag{12.12}$$

Now it is easy to show that $|A'| \leq \frac{5OPT_k(D)}{3k}$ and using Lemma 12.4.11 we see that $\max\{|A''|, |A'''|\} \leq (k-1)n \leq (k-1)OPT_k(D)/k$. Thus we have

$$|A'| + |A''| + |A'''| \leq \frac{5OPT_k(D)}{3k} + 2(k-1)OPT_k(D)/k = (2 - \frac{1}{3k})OPT_k(D),$$

as desired. □

Gabow, Goemans, Tardos and Williamson proved the following result which implies that also for directed multigraphs one can get a better approximation the higher k is. Note that for digraphs (no multiple arcs) the bound obtained from Gabow's improvement (Theorem 12.4.6) of the bound in Theorem 12.4.5 is still better than the bound below.

Theorem 12.4.14 *There exists a polynomial algorithm which is based on LP-rounding and achieves an approximation ratio of* $1 + \frac{2}{k}$ *for the problem of finding a minimum size k-arc-strong spanning subdigraph of a k-arc-strong directed multigraph.* □

[6] If there are several choices for an arc when we do this blow up process, just take an arbitrary among the possible choices.

12.5 Minimum Weight Strong Spanning Subdigraphs

Khuller, Raghavachari and Young posed the following problem concerning the weighted version of the MSSS problem. Here the goal is to find in a strong digraph with non-negative cost on the arcs, a spanning strong subdigraph of minimum cost.

Problem 12.5.1 *[592] Does there exist a μ-approximation algorithm for minimum cost strong connectivity certificates for some $\mu < 2$?*

The existence of a polynomial algorithm with approximation guarantee 2 follows from the fact that finding a minimum cost in-branching (out-branching) with a given root can be done in polynomial time (see Section 9.2). Indeed, if B_r^- (B_r^+) is a minimum cost in-branching (out-branching) rooted at r, then $D' = (V, A(B_r^-) \cup A(B_r^+))$ is strong and clearly has cost at most twice the optimum. Below, when we refer to **the branchings algorithm**, we mean the algorithm which chooses a root r and then finds a minimum cost out-branching B_r^+ and a minimum cost in-branching B_r^- and returns $D' = (V, A(B_r^-) \cup A(B_r^+))$. The example in Figure 12.2 shows that the approximation guarantee of the branchings algorithm cannot be better than 2. See also Exercise 12.11.

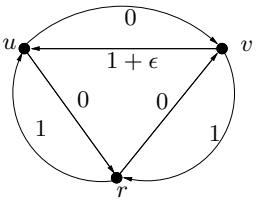


Figure 12.2 A bad example for the branchings algorithm [695, Figure 14]. The algorithm returns a solution of value 2, whereas the optimum strong spanning subdigraph has cost $1 + \epsilon$.

In order to understand why the branchings algorithm may perform poorly, it is useful to consider the example in Figure 12.2. The arc vu is not included in any of the two branchings B_r^+, B_r^- found by the branching algorithm, even though this arc is very important in the optimal solution. On the other hand, if we lower the cost of the arc only slightly from $1 + \epsilon$ to $1 - \epsilon$, then it will be included in both of the branchings B_r^+, B_r^- and the branchings algorithm will return the optimum solution.

This kind of observation is the intuition behind the so-called **drop algorithm** by Khuller, Raghavachari and Zhu [594]. They calculate the following number for each arc a:

$$drop(a) = \frac{(c(B_I) + c(B_O)) - (c(B'_I) + c(B'_O))}{c(a)}, \tag{12.13}$$

where $c(B_I)$ ($c(B_O)$) is the cost of a minimum cost in-branching (out-branching) rooted in the fixed root r in D and $c(B'_I), c(B'_O)$ are the corresponding values when the cost of a is dropped to 0.

The drop algorithm is iterative. In each iteration it determines the most important arc as the one with the highest drop value (according to (12.13)) and drops its cost to zero. The algorithm terminates when the current digraph has an in-branching and an out-branching of cost zero from r. The algorithm then returns the union of 0-cost out-branchings B_r^+, B_r^-. With respect to the original cost this solution is still a 2-approximation.

Lemma 12.5.2 *The drop algorithm is a 2-approximation algorithm.*

Proof: Exercise 12.13. □

Experiments reported in [594] indicate that the drop algorithm performs much better than this guarantee and furthermore, already running the drop algorithm with a fixed root is better than running the branchings algorithm from every possible root and taking the best solution found.

12.6 Directed Steiner Problems

In this section we discuss several directed analogues of the Steiner problem for undirected graphs (Section 9.9.2). There are at least three ways to generalize the problem to digraphs:

- Given a digraph $D = (V, A)$ with non-negative arc costs and a subset $X \subset V$ and a vertex $r \in V$ (possibly $r \in X$), we may ask for a minimum cost out-tree T_r^+ rooted at r so that this out-tree contains all vertices from X. This is the DIRECTED STEINER TREE PROBLEM.
- The DIRECTED STEINER PROBLEM is as follows. Given a directed multigraph $D = (V, A)$, possibly with a cost function on the arcs, and a subset S of its vertices; find a minimum (cost) subset A' of A such that $D' = (V, A')$ contains an (s, t)-path for every choice of $s, t \in S$. The vertices in S are called **terminals**.
- Observe that a solution to the graph Steiner problem[7] for an instance $[G, X]$ is a connected subgraph H of G with the minimum number of vertices which contains all vertices of X. A natural generalization to digraphs would be to ask for the minimum number of vertices in a strongly connected subdigraph H which contains all vertices of a given subset X of vertices in a digraph D. Generalizing this a bit further by considering real-valued vertex costs and asking for a minimum cost subdigraph containing all of X, we obtain the MINIMUM COST STRONG SUBDIGRAPH PROBLEM (see, e.g., [98]). We will return to this problem in Section 12.7.3.

[7] See Section 9.9.2.

All three problems are $\mathcal{NP}$-hard (Exercise 12.15).

If $|X| = k$, then we can obtain a trivial k-approximation algorithm for the directed Steiner tree problem, by taking for each $x \in X$ a shortest (r, x)-path in D. The following result by Charikar, Chekuri, Cheung, Dai, Goel, Guha and Li shows that one can obtain a significantly better approximation. Note that because the set covering problem reduces to the directed Steiner tree problem in an approximation preserving way (Exercise 12.16), it follows from a result of Feige [310] that we should not expect to find a polynomial approximation algorithm with a ratio better than $\log |X|$.

Theorem 12.6.1 *[198] There exists a family of algorithms that achieves an approximation ratio of $i(i-1)k^{\frac{1}{i}}$ in time $O(n^i k^{2i})$ for any fixed $i > 1$ where $k = |X|$. In particular, a polynomial $O(k^\epsilon)$-approximation algorithm exists for any $\epsilon > 0$.* □

In Exercise 12.14 the reader is asked to describe a polynomial algorithm for the directed Steiner problem in the case when $|S| = 2$. Feldman and Ruhl [311] proved that, for every fixed k, the directed Steiner problem with k terminals is solvable in polynomial time. In fact they proved that the following more general problem is polynomially solvable for every fixed p: Given a directed multigraph $D = (V, A)$ and p pairs $\{(s_1, t_1), \ldots, (s_p, t_p)\}$ of vertices; find a smallest set of arcs A' in A such that $D' = (V, A')$ contains an (s_i, t_i)-path for $i = 1, 2, \ldots, p$. Feldman and Ruhl also showed that the weighted version is still polynomial (provided p is fixed).

Bang-Jensen, Gutin and Yeo proved that for some generalizations of tournaments one can solve the directed Steiner problem in polynomial time. Among other results they proved the following:

Theorem 12.6.2 *[98] The directed Steiner problem is solvable in polynomial time for locally semicomplete digraphs and for extended semicomplete digraphs.* □

Conjecture 12.6.3 *[98] The directed Steiner problem is solvable in polynomial time for semicomplete multipartite digraphs.*

Note that this conjecture is still open even for semicomplete bipartite digraphs.

In real-life applications, such as telecommunications, one is often interested in serving only a subset of the customers from a given source and furthermore not all customers have the same demand. This gives rise to the following more general problem which is called THE DIRECTED STEINER PROBLEM WITH CONNECTIVITY CONSTRAINTS (DSCC) in [237]. Given a directed graph $D = (V, A)$ with weights on the arcs, a special vertex s and a number k_v associated with each vertex $v \in V - s$; find a minimum cost

subset $A' \subseteq A$ such that $D\langle A'\rangle$ contains k_v arc-disjoint (s,v)-paths for all $v \in V - s$. It follows from our remarks in Section 9.9.2 that this problem is $\mathcal{NP}$-complete, even if we only allow $k_v \in \{0,1\}$ for each $v \in V - s$. In [237] Dahl discusses a cutting plane approach to solving the DSCC problem.

In [694] Melkonian and Tardos consider the following common generalization of the directed Steiner tree problem and the MSSS problem.

Problem 12.6.4 *Given a digraph $D = (V, A)$ with arc costs and a function $f : 2^V \to \mathbb{Z}$ satisfying*

$$d^+_D(S) \geq f(S) \text{ for each } S \subseteq V, \tag{12.14}$$

find a minimum cost subdigraph D' of D so that (12.14) holds when we replace D by D'.

Taking $f(S) = 1$ for all proper subsets of V and $f(V) = f(\emptyset) = 0$ we obtain the MSSS problem and taking $f(S) = 1$ for all S such that $r \in S$ and $(V - S) \cap X \neq \emptyset$ and $f(S) = 0$, otherwise, we obtain the directed Steiner tree problem.

Melkonian and Tardos show in [694] that if the function f in Problem 12.6.4 is crossing supermodular[8], then every basic solution to the LP-relaxation of the problem (formulated as an integer program in the obvious way) contains a variable whose value is at least $\frac{1}{4}$. They use this to devise a 4-approximation algorithm for Problem 12.6.4 in the case of a crossing supermodular demand function f. The algorithm is based on the the following lemma and the idea of **iterative rounding** which we illustrate below.

Lemma 12.6.5 *Let $D' = (V, A')$ be a subdigraph of a digraph $D = (V, A)$ and let f be an integer-valued crossing supermodular function defined on all subsets of V. Then the function $h(S) = f(S) - d^+_{D'}(S)$ is also crossing supermodular.*

Proof: Exercise 12.19. □

This suggests the following simple algorithm. First find a basic solution to the LP-relaxation of Problem 12.6.4. Include all arcs whose value (via the corresponding variable in the LP) is at least $\frac{1}{4}$ in the solution. Denote by D' the digraph consisting of the arcs chosen so far and recursively solve Problem 12.6.4 for the digraph $H = D - A(D')$ and the function $f - d^+_{D'}$ until no positive demand remains (that is, the current function f has no positive value).

Conjecture 12.6.6 *[694] If f is crossing supermodular in Problem 12.6.4, then every basic solution to the LP-relaxation of the problem contains a variable whose value is at least $\frac{1}{2}$.*

If true, this would imply a 2-approximation for the problem using the same method as described above. For further work on Problem 12.6.4 see, e.g., the paper [695] by Melkonian and Tardos.

[8] This notion is defined in Section 11.8.

12.7 Miscellaneous Topics

12.7.1 The Directed Spanning Cactus Problem

A **directed cactus** is a strongly connected digraph in which each arc is contained in exactly one cycle.

Palbom [740] studied the complexity of various problems related to spanning directed cactii in digraphs. It is not difficult to check whether a given digraph is a cactus (Exercise 12.18), but deciding whether a digraph contains a spanning cactus is a hard problem (called THE DIRECTED SPANNING CACTUS PROBLEM).

Theorem 12.7.1 *[740] Deciding whether a given digraph has a spanning directed cactus is NP-complete.* □

Clearly this implies that the weighted case is hard. In fact it is hard even for weighted complete graphs with weights obeying the triangle inequality.

Theorem 12.7.2 *[740] Finding a spanning directed cactus of minimum weight in an asymmetric*[9] *weighted complete digraph, where the weights obey the triangle inequality, is polynomial time equivalent to finding the minimum TSP tour in the same digraph. The two problems also have the same hardness of approximation.* □

The directed spanning cactus problem is trivial for locally in-semicomplete digraphs and easy for path-mergeable digraphs (Exercise 12.17), but nontrivial already for extended semicomplete digraphs (see, e.g., Exercise 12.20).

Problem 12.7.3 *Determine the complexity of the spanning directed cactus problem for well-structured classes of digraphs such as quasi-transitive digraphs, extended semicomplete digraphs and semicomplete multipartite digraphs.*

12.7.2 An FTP Algorithm for the MSSS Problem

In view of the $\mathcal{NP}$-hardness of the MSSS problem, it makes sense to study the problem in the framework of parametrized complexity.

Since the hamiltonian cycle problem is $\mathcal{NP}$-complete we cannot hope to find an algorithm of complexity $O(f(k)n^c)$ for deciding whether a given digraph has a strong spanning subdigraph with at most $n + k$ arcs, unless $\mathcal{P} = \mathcal{NP}$. As we shall see below, the following reformulation of the MSSS problem allows us to consider fixed parameter tractability questions regarding the MSSS problem.

[9] This just means that the weight of arcs uv and vu may be different.

Problem 12.7.4 (MSSS(k)) *Given a strong digraph D on n vertices and a natural number $k \leq n-2$. Does D have a spanning strong subdigraph on at most $2n-2-k$ arcs?*

Bang-Jensen and Yeo proved that problem MSSS(k) is indeed fixed parameter tractable and showed how to either solve the problem, or obtain a problem kernel of size $(2k-1)^2$ in polynomial time when the underlying graph of D has no cut vertex.

Theorem 12.7.5 *[122] Let D be a strong digraph on $n \geq 3$ vertices with no cut vertex and let $k \leq n-2$ be a non-negative integer. In polynomial time in n we can either decide that (D,k) is a yes instance of MSSS(k) or find a strong digraph D_{ker} such that $|V(D_{ker})| \leq f(k) = (2k-1)^2$ and (D,k) is a yes instance of Problem 12.7.4 if and only if (D_{ker},k) is a yes instance of MSSS(k).* □

Corollary 12.7.6 *[122] MSSS(k) is fixed parameter tractable with respect to the parameter k given in the formulation of Problem 12.7.4.* □

12.7.3 Minimum Cost Strong Subdigraphs

In the directed Steiner problem the objective is to minimize the total number (cost) of the arcs in the solution. If instead of minimizing this we want to minimize the number of the vertices in a strong digraph containing a prescribed set X of vertices, then we obtain an instance of the minimum cost strong subdigraph problem[10] which we defined in Section 12.6.

Lemma 12.7.7 *[98] Let D be a digraph with real-valued costs on the vertices and let X be the set of vertices of negative cost. In time $O(m+n\log n)$ one can compute a cheapest cycle, which contains a fixed vertex $v \in V(D)$ and no vertices in $X-v$. In time $O(n(m+n\log n))$ one can find a cheapest cycle, which contains at most one vertex from X.*

Proof: Exercise 12.21. □

We will also use the following lemma.

Lemma 12.7.8 *[98] Let D be a digraph with real-valued costs on its vertices such that D is either locally semicomplete or extended semicomplete. Let X denote the negative cost vertices in D. If $X \neq \emptyset$ and $D\langle X\rangle$ is connected, then we can find a minimum cost strong subdigraph in D, in $O(m+n\log n)$ time.* □

Theorem 12.7.9 *[98] Let D be a strong extended semicomplete digraph with real-valued costs on the vertices. In time $O(m+n\log n)$ one can find a minimum cost strong subdigraph of D.*

[10] Give vertices in X cost -1 and all other vertices cost 1.

Proof: Let X denote the set of vertices of negative cost in D. If $X = \emptyset$, then the cheapest strong subdigraph of D is a minimum cost vertex in D. Hence, assume that X is non-empty. If $D\langle X\rangle$ is connected, the claim follows from Lemma 12.7.8. Thus, we may assume that $D\langle X\rangle$ is not connected. Then all vertices in X are similar (have exactly the same in- and out-neighbours). Let x be a minimum cost vertex from X and let C be a minimum cost cycle among those which contain x and no other vertex from X.

As all vertices from X are similar, $D\langle V(C) \cup X\rangle$ is strong. Now it is easy to see that the minimum cost strong subdigraph of D is either $D\langle V(C) \cup X\rangle$ or the vertex x, whichever has the smaller cost. By Lemma 12.7.7, we can find C in time $O(m + n \log n)$. □

Bang-Jensen, Gutin and Yeo conjecture the following extension of Theorem 12.7.9.

Conjecture 12.7.10 *[98] The minimum cost strong subdigraph problem is solvable in polynomial time for semicomplete multipartite digraphs.*

Theorem 12.7.11 *A minimum cost strong subdigraph of a locally semicomplete digraph D can be found in time $O(nm + n^2 \log n)$.*

Proof: Since D is locally semicomplete, every strong subdigraph of D with at least two vertices contains a spanning cycle by Theorem 6.3.1. Hence a minimum cost strong subdigraph D' of D can be found in time $O(nm + n^2 \log n)$ by finding a minimum cost cycle (see Theorem 6.8.2) and a minimum cost vertex of D and taking the cheapest of these two as D'. □

12.8 Exercises

12.1. Complete the proof of Proposition 12.1.2. Hint: consider the proof of Theorem 5.3.2.

12.2. Convert the proof of Theorem 12.1.3 into a polynomial algorithm for finding the desired digraph.

12.3. Prove that the tournaments $\mathcal{T}_{n,k}$ described in Figure 12.1 are k-arc-strong for all $k \geq 1$ and $n \geq 5k + 2$.

12.4. Prove that if a path P in an extended semicomplete digraph D contains two vertices from an independent set I of D, then there exists a path P' and a cycle C' in D with $V(P) = V(P') \cup V(C')$.

12.5. (+) Prove Corollary 12.2.3. Hint: the proof of Theorem 12.2.2 is algorithmic. Identify the subroutines needed to do the different steps. See also Exercise 4.67.

12.6. Show that the proof of Theorem 12.2.6 can be turned into an $O(n^4)$ algorithm for finding a minimum strong spanning subdigraph of a quasi-transitive digraph.

12.7. (+) Prove Lemma 12.2.5. Hint: consider the way we argued in the proof of Proposition 12.1.2.

12.8. Prove the complexity part of Theorem 12.3.2.

12.9. **Finding subgraphs with specified bounds on degrees**. Describe a polynomial algorithm which takes as input a digraph $D = (V, A)$ on n vertices and non-negative integers $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n$ such that $d^+_D(v_i) \geq a_i$ and $d^-_D(v_i) \geq b_i$ for $i = 1, 2, \ldots, n$ and returns a minimum cardinality subset A' of A such that the digraph $D' = (V, A')$ satisfies that $d^+_{D'}(v_i) \geq a_i$ and $d^-_{D'}(v_i) \geq b_i$ for $i = 1, 2, \ldots, n$. Hint: use flows and consider a network similar to that used in the proof of Theorem 4.11.6.

12.10. Prove that if a digraph $D = (V, A)$ contains no closed anti-directed trail, then $|A| \leq 2|V| - 1$. Hint: consider the bipartite representation $BG(D)$ of D and show that this has no cycle.

12.11. (+) Show that for every p with $1 < p < 2$ there exists a weighted digraph $D = D(p)$ for which the weight of $D' = (V, A(B^-_r) \cup A(B^+_r))$, where B^-_r (B^+_r) is a minimum cost in-branching (out-branching) rooted at r in D is at least p times the weight of a minimum cost strong spanning subdigraph of D.

12.12. Prove Lemma 12.4.11. Hint: use flows to find the set A_r.

12.13. Prove Lemma 12.5.2. Hint: consider the arcs $\{a_1, a_2, \ldots, a_p\}$ whose values were dropped to zero. Then the cost of the returned solution is at most the original cost of these arcs.

12.14. (++) Describe a polynomial algorithm for finding, in a given digraph $D = (V, A)$ with specified vertices s, t, a minimum size subset $A' \subseteq A$ such that $D' = (V, A')$ has s, t in the same strong component (Natu and Fang [722]).

12.15. Prove that each of the problems listed in the beginning of Section 12.6 is $\mathcal{NP}$-hard.

12.16. Describe an approximation preserving polynomial reduction from the set covering problem to the directed Steiner tree problem.

12.17. Prove that every strongly connected path-mergeable digraph has a spanning directed cactus.

12.18. **Recognizing cactii in polynomial time.** Describe a polynomial algorithm to decide whether a given digraph is a directed cactus.

12.19. Prove Lemma 12.6.5.

12.20. (+) Construct a strong extended semicomplete digraph D which has a spanning cactus, but no such cactus contains a longest cycle of D.

12.21. Prove Lemma 12.7.7. Hint: Show how to use Dijkstra's algorithm with Fibonacci heaps [232] to solve the problem.

13. Packings, Coverings and Decompositions

In this chapter we study problems dealing with packing subdigraphs into a digraph, covering the vertices of a digraph by certain structures or decomposing a digraph into pieces with given properties. Many results fall into this very general description and we can only cover some representative ones.

Among the topics on packing that we cover are: packing directed cuts (the Lucchesi-Younger theorem), packing dijoins (Woodall's conjecture), packing hamiltonian paths and cycles. On the border between packings and coverings we deal with path/cycle factors and then we move on to covering the vertices of a digraph by (not necessarily disjoint) cycles. We give a proof of Gallai's conjecture, saying that the vertices of every strong digraph D can be covered by a set of at most $\alpha(D)$ cycles. Finally, in the part on decompositions, we study both decompositions of the arc set (arc-disjoint strong spanning subdigraphs, hamiltonian cycles) and of the vertex set (decompositions into digraphs with a prescribed bound on the longest path, into digraphs of given minimum out-degree, etc.).

13.1 Packing Directed Cuts: The Lucchesi-Younger Theorem

In this section we consider directed multigraphs. Let $D = (V, A)$ be a directed multigraph which is connected, but not strongly connected. A **directed cut** (or just a **dicut**) in D is a set of arcs of the form $(X, V - X)$, where X is a non-empty proper subset of V such that there are no arcs from $V - X$ to X (i.e., $(V - X, X)$ is a one-way pair with $h(V - X, X) = 0$). Note that two dicuts $(X, V - X)$ and $(Y, V - Y)$ may be arc-disjoint but still $X \cap Y \neq \emptyset$. As an example consider the second power of a directed path $x_1x_2 \dots x_{2k+1}$. Here $\{(\{x_1, \dots, x_{2i-1}\}, \{x_{2i}, \dots, x_{2k+1}\}) : 1 \leq i \leq k\}$ is a family of k arc-disjoint cuts (the first and last having two arcs and all other having three arcs). Clearly these cuts overlap considerably when we consider their vertex sets. For simplicity we will sometimes denote a dicut $(X, V - X)$ just by the set X.

A **dijoin** is a subset $A' \subset A$ which covers all dicuts. Let $\Omega(D)$ denote the maximum number of arc-disjoint dicuts in D and let $\epsilon(D)$ be the minimum

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 505
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_13,

cardinality of a dijoin. In the example above, when $k = 2r + 1$, the arcs $\{x_{2i-1}x_{2i+1} : i \in [r]\}$ form a minimum dijoin.

Suppose $D = (V, A)$ is connected but not strongly connected. Then it is clear that we can obtain a strong directed multigraph by contracting certain arcs. It is also clear that if we contract an arc a which is not an arc of a dicut $(X, V - X)$, then in the resulting directed multigraph $D' = (V', A')$, the corresponding pair $(X', V' - X')$ is still a dicut. On the other hand, if A' is a dijoin and we contract all arcs of A', then the resulting directed multigraph is strong. Let $\theta(D)$ denote the minimum number of arcs whose contraction in D leads to a strong directed multigraph. Then it follows from the discussion above that

$$\Omega(D) \leq \theta(D) \leq \epsilon(D). \tag{13.1}$$

The number $a_1(D)$ is the minimum number of new arcs one has to add to D in order to obtain a strongly connected directed multigraph (see Section 14.3). Note that if D is a directed (x, y)-path on r vertices, then $a_1(D) = 1$, since we may add a new arc yx and get a strong digraph. However, in order to obtain a strong directed multigraph by contracting arcs, we must contract $r - 1$ arcs, showing that $\theta(D) = r - 1$. This proves that $\theta(D)$ and $a_1(D)$ may be arbitrarily far apart.

Let D be a directed multigraph. Recall that the operation of subdividing an arc consists of replacing the arc xy in question by the path xuy of length two, where u is a new vertex. If several arcs are subdivided, then all the new vertices (used to subdivide these arcs) are distinct.

Lemma 13.1.1 *Let $D = (V, A)$ be a directed multigraph and let D' be obtained from D by subdividing each arc once. If D has k arc-disjoint dicuts, then D' has $2k$ arc-disjoint dicuts.*

Proof: Let $D' = (V', A')$ be obtained from D by subdividing each arc once. Let $X_1, \ldots, X_k$ be chosen such that the dicuts $(X_1, V - X_1), \ldots, (X_k, V - X_k)$ are arc-disjoint in D. For each dicut $(X_i, V - X_i)$ we denote by X_i' the set we obtain in D' by taking the union of X_i and the new vertices that subdivide the arcs leaving X_i. Now it is easy to see that the dicuts $(X_1, V' - X_1), (X_1', V' - X_1'), \ldots, (X_k, V' - X_k), (X_k', V' - X_k')$ are pairwise arc-disjoint. □

The next theorem, due to Lucchesi and Younger, shows that in fact equality holds everywhere in (13.1).

Theorem 13.1.2 (Lucchesi-Younger theorem) *[661] Let $D = (V, A)$ be a directed multigraph which is connected and either D has just one vertex, or it is not strongly connected. Then $\Omega(D) = \epsilon(D)$.*

Proof: We give a proof due to Lovász [654]. The proof is by induction on the number of arcs in A. If $A = \emptyset$, then D has precisely one vertex and there are no dicuts. Hence the statement of the theorem is vacuously true.

Now let $a \in A$ be an arbitrary arc. Contract a and consider the resulting directed multigraph D/a. Note that the dicuts of D/a are exactly those in D which do not contain the arc a. By induction, $\epsilon(D/a) = \Omega(D/a)$. Hence if $\Omega(D/a) \leq \Omega(D)-1$, then we can cover all dicuts in D by $\epsilon(D/a)+1 \leq \Omega(D)$ arcs and the theorem is proved. Thus we may assume that

$$\Omega(D/a) = \Omega(D) \text{ for every arc } a \in A. \tag{13.2}$$

By Lemma 13.1.1, if we subdivide all arcs in A, then the resulting digraph has at least $\Omega(D) + 1$ arc-disjoint dicuts (with equality only if $\Omega(D) = 1$). Hence, starting from D and subdividing arbitrary (not previously subdivided) arcs, we will get a sequence of directed multigraphs $D_0 = D, D_1, \ldots, D_h$, where $\Omega(D_i) = \Omega(D)$ for each $i \leq h-1$ and $\Omega(D_h) = \Omega(D)+1$. Let f be the last arc we subdivided in this process and let $H = D_{h-1}$. Now H contains $\Omega(D) + 1$ dicuts $X_1, X_2, \ldots, X_{\Omega(D)+1}$ such that only two of them have an arc in common and that arc is f.

Observe that H/f arises from G/f by subdivision. Hence, by the assumption (13.2), $\Omega(H/f) = \Omega(D)$ and so H contains $\Omega(D)$ arc-disjoint dicuts $Y_1, Y_2, \ldots, Y_{\Omega(D)}$ none of which contains the arc f. This implies that $X_1, X_2, \ldots, X_{\Omega(D)+1}, Y_1, Y_2, \ldots, Y_{\Omega(D)}$ is a collection of $2\Omega(D)+1$ dicuts in H such that no arc belongs to more than two of these. Thus the following lemma will give us a contradiction, implying that (13.2) cannot hold and hence the theorem follows.

Lemma 13.1.3 *If a digraph D contains at most k arc-disjoint dicuts, and $\mathcal{C}$ is any collection of dicuts in D such that no arc belongs to more than two dicuts in $\mathcal{C}$, then $|\mathcal{C}| \leq 2k$.*

Proof of Lemma 13.1.3: The proof below illustrates a powerful proof technique, known as the **uncrossing technique**. Call two dicuts $(X, V-X), (Y, V-Y)$ **crossing** if X and Y are crossing as sets. The first step is to uncross crossing dicuts in the family.

It follows from (5.2) that if $(X, V-X), (Y, V-Y)$ are crossing dicuts, then each of $(X \cup Y, V-(X \cup Y)), (X \cap Y, V-(X \cap Y))$ is a dicut and $d(X, Y) = 0$. Furthermore, the dicuts $(X \cup Y, V-(X \cup Y))$ and $(X \cap Y, V-(X \cap Y))$ cover each arc of D the same number of times as the dicuts $(X, V-X)$, $(Y, V-Y)$ (here we used that $d(X, Y) = 0$). Let $\mathcal{C}' = \mathcal{C} - \{(X, V-X), (Y, V-Y)\} + \{(X \cup Y, V-(X \cup Y)), (X \cap Y, V-(X \cap Y))\}$. Then $\mathcal{C}'$ has the same property as $\mathcal{C}$ that no arc covers more than two dicuts in $\mathcal{C}$ and furthermore we have

$$\sum_{(X, V-X) \in \mathcal{C}} |X|^2 \leq \sum_{(Z, V-Z) \in \mathcal{C}'} |Z|^2, \tag{13.3}$$

because $|X \cup Y|^2 + |X \cap Y|^2 > |X|^2 + |Y|^2$ when X, Y cross. Hence, if we replace crossing dicuts pairwise as we did above, then we will eventually reach a new family $\mathcal{C}^*$ of size $|\mathcal{C}|$ such that the dicuts in $\mathcal{C}^*$ are pairwise non-crossing

and no arc of D belongs to more than two dicuts in $\mathcal{C}^*$. Hence it suffices to prove that $\mathcal{C}^*$ contains at most $2k$ dicuts.

Let $\mathcal{C}^* = \{Z_1, Z_2, \ldots, Z_M\}$ and let $A_i = (Z_i, V - Z_i)$, $i = 1, 2, \ldots, M$, be the corresponding arc sets. Construct an undirected graph $G(\mathcal{C}^*) = (V, E)$ as follows: $V = \{v_1, v_2, \ldots, v_M\}$ and there is an edge between v_i and v_j if and only if $A_i \cap A_j \neq \emptyset$. Since D contains at most k arc-disjoint dicuts, it follows that $G(\mathcal{C}^*)$ has at most k independent vertices. Hence it suffices to show that $G(\mathcal{C}^*)$ is a bipartite graph, since then we get $|\mathcal{C}| = |\mathcal{C}^*| \leq 2k$.

Let $v'_1 v'_2 \ldots v'_s v'_1$ be an arbitrary cycle in $G(\mathcal{C}^*)$. Note that the arc sets of the corresponding dicuts $A'_1, \ldots, A'_s$ must be different, since if $(Z'_i, V - Z'_i) = (Z'_j, V - Z'_j)$ for some $1 \leq i < j \leq s$, then every arc in $(Z'_i, V - Z'_i)$ is covered twice (by $(Z'_i, V - Z'_i)$ and by $(Z'_j, V - Z'_j)$) and hence the vertices v'_i, v'_j each have degree one in $G(\mathcal{C}^*)$, contradicting the fact that they are on a cycle. Note also that if two dicuts $(X, V - X)$ and $(Y, V - Y)$ have $X \cup Y = V$, then they are arc-disjoint and hence are not adjacent in $G(\mathcal{C}^*)$.

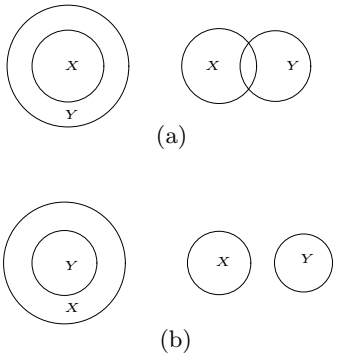


Figure 13.1 Illustration of the definition of being to the right and left for cuts. In the two situations in part (a) (part (b)) the dicut $(X, V - X)$ is to the left (right) of the dicut $(Y, V - Y)$. In the right part of (a) we have $X \cup Y = V$.

Since $A'_i \cap A'_{i+1} \neq \emptyset$ for $i = 0, 1, \ldots, s-1$, where $A'_0 = A'_s$, it follows from our remarks above that we have either $Z'_i \subset Z'_{i+1}$ or $Z'_{i+1} \subset Z'_i$. We prove that the two possibilities occur alternately and hence s is even. Suppose not, then without loss of generality we have $Z'_0 \subset Z'_1 \subset Z'_2$. Let us say that a dicut A'_i is **to the left of** another dicut A'_j if either $Z'_i \subset Z'_j$ or $Z'_i \cup Z'_j = V$ (which is equivalent to $V - Z'_i \subset Z'_j$) and that A'_i is **to the right of** A'_j if $Z'_i \cap Z'_j = \emptyset$ (which is equivalent to $Z'_i \subset V - Z'_j$) or $Z'_j \subset Z'_i$ (which is equivalent to $V - Z'_i \subset V - Z'_j$). See Figure 13.1. Since $\mathcal{C}^*$ contains no crossing members, each $A'_i \neq A'_j$ is either to the right or to the left of A'_j. Since A'_2 is to the right of A'_1 and $A'_0 = A'_s$ is to the left of A'_1, it follows that there is some $2 \leq j \leq s-1$ such that A'_j is to the right of A'_1 and A'_{j+1} is to the left of A'_1. Suppose first that $Z'_j \cap Z'_1 = \emptyset$, then we cannot have $Z'_{j+1} \subset Z'_1$ as

A'_{j+1} and A'_j have a common arc. So we must have $Z'_1 \cup Z'_{j+1} = V$, but then any arc a common to A'_j and A'_{j+1} enters Z'_1, contradicting that $d^-(Z'_1) = 0$. Hence we must have $Z'_1 \subset Z'_j$. The fact that A'_j, A'_{j+1} have a common arc a (and hence either $Z'_j \subset Z'_{j+1}$ or $Z'_{j+1} \subset Z'_j$) implies that, by the choice of j, we have $Z'_{j+1} \subset Z'_1 \subset Z'_{j+1}$. But now the arc a belongs to three dicuts A'_1, A'_j and A'_{j+1}, a contradiction. This completes the proof of the lemma and, by the remark above, also the proof of the theorem. □

Combining (13.1) and Theorem 13.1.2, we obtain:

Corollary 13.1.4 *Let D be a non-strong directed multigraph whose underlying graph is connected. Then $\theta(D) = \epsilon(D)$, that is, D can be made strongly connected by contracting $\epsilon(D)$ arcs.* □

The proof of Theorem 13.1.2 is not constructive but using submodular flows one can find a minimum dijoin $A' \subseteq A$ of D in polynomial time. As another illustration of the power of submodular flows, we show below how the Lucchesi-Younger theorem (Theorem 13.1.2) can be proved using a formulation of the problem as a minimum cost submodular flow problem and the duality theorem for linear programming. This application of submodular flows was first pointed out by Edmonds and Giles [286].

We wish to find a minimum set of arcs which cover all directed cuts in D. Let $x : A \to \{0, 1\}$ and let us interpret the value of $x(a)$ as follows. If $x(a) = 1$, then we choose a to be in the cover and otherwise (if $x(a) = 0$) a is not chosen. Since the set of chosen arcs must cover all directed cuts, we have the requirement

$$x^-(W) \geq 1 \qquad \text{for all } \emptyset \neq W \subset V \text{ such that } d^+_D(W) = 0. \tag{13.4}$$

Let $\mathcal{F} = \{W : d^+_D(W) = 0\}$. Then $\emptyset, V \in \mathcal{F}$ and (13.4) is equivalent to

$$x^+(W) - x^-(W) \leq b(W) \qquad \text{for all } W \in \mathcal{F}, \tag{13.5}$$

where $b(\emptyset) = b(V) = 0$ and $b(W) = -1$ for all $W \in \mathcal{F}' = \mathcal{F} - \{\emptyset, V\}$.

By our remark on different formulations of submodular flow problems, we see that this (having the form of (11.28)) is indeed a submodular flow formulation. Hence, by assigning cost one to each arc, we can formulate the problem of finding an optimal cover of the directed cuts as the following minimum cost submodular flow problem (in the form of (11.28)):

$$\begin{aligned} \mathcal{LY}: \text{ Minimize } & \sum_{a \in A} x(a) \\ \text{subject to } & \\ & x^-(W) \geq 1 \qquad \text{for all } W \in \mathcal{F}' \\ & 0 \leq x(a) \leq 1 \qquad \text{for all } a \in A. \end{aligned}$$

Taking dual variables y_W for each member W of $\mathcal{F}'$ and $\xi(a)$ for each arc $a \in A$, we get that the dual of $\mathcal{LY}$ is

$$\begin{aligned}
\mathcal{LY}^* : \text{Maximize} \quad & \sum_{W \in \mathcal{F}'} y_W - \sum_{a \in A} \xi(a) \\
\text{subject to} \quad & \\
& -\xi(a) + \sum_{a \in (\overline{W}, W)} y_W \leq 1 \qquad \text{for all } a \in A \\
& y_W \geq 0 \qquad \text{for all } W \in \mathcal{F}' \\
& \xi(a) \geq 0 \qquad \text{for all } a \in A.
\end{aligned}$$

Eliminate the variables $\xi(a)$ from $\mathcal{LY}^*$ and notice that, if $y_W = 0$ for all members $W \in \mathcal{F}$ which are entered by a, then the optimal choice for $\xi(a)$ is $\xi(a) = 0$. We get that $\mathcal{LY}^*$ is equivalent to the problem

$$\mathcal{LY}^{**} : \text{Maximize} \quad \sum_{W \in \mathcal{F}'} y_W + \sum_{a \in A} \min\{0, [1 - \sum_{a \in (\overline{W}, W)} y_W]\} \tag{13.6}$$

subject to

$$y_W \geq 0 \qquad \text{for all } W \in \mathcal{F}'. \tag{13.7}$$

By the Edmonds-Giles theorem, there exists an integer-valued optimum solution $\{y_W : w \in \mathcal{F}\} \cup \{\xi(a) : a \in A\}$ to $\mathcal{LY}^*$ and hence to $\mathcal{L}^{**}$. Notice that if some variable y_W in such a solution is 2 or more, then we can decrease its value to 1 without changing the value of the objective function in (13.6). Hence there exists an optimal solution to $\mathcal{LY}^{**}$ in which all values are 0 or 1. It follows from the optimality of the solution that if $y_W = y_{W'} = 1$, then we can assume that no arc enters both of W, W' (otherwise we may put $y_{W'} = 0$ without changing the value of the objective function). This shows that the cuts corresponding to the non-zero values of y are arc-disjoint and hence we have shown that the size of an optimal cover equals the maximum number of arc-disjoint directed cuts, which is exactly the statement of Theorem 13.1.2. Furthermore, by Theorem 11.8.5, we obtain the following corollary:

Corollary 13.1.5 *There exists a polynomial algorithm which given a directed multigraph $D = (V, A)$ finds a minimum dijoin $A' \subseteq A$ of D.* □

Note that we can minimize the function b from (13.5) over a given collection of sets in polynomial time (using flows). Namely, the minimum value is -1 if the collection contains a member of $\mathcal{F}'$ and 0 otherwise.

It follows from the formulation of the minimum directed cut covering problem as a submodular flow problem and Theorem 11.8.5 that we can also

solve the minimum cost version of the problem, even if there are non-uniform costs on the arcs and we want to find a minimum cost cover of the directed cuts. Furthermore, we can also solve the problem of finding a set of arcs which cover each directed cut at least k times for each k (simply replace the number -1 by $-k$ in (13.5)).

13.2 Packing Dijoins: Woodall's Conjecture

Woodall [909, 910] posed the following challenging conjecture which interchanges the role of directed cuts and dijoins in the Lucchesi-Younger theorem.

Conjecture 13.2.1 (Woodall's conjecture) *In a digraph, the minimum size of a directed cut is equal to the maximum number of disjoint dijoins.*

The following observation is attributed to Frank in [804].

Proposition 13.2.2 *If D is a directed multigraph such that the minimum size of a directed cut is at least two, then D contains two disjoint dijoins.*

Proof: Since $UG(D)$ is 2-edge-connected it has a strong orientation $D_s = (V, A_s)$ by Theorem 1.6.1. Let $X_1 \subset A$ be the set of arcs in D which are also arcs in D_s and let $X_2 = A - X_1$. Then it is easy to see that X_1, X_2 are the desired dijoins. □

A digraph is **source-sink connected** if it contains a directed path from each initial component to each terminal component. Schrijver [797] and Feofiloff and Younger [314] proved that Woodall's conjecture is true for source-sink connected digraphs.

Theorem 13.2.3 *[314, 797] In a source-sink connected digraph, the minimum size of a directed cut is equal to the maximum number of dijoins.* □

For planar digraphs, by duality, Conjecture 13.2.1 is equivalent to the following conjecture by Woodall.

Conjecture 13.2.4 *[909] In a planar digraph D the length of a shortest cycle is equal to the maximum number of arc-disjoint feedback arc sets of D.*

This conjecture does not hold when we replace planar digraphs by tournaments as shown by the following example due to Thomassen (see, e.g., [268]): It is easy to construct a strong tournament X on 5 vertices such that every feedback arc set of X has at least three arcs (Exercise 13.1). Now let T be the tournament on 15 vertices obtained by taking three copies X_1, X_2, X_3 of X and adding arcs between them so that between X_i and X_{i+1}, $i = 1, 2, 3$, we have the arcs of a matching of size five in the direction towards X_{i+1} and

all the remaining 20 arcs go from X_{i+1} to X_i. It can be shown by a careful analysis (Exercise 13.2) that one must delete at least 39 (which is more than $\frac{1}{3}|A(T)|$) arcs in T in order to obtain an acyclic digraph. Thus T cannot contain 3 disjoint feedback arc sets.

Using a directed analogue of the Szemerédi regularity lemma (see, e.g., [268]), Donadelli and Kohayakawa [268] showed that, in fact, one can obtain an infinite family of sparse oriented graphs which have many directed cycles but no two arc-disjoint feedback arc sets.

Theorem 13.2.5 *[268] Let an integer $\ell \geq 3$ and a real number $\beta > 0$ be given. For any sufficiently large n, there exists an oriented graph $D_n = (V, A)$ with $O(n^{1+\frac{1}{(\ell-1)}})$ arcs and girth ℓ such that any subset $A' \subset A$ with $|A'| \geq (\frac{1}{2} + \beta)|A|$ induces a digraph with at least one cycle of length ℓ.* □

This result is best possible in the sense that every digraph contains an acyclic subdigraph with at least half of its arcs (just take a random ordering of its vertices and keep all arcs pointing forward or all arcs pointing backward).

Lee and Wakabayashi proved that Conjecture 13.2.4 is true if we replace planar digraphs by series-parallel digraphs.

Theorem 13.2.6 *[638] In a series-parallel digraph the length of a shortest cycle is equal to the maximum number of disjoint feedback arc sets of D.* □

They showed that for series-parallel digraphs, even the extension of Theorem 13.2.6 where we have weights $c(a)$ on the arcs and each arc is allowed to be in at most $c(a)$ of the feedback arc sets holds[1].

In [817] Shepherd and Vetta survey results related to Woodall's conjecture and the Lucchesi-Younger theorem, see also [804, Chapter 56].

13.3 Packing Cycles

For a digraph D, the maximum number of vertex-disjoint (arc-disjoint) cycles is denoted by $\nu_0(D)$ ($\nu_1(D)$).

Proposition 13.3.1 *For every digraph D there exist digraphs D' and D'' such that $\nu_0(D) = \nu_1(D')$ and $\nu_1(D) = \nu_0(D'')$. The digraphs D' and D'' can be constructed from D in polynomial time.*

Proof: The digraph D'' can be defined as $D'' := L(D)$. To construct D' simply apply the vertex splitting procedure (see Subsection 4.2.4) to all vertices of D. The reader is advised to verify that the equalities of this proposition indeed hold. □

Theorem 13.3.2 *Given a digraph D and an integer k, it is $\mathcal{NP}$-complete to decide whether $\nu_0(D) \geq k$ ($\nu_1(D) \geq k$).*

[1] Theorem 13.2.6 is the case when $c = 1$ for all arcs.

Proof: By Proposition 13.3.1 it is sufficient to show this claim only for ν_0. A scheme of the proof of the assertion for ν_0 is given in Exercise 13.25. □

It turns out that one of the sufficient conditions to guarantee the existence of a large number of vertex-disjoint cycles in a digraph D is that $\delta^+(D)$ is large enough. Let $f(k)$ be the least integer such that every digraph of minimum out-degree at least $f(k)$ contains k vertex-disjoint cycles. The very existence of $f(k)$ for every $k \geq 1$ is not obvious. Thomassen [858] was the first to prove this fact. He proved that $f(k) \leq (k+1)!$.

Bermond and Thomassen made the following conjecture which would be best possible (see, e.g., Exercise 13.3).

Conjecture 13.3.3 *[152] Every digraph with minimum out-degree at least $2k-1$ has k disjoint cycles (that is, $f(k) = 2k-1$).*

The conjecture holds for $k = 1$, as every acyclic digraph has a vertex of out-degree zero. Thomassen proved that the conjecture also holds for $k = 2$ (see Exercises 13.3 and 13.4). Lichiardopol, Pór and Sereni [644] have verified the conjecture for $k = 3$. Alon [17] was the first to prove that the function $f(k)$ is linear. He obtained the following result.

Theorem 13.3.4 *There exists an absolute constant C so that $f(k) \leq Ck$ for all k. In particular, $C = 64$ will do.* □

We will not give a proof of Theorem 13.3.4 as it is somewhat tedious. However, we will prove a slightly weaker result, Theorem 13.3.7. This proof shows basic ideas involved in the proof of Theorem 13.3.4 in [17]. We leave as Exercise 13.5 the proof of the following corollary.

Corollary 13.3.5 *[17] Every digraph with minimum out-degree k has at least $k^2/128$ arc-disjoint cycles.* □

For k-regular digraphs, the result of this corollary seems far from being sharp. Alon, McDiarmid and Molloy [27] conjectured the following:

Conjecture 13.3.6 *Every k-regular digraph contains $\binom{k+1}{2}$ arc-disjoint cycles.*

This conjecture was verified for $k \leq 3$ in [27]. Now we formulate Theorem 13.3.7.

Theorem 13.3.7 *[17] For k large enough, $f(k) \leq (3 + o(1))k \log_e k$.*

Proof: For technical reasons, we prove this theorem not only for digraphs, but for directed pseudographs without parallel arcs. However, for simplicity we will still use the term 'digraphs' in the rest of this subsection for digraphs with possible loops.

Clearly, Theorem 13.3.7 holds for $k = 1$. Assume that Theorem 13.3.7 is true for all values up to some k and $k+1$ is the minimum integer violating the inequality. Then, $f(k+1) > f(k) + 4$. Let $D = (V, A)$ be a digraph of minimum out-degree r, $r = f(k+1)-1$, such that D does not have $k+1$ vertex-disjoint cycles. We also assume that D has the minimum possible number of vertices and, subject to this property, the minimum size. By the definition of D, the out-degree of every vertex of D is exactly r and $\delta^-(D) > 0$. Moreover, D has no loop, since otherwise the digraph obtained from D by deleting a vertex with a loop cannot contain k vertex-disjoint cycles, showing that $f(k+1) - 2 = r - 1 \leq f(k) - 1$, which is impossible as we saw above that $f(k+1) > f(k) + 4$.

We proceed by proving certain properties of D formulated as lemmas. The proof of Lemma 13.3.10 exploits a probabilistic argument. The first lemma is due to Thomassen [858] and the next two to Alon [17].

Lemma 13.3.8 *[858] For every $v \in V$, the subdigraph $D\langle N^-(v)\rangle$ contains a cycle.*

Proof: Fix an arbitrary vertex $v \in V$. Put $H = D\langle N^-(v)\rangle$. It suffices to show that $\delta^-(H) > 0$. Assume that $u \in V(H)$ and $d^-_H(u) = 0$. Then, there is no vertex in D that dominates both u and v. This implies that the digraph D', obtained from D by first deleting the arcs with tail u except for uv and then contracting uv, has minimum out-degree r. (Notice that D' may have a loop.) By the minimality of D, the digraph D' has $k+1$ vertex-disjoint cycles. These cycles can easily be transformed into vertex-disjoint cycles of D, a contradiction. □

Lemma 13.3.9 *[17] We have $|V| \leq k(r^2 - r + 1)$.*

Proof: Put $n = |V|$ and let G be the undirected graph with vertex set V in which a pair u and v of distinct vertices is adjacent if and only if there is a vertex in D that dominates both. Define $m = n\binom{r}{2}$ and observe that the size of G is at most m (since every vertex of D has out-degree r). Therefore, as is well known (see, e.g., Berge [143, page 282]) G has an independent set of cardinality at least $\frac{n^2}{2m+n}$. If this number is at least $k+1$, then there is a set $x_1, \ldots, x_{k+1}$ of independent vertices of G. This means that the sets $N^-(x_1), \ldots, N^-(x_{k+1})$ are pairwise disjoint. It now follows from Lemma 13.3.8 that D has $k+1$ vertex-disjoint cycles, a contradiction. Hence, $\frac{n^2}{2m+n} \leq k$. This implies the inequality of Lemma 13.3.9. □

Lemma 13.3.10 *[17] We have $k(r^2 - r + 2)(1 - \frac{1}{k+1})^r \geq 1$.*

Proof: Assume that the inequality of this lemma is false and

$$k(r^2 - r + 2)(1 - \frac{1}{k+1})^r < 1.$$

Assign independently to every vertex $v \in V$ a colour $i \in [k+1]$ with probability $p = \frac{1}{k+1}$. Let V_i be the set of vertices coloured i. For each vertex $v \in V$, let E_v denote the event that all out-neighbours of v are of colours different than that of v. Since every vertex of D has out-degree r we have $\text{Prob}(E_v) = (1-p)^r$. For each $i \in [k+1]$, let F_i denote the event that $V_i = \emptyset$. Then $\text{Prob}(F_i) = (1-p)^n \leq (1-p)^{r+1}$. Hence, by Lemma 13.3.9,

$$\begin{aligned}\sum_{v \in V} \text{Prob}(E_v) + \sum_{i=1}^{k+1} \text{Prob}(F_i) &\leq n(1-p)^r + (k+1)(1-p)^{r+1} \\ &\leq k(r^2 - r + 1)(1-p)^r + k(1-p)^r \\ &= k(r^2 - r + 2)(1-p)^r \\ &< 1.\end{aligned}$$

This implies that with positive probability each $D\langle V_i \rangle$ is non-empty and has a positive minimum out-degree, and hence possesses a cycle. Thus, there is a choice of $V_1, \ldots, V_{k+1}$ giving $k+1$ disjoint cycles in D, a contradiction. □

Conclusion of the proof of Theorem 13.3.7: Lemma 13.3.10 implies that

$$k(r^2 - r + 2) \geq e^{r/(k+1)}.$$

Hence, for k large enough, $f(k) \leq f(k+1) - 1 = r \leq (3 + o(1))k \log_e k$. Thus, Theorem 13.3.7 is proved. □

13.4 Arc-Disjoint Hamiltonian Paths and Cycles

From Euler's theorem (Theorem 1.7.2) one easily derives the following result attributed to Veblen in [152] (see also Exercise 13.28).

Theorem 13.4.1 *The arcs of a digraph can be partitioned into cycles if and only if, for each vertex x, we have $d^+(x) = d^-(x)$.* □

The proof of the following strengthening of Theorem 13.4.1 for regular digraphs by Kotzig is left to the reader as Exercise 13.6.

Theorem 13.4.2 *[625] If D is a regular digraph, then the arc set of D can be partitioned into cycle factors.* □

We now consider decompositions of the arc set of a digraph into hamiltonian cycles. Deciding whether such a decomposition exists for an arbitrary digraph is an extremely hard problem. Even for complete digraphs this is non-trivial. It is an old result due to Walecki (see [36]) that the edge set of the complete undirected graph K_n has a decomposition into hamiltonian cycles if and only if n is odd (if n is even, then each vertex has odd degree

and no decomposition can exist). Using this result we easily conclude that the arc set of $\overleftrightarrow{K}_n$ can be decomposed into hamiltonian cycles when n is odd. However, for even n another approach is needed by the remark above.

It is easy to check that the arcs of $\overleftrightarrow{K}_4$ cannot be decomposed into hamiltonian cycles. Indeed, without loss of generality, the first cycle in such a decomposition is 12341 where the vertices of $\overleftrightarrow{K}_4$ are labelled 1,2,3,4. After removing these arcs one obtains a strong semicomplete digraph with a unique hamiltonian cycle 14321 and hence the desired decomposition cannot exist. With a little more effort one can also prove that the arc set of $\overleftrightarrow{K}_6$ cannot be decomposed into five hamiltonian cycles (Exercise 13.29). On the other hand, Tillson proved that for all other values of n such a decomposition does indeed exist.

Theorem 13.4.3 (Tillson's decomposition theorem) *[874] The arcs of $\overleftrightarrow{K}_n$ can be decomposed into hamiltonian cycles if and only if $n \neq 4, 6$.* □

Theorem 13.4.3 will be used in Section 13.10. Answering a question of Alspach, Bermond and Sotteau, Ng [725] extended Theorem 13.4.3 to the following:

Theorem 13.4.4 *[725] The arcs of $\overleftrightarrow{K}_{r,r,\ldots,r}$ (s times) can be decomposed into hamiltonian cycles if and only if $(r, s) \neq (4, 1)$ and $(r, s) \neq (6, 1)$.* □

The following conjecture, due to Kelly (see [704]), is probably one of the most famous conjectures in tournament theory:

Conjecture 13.4.5 (Kelly's conjecture) *The arcs of a regular tournament of order n can be partitioned into $(n-1)/2$ hamiltonian cycles.*

This conjecture was verified for $n \leq 9$ by Alspach [152, page 28]. Jackson [555] proved that every regular tournament of order at least 5 contains a hamiltonian cycle C and a hamiltonian path arc-disjoint from C. Zhang proved in [928] that there are always two arc-disjoint hamiltonian cycles for $n \geq 5$. A digraph D is **almost regular** if $\Delta^0(D) - \delta^0(D) \leq 1$. Thomassen [857] proved the following:

Theorem 13.4.6 *[857] Every regular or almost regular tournament of order n has at least $\lfloor \sqrt{n/1000} \rfloor$ arc-disjoint hamiltonian cycles.* □

This result was improved by Häggkvist to the following:

Theorem 13.4.7 *[487] There is a positive constant c (in fact, $c \geq 2^{-18}$) such that every regular tournament of order n contains at least cn arc-disjoint hamiltonian cycles.* □

For n sufficiently large, the following result due to Keevash, Kühn and Osthus follows directly from Theorem 6.5.1.

Theorem 13.4.8 *[587] There exists an integer k_0 such that every k-regular tournament with $k \geq k_0$ has at least $\frac{2k+1}{8}$ arc-disjoint hamiltonian cycles.* □

Thomassen [861] proved that the arcs of every regular tournament of order n can be covered by $12n$ hamiltonian cycles. So far the Kelly conjecture remains unsettled and it remains a serious challenge to find a proof of this longstanding and very interesting conjecture.

For further results on decompositions into hamiltonian cycles we refer the reader to the paper [36] by Alspach, Bermond and Sotteau and the paper [726] by Ng.

Let T be the tournament on $n = 4m+2$ vertices obtained from two regular tournaments T_1 and T_2, each on $2m+1$ vertices, by adding all arcs from the vertices of T_1 to T_2 (i.e., $V(T_1) \mapsto V(T_2)$ in T). Clearly T is not strong and so has no hamiltonian cycle. The minimum in-degree and minimum out-degree of T is m which is about $\frac{n}{4}$. Bollobás and Häggkvist [165] showed that if we increase the minimum in- and out-degree slightly, then, not only do we obtain many arc-disjoint hamiltonian cycles, we also obtain a very structured set of such cycles provided that the tournament has enough vertices.

Theorem 13.4.9 *[165] For every $\epsilon > 0$ and every natural number k there is a natural number $n(\epsilon, k)$ with the following property. If T is a tournament of order $n > n(\epsilon, k)$ such that $\delta^0(T) \geq (\frac{1}{4} + \epsilon)n$, then T contains the kth power of a hamiltonian cycle.* □

It is easy to prove that every tournament on n vertices with minimum in- and out-degree at least $\frac{n}{4}$ is strongly connected (see Exercise 1.12).

We now turn our attention to other results concerning arc-disjoint hamiltonian paths and cycles in tournaments. Thomassen [857] completely characterized tournaments having at least two arc-disjoint hamiltonian paths. A tournament is **almost transitive** if it is obtained from a transitive tournament with acyclic ordering $u_1, u_2, \ldots, u_n$ (i.e., $u_i \rightarrow u_j$ for all $1 \leq i < j \leq n$) by reversing the arc u_1u_n. Let T be a non-strong tournament with the acyclic ordering $T_1, T_2, \ldots, T_k$ of its strong components. Two components T_i, T_{i+1} are called **consecutive** for $i = 1, 2, \ldots, k-1$.

Theorem 13.4.10 *[857] A tournament T fails to have two arc-disjoint hamiltonian paths if and only if T has a strong component which is an almost transitive tournament of odd order or T has two consecutive strong components of order 1.* □

Deciding whether a given tournament T has a hamiltonian path P and a hamiltonian cycle C such that P and C are arc-disjoint seems to be a difficult problem. Thomassen found the following partial solution involving arc-3-cyclic tournaments:

Theorem 13.4.11 *[857] Let T be an arc-3-cyclic tournament of order at least 3. Then T has a hamiltonian path P and a hamiltonian cycle arc-disjoint from P, unless T is a 3-cycle or the tournament of order 5 obtained from a 3-cycle by adding two vertices x, y and the arc xy and letting y (respectively x) dominate (respectively be dominated by) the vertices of the 3-cycle.* □

It is easy to see that regular tournaments are arc-3-cyclic (Exercise 13.19). Hence Theorem 13.4.11 generalizes the result of Jackson above. But Theorem 13.4.11 goes much further since, as we mentioned in Section 8.1.6, almost all tournaments satisfy the assumption of the theorem (see [704]). The following conjecture, in some sense generalizing Kelly's conjecture, was proposed by Thomassen:

Conjecture 13.4.12 *[857] For any $\epsilon > 0$ almost all tournaments of order n have $\lfloor (\frac{1}{2} - \epsilon)n \rfloor$ arc-disjoint hamiltonian cycles.*

Erdős (see [857]) raised the following problem:

Problem 13.4.13 *Do almost all tournaments have $\delta^0(T)$ arc-disjoint hamiltonian cycles?*

As we mentioned in the beginning of Section 7.4 there is no degree condition which guarantees that a strong tournament contains two arc-disjoint hamiltonian cycles. In fact, one can easily show that even high arc-strong connectivity does not exclude the existence of one arc which is in all hamiltonian cycles (see Exercise 7.19). Thomassen posed the following conjecture.

Conjecture 13.4.14 *[857] For each integer $k \geq 2$ there exists an integer $h(k)$ such that every $h(k)$-strong tournament has k arc-disjoint hamiltonian cycles.*

Thomassen [857] showed by an example that $h(2) > 2$ and conjectured that $h(2) = 3$. His example also shows that h is not bounded by any linear function. See also Figure 13.4 for an example of a 2-strong tournament without two arc-disjoint hamiltonian cycles.

By Theorem 7.4.7, Conjecture 13.4.14 would follow for $k = 2$ from the following conjecture due to Bang-Jensen and Yeo ($h(2) = 3$ would hold). Note that in Figure 13.4 it suffices to remove two arcs to destroy all hamiltonian cycles.

Conjecture 13.4.15 *[120] Every tournament T either contains two arc-disjoint hamiltonian cycles or a set A' of at most two arcs such that $T - A'$ has no hamiltonian cycle.*

13.5 Path Factors

Recall that the path covering number of a digraph is the minimum number of disjoint paths needed to cover $V(D)$.

The following attainable lower bound for the path covering number of a digraph D is quite trivial: $\mathrm{pcc}(D) \leq \mathrm{pc}(D)$. Clearly $\mathrm{pcc}(D) = \mathrm{pc}(D)$ for acyclic digraphs and we saw in Theorem 6.6.2 that this also holds for semicomplete multipartite digraphs D. The aim of this section is to prove theorems by Gallai and Milgram, respectively, Dilworth which provide bounds on the path covering number in terms of the independence number. These bounds are useful in several applications (see, e.g., Section 13.9).

Lemma 13.5.1 *Let $D = (V, A)$ be a digraph and let $\hat{D}$ be obtained from D by adding a new vertex s and all possible arcs from s to V. Then* $\mathrm{pc}(D) = \ell_{\min}(\hat{D})$.

Proof: Exercise 13.7. □

Recall that the independence number $\alpha(D)$ of a digraph D is the cardinality of a maximum independent set of vertices of D. Rédei's theorem (Theorem 1.4.2) can be rephrased as saying that every digraph with independence number one has a hamiltonian path and hence path covering number equal one. Gallai and Milgram generalized this as follows.

Theorem 13.5.2 (Gallai-Milgram theorem) *[388] For every digraph D,* $\mathrm{pc}(D) \leq \alpha(D)$.

Proof: Combine Lemma 13.5.1 and Theorem 9.7.1. □

The following theorem due to Erdős and Szekeres [300] follows easily from Theorem 13.5.2.

Theorem 13.5.3 *Let n, p, q be positive integers with $n > pq$, and let $I = (i_1, i_2, \ldots, i_n)$ be a sequence of n distinct integers. Then there exists either a decreasing subsequence of I with more than p integers or an increasing subsequence of I with more than q integers.*

Proof: Let $D = (V, A)$ be the digraph with $V = \{i_1, i_2, \ldots, i_n\}$ and $A = \{i_m i_k : \ m < k \text{ and } i_m < i_k\}$. Observe the obvious correspondence between independent sets of D and decreasing subsequences of I (respectively paths of D and increasing subsequences of I). Let $\mathcal{F} = P_1 \cup \ldots \cup P_s$ be a minimum path factor of D. By Theorem 13.5.2, $s \leq \alpha(D)$. Hence, $\alpha(D) \cdot \max_{j=1}^{s} |P_j| \geq n > pq$. Thus, either $\alpha(D) > p$, i.e., there exists a decreasing subsequence with $\alpha(D) > p$ integers, or $\max_{j=1}^{s} |P_j| > q$, i.e., there exists an increasing subsequence with more than q integers. □

The well-known Chvátal-Erdős sufficient condition for hamiltonicity of undirected graphs states that if G is a k-connected undirected graph and

$\alpha(G) \leq k$, then G is hamiltonian [221]. This does not extend to digraphs as shown in [558] (see also Exercise 13.10 and Section 13.8 below). Clearly, if a digraph $D = (V, A)$ has a hamiltonian cycle, then for every choice of distinct vertices $v_1, v_2, \ldots, v_r \in V$, D has an r-path factor $P_1, \ldots, P_r$ such that v_i is the initial vertex of P_i, $i = 1, 2, \ldots, r$. Inspired by this observation, Bessy [156] posed the following conjecture.

Conjecture 13.5.4 *[156] Let D be a digraph with $\alpha(D) \leq \kappa(D)$. Then, for every choice of $\alpha = \alpha(D)$ vertices $\{x_1, x_2, \ldots, x_\alpha\}$, there exists an α-path factor $P_1, P_2, \ldots, P_\alpha$ of D such that x_i is the initial vertex of P_i, $i = 1, 2, \ldots, \alpha$.*

Bessy showed by an example that if $\alpha(D) > \kappa(D)$, then the desired paths may not exist. He also verified Conjecture 13.5.4 for digraphs with $\alpha \leq 2$ (by the remark above, the case $\alpha = 1$ follows from Camion's theorem).

Theorem 13.5.5 *[156] Let $D = (V, A)$ be a 2-strong digraph with $\alpha(D) \leq 2$. Then for every choice of distinct vertices $x, y \in V$, D has a 2-path factor P_x, P_y so that P_x starts in x and P_y starts in y.* □

For acyclic digraphs it turns out that the minimum path factor problem can be solved quite efficiently. This is important since this problem has many practical applications. One such example is as follows.

A news agency wishes to cover a set of events $E_1, E_2, \ldots, E_n$ each of which takes place within the coming week starting at a prescribed time T_i, $i = 1, 2, \ldots, n$. For each event E_i its duration time t_i and geographical site O_i are known. The news agency wishes to cover each of these events by having one reporter present for the full duration of the event. At the same time it wishes to use as few reporters as possible. Assuming that the travel time t_{ij} from O_i to O_j is known for each $1 \leq i, j \leq n$, we can model this problem as follows. Form a digraph $D = (V, A)$ by letting $V = \{v_1, v_2, \ldots, v_n\}$ and for every choice of $i \neq j$ put an arc from v_i to v_j if $T_j \geq T_i + t_i + t_{ij}$. It is easy to see that D is acyclic. Furthermore, if the events can be covered by k reporters, then D has a k-path factor (just follow the routes travelled by the reporters). It is also easy to see that the converse holds. Hence having an algorithm for the minimum path factor problem for acyclic digraphs will provide a solution to this and a large number of similar problems (such as airline and tanker scheduling, see Exercise 13.8).

Using flows in networks, we can effectively find a $\mathrm{pcc}(D)$-path-cycle factor in any digraph D (see Exercises 4.7 and 4.67). Since a k-path-cycle factor in an acyclic digraph has no cycles, this implies that the minimum path factor problem for acyclic digraphs is easy (at least from an algorithmic point of view).

Theorem 13.5.6 *For acyclic digraphs the minimum path factor problem is solvable in time $O(\sqrt{n}m)$.* □

The following is a direct corollary of Theorem 9.7.3 and Lemma 13.5.1.

Theorem 13.5.7 *Every acyclic transitive digraph D has* $\mathrm{pc}(D) = \alpha(D)$. □

Another application of the path covering number of acyclic digraphs is for partial orders. A **partial order** consists of a set X and a binary relation '$\prec$' which is transitive (that is, $x \prec y, y \prec z$ implies $x \prec z$). Let $P = (X, \prec)$ be a partial order. Two elements $x, y \in X$ are **comparable** if either $x \prec y$ or $y \prec x$ holds. Otherwise x and y are **incomparable**. A **chain** in P is a totally ordered subset Y of X, that is, all elements in Y are pairwise comparable. An **antichain** on P is a subset Z of X, no two elements of which are comparable. Dilworth proved the following well-known min-max result relating chains to antichains:

Theorem 13.5.8 (Dilworth's theorem) *[260] Let $P = (X, \prec)$ be a partial order. Then the minimum number of chains needed to cover X equals the maximum number of elements in an antichain.*

Proof: Given $P = (X, \prec)$, let $D = (X, A)$ be the digraph such that $xy \in A$ for $x \neq y \in X$ if and only if $x \prec y$. Clearly, D is transitive. Furthermore, a path (an independent set) in D corresponds to a chain (antichain) in P. Thus the claim follows from Theorem 13.5.7. □

We mentioned in Section 9.7.1 that Dilworth's theorem is equivalent to Theorem 9.7.3 and saw above how to get Dilworth's theorem from Theorem 9.7.3. To see the other direction, suppose that D is transitive and has a unique source s. Then every out-branching is rooted in s and we have $pc(D) \leq \ell_{min}(D)$. By Theorem 9.7.1, $\ell_{min}(D) \leq \alpha(D)$ and by Dilworth's theorem, $\mathrm{pc}(D) = \alpha(D)$. Hence $\ell_{min}(D) = \alpha(D)$.

We conclude this section with a simple result on extended semicomplete digraphs. Lemma 13.5.9 is used in Section 12.2.

Lemma 13.5.9 *Let D be an acyclic extended semicomplete digraph with $\alpha(D) = k$, then the following holds:*

(a) $\mathrm{pc}(D) = k$.
(b) One can obtain a minimum path factor of D as follows: choose a longest path P in D, remove $V(P)$ and continue recursively.

Proof: Exercise 13.27. □

13.6 Cycle Factors with the Minimum Number of Cycles

The MINIMUM CYCLE FACTOR PROBLEM is as follows: Given a directed graph D with a cycle factor; find a cycle factor of D with the minimum number

of cycles among all cycle factors of D. The problem is clearly $\mathcal{NP}$-hard for general digraphs as the answer is one if and only if D is hamiltonian.

The minimum cycle factor problem is easy for extended semicomplete digraphs and semicomplete bipartite digraphs (Exercise 13.15) but seems very difficult for general semicomplete multipartite digraphs. Below we consider the minimum cycle factor problem for quasi-transitive digraphs. The exposition is based on results by Bang-Jensen and Nielsen [113]. We start with some new notation.

Definition 13.6.1 *For every digraph, D, with at least one cycle and every non-negative integer, i, let $\eta_i(D) = \min\{j \mid D \text{ has a } j\text{-path-}i\text{-cycle factor}\}$.*

Thus $\eta_0(D) = pc(D)$ and $\eta_i(D) = 0$ if and only if D has an i-cycle factor, so for general digraphs the computation of $\eta_i(D)$ is $\mathcal{NP}$-hard already for $i = 0, 1$. By Theorems 6.7.5 and 6.7.4, calculating $\eta_0(D)$ and $\eta_1(D)$ can be done in polynomial time for quasi-transitive digraphs.

We also make use of the following technical definition. At the end of this section we will also supply some motivation for the definition.

Definition 13.6.2 *Let $\mathcal{F} = C_1 \cup \ldots \cup C_q$ be a q-cycle factor of a digraph D. We say that $\mathcal{F}$ is* **reducible** *if there exists a q'-cycle factor $\mathcal{F}' = C'_1 \cup \ldots \cup C'_{q'}$ of D such that each of the following holds:*

(a) $q' < q$;
(b) for every $i \in [q]$ there is a $j \in [q']$ such that $V(C_i) \subseteq V(C'_j)$.

Such an $\mathcal{F}'$ is called a **reduction of** *$\mathcal{F}$. If no reduction of $\mathcal{F}$ exists, then $\mathcal{F}$ is said to be* **irreducible**.

It is clear that every minimum cycle factor is irreducible. In the following D will always denote a quasi-transitive digraph with canonical decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$, where S is either a strong semicomplete digraph or a transitive oriented digraph, depending on whether D is strong or not. Furthermore, those cycles of a cycle factor $\mathcal{F}$ that are contained in a Q_i are called **small** cycles and all other cycles of $\mathcal{F}$ are called **large** cycles.

Lemma 13.6.3 *Let $\mathcal{F}$ be an irreducible cycle factor in D. Then the following holds:*

(a) If D is non-strong, then $\mathcal{F}$ has no large cycle.
(b) If D is strong, then $\mathcal{F}$ has precisely one large cycle.

Proof: Exercise 13.11. □

Corollary 13.6.4 *Every minimum cycle factor in a strong quasi-transitive digraph contains exactly one large cycle.* □

For a strong quasi-transitive digraph $D = S[Q_i, \ldots, Q_s]$ we denote by $D_0 = S[\overline{K}_{n_1}, \ldots, \overline{K}_{n_s}]$ the strong extended semicomplete digraph we obtain from D by deleting all arcs from each Q_i, $i \in [s]$. Let $\mathcal{C}$ be the set of all cycle subdigraphs of D_0 and let $m_i(D) = \max_{S \in \mathcal{C}}\{|V(S) \cap V(\overline{K}_{n_i})|\}$, for all $i \in [s]$. By Theorem 6.6.8, every longest cycle in D_0 contains exactly $m_i(D)$ vertices of H_i.

Lemma 13.6.5 *[113] If $D = S[Q_i, \ldots, Q_s]$ is a strong quasi-transitive digraph containing a cycle factor, then D has a minimum cycle factor in which the (unique) large cycle C intersects Q_i in exactly $m_i(D)$ paths for each $i \in [s]$. That is, by contracting each maximal subpath of C which lies inside Q_i (for every $i \in [s]$), we obtain a longest cycle of D_0.*

Proof: Exercise 13.12. □

A **canonical minimum cycle factor** of a quasi-transitive digraph D is one for which the unique large cycle intersects each Q_i in exactly $m_i(D)$ paths. By Lemma 13.6.5, every strong quasi-transitive digraph with a cycle factor has a canonical minimum cycle factor.

Let $I(D) = \{i \mid m_i(D) < pc(Q_i)\}$ and note that for every $i \in [s]$, Q_i has the same number c_i of small cycles with respect to every canonical minimum cycle factor (for $i \notin I(D)$ this number is zero). Since, for every $t \geq 0$ and every digraph H, we have $\eta_{t+1}(H) \geq \eta_t(H) - 1$, we see that, for $i \in I(D)$, we have $c_i = \min\{j \mid \eta_j(Q_i) = m_i(D)\}$. Hence, we have the following characterization of the number, $k_{\min}(D)$, of cycles in a minimum cycle factor:

Theorem 13.6.6 *[113] For every strong quasi-transitive digraph, D, containing a cycle factor, we have*

$$k_{\min}(D) = 1 + \sum_{i \in I(D)} \min\{j \mid \eta_j(Q_i) = m_i(D)\}.$$

Furthermore, every cycle factor of D has at least $1 + \sum_{i \in I(D)} (pc(Q_i) - m_i(D))$ cycles. □

It follows from Theorem 13.6.6 that we could determine $k_{\min}(D)$ in polynomial time, provided we could calculate $\eta_j(Q_i)$, for every $i \in [r]$ and every $j \in \{0, 1, \ldots, \lfloor \frac{|V(Q_i)|}{2} \rfloor\}$, in polynomial time. However, it is unlikely the approach used in [113] (using network flows) can be extended to the general case ($j \geq 3$).

Problem 13.6.7 *Does there exist a polynomial algorithm which for a given quasi-transitive digraph D finds the numbers $\eta_j(D)$, $j = 0, 1, \ldots, \lfloor \frac{|V(D)|}{2} \rfloor$, and a corresponding path-cycle factor whenever $\eta_j(D) < \infty$?*

Problem 13.6.8 *[113] Determine the complexity of computing $k_{\min}(D)$ and finding a minimum cycle factor of a quasi-transitive digraph D.*

When $k_{\min}(D)$ is small, in particular when $k_{\min}(D) \in \{1, 2, 3\}$, the problem is polynomially solvable: the case $k_{\min}(D) = 1$ is the hamiltonian cycle problem, which is polynomial by Theorem 6.7.1, and the cases $k_{\min}(D) = 2, 3$ are covered by the next theorem.

Theorem 13.6.9 *[113] For $k \in \{2, 3\}$ there exist polynomial algorithms to verify whether a quasi-transitive digraph has a cycle factor with at most k cycles.* □

Conjecture 13.6.10 *[113] For each fixed k there is a polynomial algorithm which determines whether a given quasi-transitive digraph D has a cycle factor with at most k cycles and, if so, finds a minimum cycle factor of D.*

By Theorem 13.6.9, the conjecture holds for $k \leq 3$ and the techniques used in the paper can be modified to work for slightly higher k values, but do not extend to arbitrary (fixed) values of k.

Let us return to the notion of an irreducible cycle-factor. We have noted that every minimum cycle factor is irreducible and Bang-Jensen and Nielsen proved that irreducible cycle factors can be found efficiently.

Theorem 13.6.11 *[113] There is an $\mathcal{O}(n^5)$ algorithm which, given a cycle factor $\mathcal{F}$ in a quasi-transitive digraph $D = R[H_1, H_2, \ldots, H_r]$, either confirms that $\mathcal{F}$ is irreducible or returns a reduction of $\mathcal{F}$. Hence in time $\mathcal{O}(n^6)$ any given cycle factor can be converted into an irreducible one.* □

It is easy to see that an irreducible cycle factor $\mathcal{F}$ with $c \leq 2$ cycles in a strong quasi-transitive digraph D is also minimum: if $c = 2$, every Hamilton cycle in D would be a reduction of $\mathcal{F}$. It is thus natural at this point to ask whether every irreducible cycle factor in a strong quasi-transitive digraph D is also minimum or, at least, close to being minimum. Unfortunately, this is not always the case as shown by an example in [113]. In fact, the example in [113] shows that $\mathcal{F}$ may be irreducible and have arbitrarily many cycles even though $k_{\min}(D) = 2$.

Solving (perhaps even partially) the following problem seems to be closely related to solving the minimum cycle factor problem for quasi-transitive digraphs.

Problem 13.6.12 *[113] Characterize those cycle factors of a quasi-transitive digraph which are irreducible but not minimum cycle factors.*

Problem 13.6.13 *What is the complexity of deciding whether a digraph has a cycle factor with at least two directed cycles?*

13.7 Cycle Factors with a Fixed Number of Cycles

Two cycles C, C' in a digraph $D = (V, A)$ are **complementary** if these cycles form a 2-cycle factor[2] in D.

Reid proved the case $k = 3$ in the following result (see also Exercise 13.30). Song extended this to arbitrary $3 \leq k \leq n - 3$.

Theorem 13.7.1 *[773, 829] Every 2-strong tournament on at least 8 vertices has a 2-cycle factor consisting of a k-cycle and an $(n - k)$-cycle for every $3 \leq k \leq n - 3$.* □

The theorem also holds for 2-strong tournaments on 6 vertices and the only exception on 7 vertices is a 3-regular tournament T_7 (see [773]). There are infinite families of tournaments T with $\kappa(T) = 1$ which do not have complementary cycles. One such example was given in [643] by Li and Shu. This example shows that the following result is best possible in terms of the lower bound of 3.

Theorem 13.7.2 *[643] Let T be a strong tournament on at least 6 vertices. If $\max\{\delta^-(T), \delta^+(T)\} \geq 3$ and T is not isomorphic to the tournament T_7, then T has a 2-cycle factor.* □

There are a number of results on 2-cycle factors in bipartite tournaments. One of these is the following due to Song:

Theorem 13.7.3 *[828] Let R be a bipartite tournament with $2k+1$ vertices in each partite set ($k \geq 4$). If every vertex of R has out-degree and in-degree at least k, then for any vertex x in R, R contains a 2-cycle factor $C \cup C'$ such that C includes x and the length of C is at most 6 unless R is isomorphic to $\vec{C}_4[\overline{K}_{k+1}, \overline{K}_{k+1}, \overline{K}_k, \overline{K}_k]$.* □

For further results on 2-cycle factors in semicomplete bipartite digraphs see, e.g., the paper [931] by Zhang and Wang and [930] by Zhang, Manoussakis and Song.

The problem of deciding the existence of a 2-cycle factor in semicomplete p-partite digraphs with $p \geq 3$ is quite difficult. Giving a partial answer to a conjecture by Yeo in [918], Volkmann proved the following for the case of regular multipartite tournaments.

Theorem 13.7.4 *[892] Let D be regular p-partite tournament D with $p \geq 4$ and $|V(D)| \geq 6$. Then D has a 2-cycle factor in which one of the cycles has length 3, unless D is isomorphic to one of two 3-regular 4-partite multipartite tournaments $D_{4,2}, D^*_{4,2}$ on 8 vertices or the 3-regular tournament T_7.* □

[2] This should not be confused with a cycle factor consisting of cycles of length two.

The two multipartite tournaments $D_{4,2}, D^*_{4,2}$ do have a 2-cycle factor but in every such 2-cycle factor both cycles have length 4. Volkmann also settled the case of regular 3-partite tournaments and his results imply the following. For a further result see [891].

Theorem 13.7.5 *[895, 897] Every regular multipartite tournament on at least 8 vertices has a 2-cycle factor.* □

The following conjecture has been proposed by Volkmann. Observe that for a semicomplete multipartite digraph D with p partite sets $V_1, V_2, \ldots, V_p$, the independence number $\alpha(D)$ is equal to the size of a largest set among the V_i's.

Conjecture 13.7.6 *[890] Let D be a p-partite tournament with partite sets $V_1, V_2, ..., V_p$ and let $\alpha = \alpha(D)$. If D is $(\alpha + 1)$-strong, then D has a 2-cycle factor, unless D is a member of a finite family of multipartite tournaments.*

In fact, Conjecture 13.7.6 is just one instance of the following meta-conjecture due to Volkmann (private communication, 1997): Several results which hold for k-strong tournaments should also hold for every semicomplete multipartite digraph D provided that D is $(\alpha(D) + k - 1)$-strong. One instance where this is known to be true is for the hamiltonian cycle problem (see Corollary 6.6.24).

An obvious necessary condition for a digraph D to contain a 2-cycle factor is that the girth of D is at most $n/2$. The second power $D = \vec{C}^2_{2k+1}$ of an odd cycle has girth $k + 1$ and D is a 2-strong locally semicomplete digraph. This shows that Theorem 13.7.1 cannot be extended to locally semicomplete digraphs. Confirming a conjecture by Bang-Jensen [69], Guo and Volkmann proved that powers of odd cycles are the only exceptions when $n \geq 8$.

Theorem 13.7.7 *[442] Let D be a 2-strong locally semicomplete digraph on $n \geq 8$ vertices. Then D has a 2-cycle factor such that both cycles have length at least 3 if and only if D is not the second power of an odd cycle.* □

Guo and Volkmann have shown that, although Theorem 13.7.1 cannot be extended to locally semicomplete digraphs, there is still enough structure to allow 2-cycle factors with many different lengths. We refer the reader to [443] for details.

Meierling and Volkmann [692] proved the following extension of Theorem 13.7.7 in the case of oriented graphs.

Theorem 13.7.8 *[692] Every 2-strong locally in-tournament digraph D on $n \geq 8$ vertices has a pair of complementary cycles if and only if D is not the second power of an odd cycle.* □

Bang-Jensen and Nielsen [112] gave a polynomial algorithm for checking whether a given locally semicomplete digraph has a 2-cycle factor. They also

showed in [113] that there is a polynomial algorithm for checking the existence of a 2-cycle factor in a quasi-transitive digraph.

The existence of a 2-cycle factor such that each cycle contains a prescribed vertex and has a prescribed length in a bipartite digraph has been studied in the papers [646, 899] by Little, Teo and Wang.

We now turn to cycle factors with more than two cycles. Bollobás (see [829]) posed the following problem:

Problem 13.7.9 *Let k be a positive integer. What is the least integer $g(k)$ so that all but a finite number of $g(k)$-strong tournaments contain a k-cycle factor?*

Chen, Gould and Li [203] answered this problem by proving that $g(k) \leq 3k^2+k$. In relation to Problem 13.7.9, Song made the following much stronger conjecture:

Conjecture 13.7.10 *[829] For any k integers $n_1, n_2, ..., n_k$ with $n_i \geq 3$ for $i = 1, 2, ..., k$ and $\sum_{i=1}^{k} n_i = n$, all but a finite number of k-strong tournaments on n vertices contain a k-cycle factor such that the k cycles have the lengths $n_1, n_2, ..., n_k$, respectively.*

If, instead of tournaments, we consider digraphs which are almost complete, then, by the following result, due to Amar and Raspaud, we may almost completely specify the lengths of the cycles in a cycle factor.

Theorem 13.7.11 *[40] Let D be a strong digraph on n vertices and at least $(n-1)(n-2)+3$ arcs. For every partition $n = n_1 + n_2 + \ldots + n_k$ such that $n_i \geq 3$ for all $i \in [k]$, D contains a k-cycle factor $C_1 \cup C_2 \cup \ldots \cup C_k$ such that C_i has length n_i for all $i \in [k]$ except in two cases:*

$n = 6$, $n_1 = n_2 = 3$ and $\alpha(D) = 3$, or
$n = 9$, $n_1 = n_2 = n_3 = 3$ and $\alpha(D) = 4$. □

Bessy posed the following problem.

Problem 13.7.12 *[156] Does there exist a function f such that every digraph D with $\alpha(D) \leq \kappa(D)$ has a cycle factor with at most $f(\alpha(D))$ cycles?*

By Camion's theorem $f(1) = 1$. It is not hard to show that $f(2) \geq 2$ (Exercise 13.10). An example in [558] also given in [156] shows that $f(3) \geq 2$ and Bessy also shows by an example in [156] that if we do not require $\alpha(D) \leq \kappa(D)$, then no function f exists.

13.8 Cycle Subdigraphs Covering Specified Vertices

In the solution of several algorithmic problems, such as finding the longest cycle in an extended semicomplete digraph or a semicomplete bipartite digraph, it is an important subproblem to find a cycle subdigraph which covers as many vertices as possible. Below we show how to solve this problem using an idea due to Alon (see [452]).

Theorem 13.8.1 *There is an $O(n^3)$ algorithm which finds, for any given digraph D, a cycle subdigraph covering the maximum number of vertices in D.*

Proof: Let D be a digraph and let D' be the directed pseudograph one obtains by adding a loop at every vertex. Let B be the weighted bipartite graph one obtains from the bipartite representation $BG(D')$ of D by adding the following weights to the edges: the weight of an edge $x'y''$ of B equals 1 if $x \neq y$ and equals 2 if $x = y$. It is easy to see (Exercise 13.13) that, by solving the assignment problem for B (in time $O(n^3)$, see Section 4.10.3) and then removing all the edges with weight 2 from the solution, we obtain a set of edges of B corresponding to some 1-regular subdigraph F of D of maximum order. □

Jackson and Ordaz [557] proved the following sufficient condition for the existence of a cycle factor in a digraph. (For undirected graphs this is the Chvátal-Erdős condition which we mentioned in Section 13.5.)

Proposition 13.8.2 *[557] If D is a k-strong digraph such that $\alpha(D) \leq k$, then D has a spanning cycle subdigraph.* □

We now prove a generalization of this result. Deciding whether there is a cycle containing all vertices from a prescribed set X in an arbitrary digraph is an $\mathcal{NP}$-complete problem already when $|X| = 2$ (see Theorems 10.2.1 and 10.2.5). Proposition 13.8.2 corresponds to the special case $X = V$ in the following theorem, due to Bang-Jensen, Gutin and Yeo.

Theorem 13.8.3 *[95] Let $D = (V, A)$ be a k-strong digraph and let $X \subset V(D)$ be such that $\alpha(D\langle X\rangle) \leq k$, then D has a cycle subdigraph[3] covering X.*

Proof: This can be proved directly from Theorem 4.8.2 (Exercise 13.16). We give a simple proof based on Proposition 4.11.7 which also holds for directed pseudographs (see Exercise 4.66).

Let D and X be as defined in the theorem. Form the directed pseudograph D' from D by adding a loop at each vertex not in X. Then D has a cycle subdigraph covering X if and only if D' has a cycle factor, because the new

[3] Note that the cycle subdigraph is not necessarily spanning.

arcs cannot contribute to cycles which cover vertices from X. Suppose D' has no cycle factor. Then, by Proposition 4.11.7(c), we can partition the vertices of V into sets R_1, R_2, Y, Z so that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z|$ and Y is an independent set. Note that no vertex with a loop can be in an independent set. Thus we have $Y \subseteq X$. It follows from the description of the arcs between the sets above that there is no path from Y to R_1 in $D-Z$. Thus we must have $|Z| \geq k$ since D is k-strong. But now we have the contradiction

$$k \leq |Z| < |Y| \leq \alpha(D\langle X \rangle) \leq k.$$

Thus D' has a cycle factor, implying that D has a cycle subdigraph covering X. □

Theorem 13.8.3 shows that the obvious necessary condition for the existence of a cycle covering a specified subset X, namely, that there exists some collection of disjoint cycles covering X, is satisfied in many cases. Indeed, if D is k-strong, then we may take X arbitrarily large, provided its independence number stays below $k+1$.

We point out that when $|X| = k$ and D is k-strong, then the existence of a cycle subdigraph covering X can also be proved easily using Menger's theorem (Theorem 5.4.1). See Exercise 5.15.

The proof above combined with that of Theorem 13.8.1 immediately implies the following result.

Theorem 13.8.4 *There exists an $O(n^3)$ algorithm for checking whether a given digraph $D = (V, A)$ with a prescribed subset $X \subseteq V$ has a cycle subdigraph covering X.* □

13.9 Proof of Gallai's Conjecture

Now we discuss the problem of covering the vertices of a digraph with a small number of not necessarily disjoint cycles. A collection $\mathcal{R}$ of cycles is **spanning** if every vertex in $V(D)$ is contained in at least one cycle of $\mathcal{R}$.

Since semicomplete digraphs have a rich structure, it is natural to believe that some of this structure is present in digraphs with small independence number, in particular for digraphs of independence number two.

Two cycles C, C' are **consistent** if they are either disjoint or their intersection is a subpath in both cycles. If D has a pair of consistent cycles C, C' which are spanning and not disjoint, then these along with all remaining arcs of D (not on C, C') form an ear decomposition with precisely two non-trivial ears. Clearly, the converse also holds. Bondy [168] gave a short proof of the following result by Chen and Manalastras.

Theorem 13.9.1 *[201] If D is strong and $\alpha(D) \leq 2$, then D is either hamiltonian or it has a pair of consistent cycles which is spanning.* □

Theorem 13.9.1 immediately implies the following result, which again implies Theorem 9.7.5 in the case $\alpha(D) = 2$:

Corollary 13.9.2 *[201] If D is strong and $\alpha(D) \leq 2$, then D is traceable.* □

It is tempting to ask whether one can generalize Corollary 13.9.2 to the statement that every k-strong digraph D with $\alpha(D) \leq k + 1$ is traceable. However, the example in Figure 13.2, due to Bondy [168], shows that such a generalization is not possible. We leave it to the reader to prove that D has no hamiltonian path (Exercise 13.31).

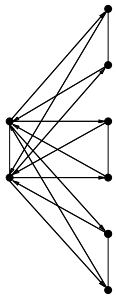


Figure 13.2 A 2-strong digraph D with $\alpha(D) = 3$ and no hamiltonian path. The vertical edges correspond to directed 2-cycles.

Note that if a digraph $D = (V, A)$ has a hamiltonian path, then $\mathrm{pc}(D - X) \leq |X| + 1$ for every $X \subset V$ (see also Proposition 1.4.3). In the digraph in Figure 13.2 we have $\mathrm{pc}(D - X) = 3 = |X| + 1$ when X consists of the two vertices to the left. Hence, the example in Figure 13.2 shows that the condition above is not always sufficient to guarantee a hamiltonian path in a digraph.

Gallai [386] conjectured that every strong digraph D has a spanning collection of $\alpha(D)$ not necessarily disjoint cycles. For $\alpha = 2$ the conjecture follows from Theorem 13.9.1. Bessy and Thomassé [158] gave an elegant proof of the full conjecture which we describe below.

Let $D = (V, A)$. Given an ordering $E = v_1, \ldots, v_n$ of V, we say that an arc $v_i v_j$ is **forward** if $i < j$ and **backward** if $j < i$. A directed path P of D is a **forward path** with respect to an ordering $v_1, v_2, \ldots, v_n$ if it does not contain any backward arc with respect to $v_1, v_2, \ldots, v_n$.

An ordering $E = v_1, \ldots, v_n$ is **elementary equivalent** to another ordering E' of V, if one of the following holds:

(i) $E' = v_n, v_1, \ldots, v_{n-1}$,
(ii) $E' = v_2, v_1, v_3, \ldots, v_n$ and neither $v_1 v_2$ nor $v_2 v_1$ is an arc of D.

Two orderings E, E' of V are **equivalent** if there is a sequence $E = E_1, \ldots, E_k = E'$ such that E_i and E_{i+1} are elementary equivalent, for $i = 1, \ldots, k-1$. The classes of this equivalence relation are called the **cyclic orders** of D. Roughly speaking, a cyclic order $\mathcal{O}$ is a class of orderings of the vertices on the integers modulo n, where one stays in the class while switching consecutive vertices which are not joined by an arc.

Given an ordering E, the **index** of a directed cycle C with respect to E, denoted $i_E(C)$, is the number of backward arcs of C with respect to E. This corresponds to the number of times the cycle winds around in the ordering. Observe that if E' is elementary equivalent to E, then $i_E(C) = i_{E'}(C)$. Consequently, the index of a cycle is invariant in a given cyclic order $\mathcal{O}$, we denote it by $i_{\mathcal{O}}(C)$. By extension, the index $i_{\mathcal{O}}(\mathcal{R})$ of a set of cycles $\mathcal{R}$ is the sum of the indices of the cycles of $\mathcal{R}$. A cycle is **simple** if it has index one. A cyclic order $\mathcal{O}$ is **coherent** if every arc of D is contained in a simple cycle.

The following important result has many consequences, e.g., Theorem 13.9.7. Note also that Camion's theorem (Theorem 1.5.2) is an easy consequence of Theorem 13.9.3 (see Exercise 13.32).

Theorem 13.9.3 *[158] Every strong digraph has a coherent cyclic order.*

Proof: Let us consider a cyclic order $\mathcal{O}$ which minimizes $i_{\mathcal{O}}(\mathcal{C})$, where $\mathcal{C}$ is the set of all cycles in D. We suppose for contradiction that $\mathcal{O}$ is not coherent. Then there exists an ordering $E = v_1, \ldots, v_n$ of $\mathcal{O}$ and a backward arc $a = v_j v_i$ which is not in a cycle of index 1 with respect to $\mathcal{O}$. Assume moreover that E and a are chosen in order to minimize $j - i$. Let k be the largest integer $i \leq k < j$ such that there exists a forward path from v_i to v_k. Observe that v_k has no out-neighbour in $\{v_p : k < p \leq j\}$. If $k \neq i$, by the minimality of $j - i$, v_k has no in-neighbour in $\{v_p : k < p \leq j\}$. In particular the ordering $E' = v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_j, v_k, v_{j+1}, \ldots, v_n$ is equivalent to E, and contradicts the minimality of $j-i$. Thus $k = i$, and by the minimality of $j - i$, there is no in-neighbour of v_i in $\{v_p : i < p < j\}$. In particular the ordering $E' = v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{j-1}, v_i, v_j, \ldots, v_n$ is equivalent to E. Observe now that in $E'' = v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{j-1}, v_j, v_i, v_{j+1}, \ldots, v_n$, every cycle C satisfies $i_{E''}(C) \leq i_{E'}(C)$, and the inequality is strict if the arc a belongs to C. Since D is strong, there exists a cycle which contains a, contradicting the choice of $\mathcal{O}$. □

Corollary 13.9.4 *[155] Given a strong digraph $D = (V, A)$ on n vertices and m arcs we can find a coherent cyclic ordering $\mathcal{O}$ of D in time $O(nm^2)$.*

Proof: For a given ordering $E = v_1, v_2, \ldots, v_n$ of V, we define the index, $i_E(a)$, of the arc $a \in A$ as the minimum index among all cycles of D which contain a. Furthermore, define the index $i_E(v)$ of each vertex $v \in V$ as the maximum index of an arc with tail v.

We show how to modify, in time $O(m)$, a given ordering $E = v_1, v_2, \ldots, v_n$ to a new ordering E' so that we have $i_{E'}(v_n) \leq i_E(v_n)$ and $i_{E'}(v_n) < i_E(v_n)$

if $i_E(v_n) > 1$ (note that v_n may not be the last vertex in E', see below) and all backward arcs with respect to E' are also backward with respect to E. This will imply the desired algorithm as, by the definition of a cyclic order, we may always reorder a given ordering so that a particular vertex w becomes the last one in the ordering. In particular, if the index of v_n is still larger than one in the new ordering, we may move it to the last position in a new equivalent order and repeat the step above. As $i_E(v) \leq n$ for every ordering, we need time at most $O(nm)$ to find an ordering in which v_n has index 1 and hence the total time will be $O(n^2m)$ as claimed.

Let D_E be the spanning subdigraph of D consisting of all arcs of A that are forward with respect to E and let $T^-_{v_n}$ be a maximum size in-tree rooted in v_n in D_E (can be found by an execution of DFS backwards from v in D_E). Observe that every arc v_nv_j, where $v_j \in V(T^-_{v_n})$, has $i_E(v_nv_j) = 1$. If every out-neighbour of v_n is in $V(T^-_{v_n})$, then $i_E(v_n) = 1$ and we return $E' = E$.

Otherwise, let i be the smallest index so that v_nv_i is an arc of D and $v_i \notin V(T^-_{v_n})$. Let E' be the ordering obtained from E by moving all vertices of $V(T^-_{v_n}) \cap \{v_{i+1}, v_{i+2}, \ldots, v_n\}$ forward so that all these vertices now occur in their original order between v_{i-1} and v_i.

Because v_i is not a vertex of $T^-_{v_n}$, no new backward arc is created. Hence, since all arcs from v_n to $\{v_1, \ldots, v_{i-1}\}$ have index one, we get $i_{E'}(v_n) < i_E(v_n)$ (all backward arcs leaving v_n either have index 1 or index one smaller in E' than in E). Clearly the complexity of the algorithm above is $O(m)$. □

An independent set X of D is **cyclic independent** with respect to $\mathcal{O}$ if there exists an ordering $v_1, \ldots, v_n$ of $\mathcal{O}$ such that $X = \{v_1, \ldots, v_k\}$. The **cyclic independence number**, denoted $\alpha(\mathcal{O})$, of a coherent cyclic order $\mathcal{O}$ is the maximum k such that D has a cyclic independent set X with respect to $\mathcal{O}$ such that $|X| = k$. Observe that $\alpha(\mathcal{O})$ depends on the choice of $\mathcal{O}$ (Exercise 13.33).

Lemma 13.9.5 *[158] Let D be a strong digraph and let $v_1, \ldots, v_n$ be an ordering of a coherent cyclic order $\mathcal{O}$ of D. Let X be a subset of vertices of D such that there is no forward path between any two distinct vertices of X. Then X is a cyclic independent set. In particular, $|X| \leq \alpha(\mathcal{O})$.*

Proof: We consider an ordering $E = v_1, \ldots, v_n$ of $\mathcal{O}$ such that there is no forward path between any two distinct vertices of X, and E is chosen in such a way that $j - i$ is minimum, where v_i is the first element of X in the ordering, and v_j is the last element of X in the ordering. Suppose for the sake of contradiction that $X \neq \{v_i, \ldots, v_j\}$. Then there exists $v_k \notin X$ for some $i < k < j$. There cannot exist both a forward path from $X \cap \{v_i, \ldots, v_{k-1}\}$ to v_k and a forward path from v_k to $X \cap \{v_{k+1}, \ldots, v_j\}$. Without loss of generality, we assume that there is no forward path from $X \cap \{v_i, \ldots, v_{k-1}\}$ to v_k. Suppose moreover that v_k is chosen with minimum index k. Clearly, v_k has no in-neighbour in $\{v_i, \ldots, v_{k-1}\}$, and since $\mathcal{O}$ is coherent, v_k has no out-neighbour in $\{v_i, \ldots, v_{k-1}\}$ (such an arc cannot belong to a cycle of index

one by the assumption that there is no forward path from $X \cap \{v_i, \ldots, v_{k-1}\}$ to v_k). Thus the ordering $v_1, \ldots, v_{i-1}, v_k, v_i, \ldots, v_{k-1}, v_{k+1}, \ldots, v_n$ belongs to $\mathcal{O}$, contradicting the minimality of $j-i$. Consequently, $X = \{v_i, \ldots, v_j\}$, and there is no forward arc, and (by the same argument as for v_k above) no backward arc between any two vertices of X. This shows that X is independent and considering now the ordering $v_i, \ldots, v_n, v_1, \ldots, v_{i-1}$, we conclude that X is a cyclic independent set. □

Theorem 13.9.6 *[158] Let D be a strong digraph and let $\mathcal{O}$ be a coherent cyclic order of $V(D)$. The minimum value of $i_{\mathcal{O}}(\mathcal{R})$, where $\mathcal{R}$ is a spanning collection of cycles of D, is equal to $\alpha(\mathcal{O})$.*

Proof: We consider a coherent cyclic order $\mathcal{O}$ of D with cyclic independence number $k = \alpha(\mathcal{O})$. Let $E = v_1, \ldots, v_n$ be an ordering of $\mathcal{O}$ such that $S = \{v_1, \ldots, v_k\}$ is an independent set of D. Clearly, if a cycle C contains q vertices of S, then the index of C is at least q. In particular the inequality $i_{\mathcal{O}}(\mathcal{R}) \geq k$ is satisfied for every spanning collection of cycles of D. To prove that equality holds, we consider an auxiliary digraph D' on vertex set $V \cup S'$ where $S' = \{v'_1, \ldots, v'_k\}$ whose arc set consists of every forward arc of E and every arc $v_i v'_j$ for which $v_i v_j$ is an arc of D and $v_j \in S$. Note that D' is acyclic and every vertex in S (S') has in-degree (out-degree) zero. Let T' be the transitive closure of D'. We will prove that the size of a maximum independent set in T' is exactly k. Consider such an independent set I, and let $I' = I \cap S'$.

Since one can arbitrarily permute the vertices of S in the ordering E and still remain in $\mathcal{O}$, we may assume that $I' = \{v'_1, \ldots, v'_j\}$ for some $0 \leq j \leq k$. Using that every vertex is in a simple cycle (as $\mathcal{O}$ is coherent), we conclude that there is a directed path in D' from v_i to v'_i, and consequently $v_i v'_i \in A(T')$, showing that we cannot have both $v_i \in I$ and $v'_i \in I$. Clearly, the ordering $E' = v_{j+1}, \ldots, v_n, v_1, \ldots, v_j$ belongs to $\mathcal{O}$. By the fact that I is an independent set in T', there is no forward path joining two elements of $I_D = (I \cap V) \cup \{v_1, \ldots, v_j\}$ in E' (I_D is the set corresponding to I back in D), and thus, by Lemma 13.9.5, $|I_D| \leq k$, implying that $|I| \leq k$.

Observe also that $\{v_1, \ldots, v_k\}$ are the sources of T' and $\{v'_1, \ldots, v'_k\}$ are the sinks of T', and both are maximum independent sets of T'. We now apply Theorem 13.5.7, in order to obtain a covering of $V(T')$ by k disjoint paths $P'_1, \ldots, P'_k$ so that these paths start in $\{v_1, \ldots, v_k\}$ and end in $\{v'_1, \ldots, v'_k\}$. We can assume without loss of generality that the initial vertex of P'_i is exactly v_i, for all $i = 1, \ldots, k$. Let us now denote by σ the permutation of $\{1, \ldots, k\}$ such that $v'_{\sigma(i)}$ is the terminal vertex of P'_i, for all i. Back in D' the paths $P'_1, \ldots, P'_k$ correspond to a collection of paths $P_1, \ldots, P_k$ which cover all vertices of D' but are not necessarily disjoint (recall that T' is the transitive closure of D'). The only arc on P_i which is a backward arc in D is the last arc into $v_{\sigma(i)}$. Hence a cycle of length r in the permutation σ corresponds to a closed walk W back in D whose number of backward arcs with respect to $\mathcal{O}$ is exactly r. Note that W may intersect itself since, as we remarked, the paths $P_1, \ldots, P_k$ do not have to be vertex disjoint (see Figure 13.3).

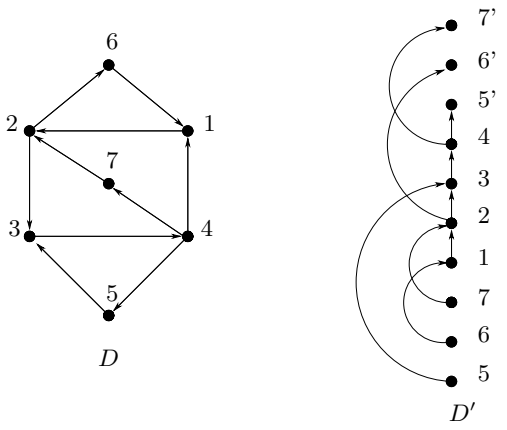


Figure 13.3 An illustration of the proof of Theorem 13.9.6. The left part of the figure shows a strong digraph D on 7 vertices with a coherent cyclic order 1,2,3,4,5,6,7 indicated. The given coherent order shows that 5,6,7 is a cyclic independent set of size 3. The right part of the figure shows the acyclic digraph D' obtained as described in the proof. In the transitive closure T' of D' the paths 5345', 6126' and 77' form a path cover of T' which maximizes the number of cycles in the permutation σ. Back in D' these correspond to the paths 5345', 6126' and 72347' which cover $V(D')$ but are not disjoint. In D these correspond to the three cycles 5345, 6126 and 72347 which span $V(D)$.

Now assume that among all sets of paths covering $V(D')$, we have chosen $P_1, \ldots, P_k$ (starting in $v_1, \ldots, v_k$, respectively) in such a way that the permutation σ has a maximum number of cycles. We claim that if $(i_1, \ldots, i_p)$ is a cycle of the permutation σ (meaning that $\sigma(i_j) = i_{j+1}$, $j = 1, 2, \ldots, p-1$ and $\sigma(i_p) = i_1$), then the paths $P_{i_1}, \ldots, P_{i_p}$ are pairwise vertex-disjoint (implying that the corresponding closed walk back in D is in fact a cycle). If not, suppose that v is a common vertex of P_{i_l} and P_{i_m}, and replace P_{i_l} by $P_{i_l}[v_{i_l}, v] \cup P_{i_m}[v, v_{\sigma(i_m)}]$ and P_{i_m} by $P_{i_m}[v_{i_m}, v] \cup P_{i_l}[v, v_{\sigma(i_l)}]$. This is a contradiction to the maximality of the number of cycles of σ. Now, in the set of paths $P_1, \ldots, P_k$, contract all the pairs $\{v_i, v_i'\}$, for $i = 1, \ldots, k$. This gives a spanning set $\mathcal{R}$ of cycles of D which satisfies $i_{\mathcal{O}}(\mathcal{R}) = k$. □

Now we can finish the proof of Gallai's conjecture.

Theorem 13.9.7 (Bessy-Thomassé theorem) *[158] Every strong digraph D is spanned by $\alpha(D)$ cycles.*

Proof: By Theorem 13.9.3, D has a coherent cyclic order $\mathcal{O}$. By Theorem 13.9.6, D is spanned by a set $\mathcal{R}$ of cycles such that $|\mathcal{R}| \leq i_{\mathcal{O}}(\mathcal{R}) = \alpha(\mathcal{O}) \leq \alpha(D)$. □

Corollary 13.9.8 *There is a polynomial algorithm which, given a strong digraph D and a coherent order $\mathcal{O}$ of D, calculates $\alpha_{\mathcal{O}}(D)$.*

Proof: Given D on n vertices and $\mathcal{O}$, we proceed as follows: Fix an ordering $E = v_1, v_2, \ldots, v_n$ in $\mathcal{O}$. Let D_{ST} be obtained from D by applying the vertex splitting procedure (see Section 4.2.4) to D. Form a flow network $\mathcal{N}$ from D_{ST} by assigning capacity n to all arcs, lower bound 1 to all arcs of the form $v_t v_s$, $v \in V$, lower bound 0 to all arcs of the kind $v_s w_t$ and finally giving all arcs a cost of 0, except those arcs $v_s w_t$ which correspond to an arc $vw \in A(D)$ which is a backward arc with respect to E. Every feasible circulation in $\mathcal{N}$ corresponds to a collection of cycles in D which cover all vertices and conversely. Hence, by Theorem 13.9.6, the cost of a minimum cost flow x in $\mathcal{N}$ equals $\alpha_{\mathcal{O}}(D)$. □

In the algorithm above we only find the number $\alpha_{\mathcal{O}}(D)$ and not an actual maximum cyclic independent set S. A polynomial algorithm to find such a set S can be derived from the proof of Theorem 13.9.6, see Exercise 13.36.

Charbit and Sebő [196] gave a shorter non-algorithmic proof of Theorem 13.9.7 based on LP-duality and total unimodularity. We briefly describe their main result and show how to deduce Theorem 13.9.7 from this. Let D be endowed with a cyclic order $\mathcal{O}$. A vertex-weighting $w : V \to \mathbb{N}$ of D is **index-bounded** (w.r.t. $\mathcal{O}$) if $w(C) \leq i_{\mathcal{O}}(C)$ for every cycle C of D.

Theorem 13.9.9 *[196] Let D be a digraph in which every vertex lies on a cycle. Suppose $\mathcal{O}$ is a cyclic order of D. Then the minimum index $i_{\mathcal{O}}(\mathcal{C})$ over all families $\mathcal{C}$ of cycles spanning $V(D)$ is equal to the maximum of $w(D)$ over all index bounded weightings w of D.* □

We can get Theorem 13.9.7 from this result as follows: Let D be strong and let $\mathcal{O}$ be a coherent cyclic order of D. Then $|\mathcal{C}| \leq i_{\mathcal{O}}(\mathcal{C})$ whenever $\mathcal{C}$ is a family of cycles spanning $V(D)$, because every cycle in $\mathcal{C}$ has index at least one. Furthermore, since every vertex is contained in a cycle of index one with respect to $\mathcal{O}$, every index-bounded weighting of D must be $\{0,1\}$-valued. Now we see that for a given index-bounded weighting w, the set of vertices with $w(v) = 1$ must form an independent set because every arc is contained in a cycle C with $i_{\mathcal{O}}(C) = 1$ and $w(C) \leq i_{\mathcal{O}}(C)$ as w is index-bounded. This shows that $w(D) \leq \alpha(D)$ and now Theorem 13.9.9 gives that $|\mathcal{C}| \leq i_{\mathcal{O}}(\mathcal{C}) \leq \alpha(D)$ for every family of cycles $\mathcal{C}$ which spans $V(D)$ and minimizes $i_{\mathcal{O}}(\mathcal{C})$.

A different algorithmic proof of Theorem 13.9.7 can be obtained by combining Corollary 13.9.4 with the algorithmic version of the coflow theorem by Cameron and Edmonds [188, 190]. For details see [191].

Theorem 13.9.10 (Coflow theorem) *[188, 190] Let $D = (V, A)$, let $\omega : A \to \mathbb{Z}_0$ be a weighting of its arcs and extend ω to sets of arcs in the obvious way*[4]*. Then*

[4] $\omega(X) = \sum_{a \in X} \omega(a)$ for every $X \subseteq A$.

$$\max\{|S| : S \subseteq V; \forall \text{ cycle } C, |S \cap C| \leq \omega(C)\}$$
$$= \min\{\sum_{C \in \mathcal{C}} \omega(C) + |V - \bigcup_{C \in \mathcal{C}} V(C)| : \mathcal{C} \text{ is a family of cycles of } D\}.$$

□

The **cyclomatic** number of an (un)directed graph $D = (V, A)$ is the parameter $|A| - |V| + c(D)$, where $c(D)$ denotes the number of connected components of $UG(D)$. A digraph is **cyclic** if every vertex belongs to a cycle. Note that a cyclic digraph is not necessarily strongly connected. The following conjecture, which Bondy [168] attributes to Chen and Manalastras [201], generalizes Gallai's conjecture above and Theorem 9.7.5.

Conjecture 13.9.11 *[168, 201] Every strong digraph D contains a cyclic spanning subdigraph with cyclomatic number at most $\alpha(D)$.*

The example below due to Favaron (see [168]) shows that one cannot hope to find, for every strong digraph D, a strong spanning subdigraph of D with cyclomatic number at most $\alpha(D)$. Let $r \geq 2$ and take r copies $T_1, T_2, \ldots, T_r$ of the strong tournament on four vertices. Let the vertices be labelled so that the unique hamiltonian cycle in the ith copy is $u_i x_i v_i y_i u_i$, $i = 1, 2, \ldots, r$. Let D_r be the digraph obtained from the disjoint union of $T_1, T_2, \ldots, T_r$ by adding the arcs $u_i u_{i+1}$ and $v_{i+1} v_i$ for all odd i, respectively, $u_{i+1} u_i$ and $v_i v_{i+1}$ for all even i, $1 \leq i \leq r$ (indices modulo r). Then D_r is strong, $\alpha(D_r) = r$ and it can be shown that D_r has no strong spanning subdigraph with cyclomatic number less than $2r - 1$ (Exercise 13.37). Moreover, every cyclic spanning subdigraph of D_r with cyclomatic number r consists of r disjoint 4-cycles.

13.10 Decomposing a Tournament into Strong Spanning Subdigraphs

The purpose of this section, based on [120], is to give a number of results supporting Conjecture 13.10.2 below.

The following result due to Yeo (personal communication, 2001) can be proved similarly to Theorem 6.1.3. Clearly this implies that it is an $\mathcal{NP}$-complete problem to decide whether a given digraph has two arc-disjoint spanning strong subdigraphs.

Theorem 13.10.1 *It is an $\mathcal{NP}$-complete problem to decide whether a given 2-regular digraph contains arc-disjoint hamiltonian cycles.* □

In [120] Bang-Jensen and Yeo posed the following conjecture which contains the Kelly conjecture (Conjecture 13.4.5) as the special case when $n = 2k + 1$.

Conjecture 13.10.2 *[120] The arc set of a tournament T can be decomposed into k arc-disjoint spanning strong subdigraphs if and only if T is k-arc-strong.*

Let S_{2k} be the semicomplete digraph one obtains from two disjoint copies of $\overleftrightarrow{K}_k$ with vertex sets $\{x_1, x_2, \ldots, x_k\}$ and $\{y_1, y_2, \ldots, y_k\}$ by adding a perfect matching $\{x_1y_1, x_2y_2, \ldots, x_ky_k\}$ oriented from the first copy to the second and all arcs of the kind y_jx_i where $i \neq j$ in the opposite direction. Note that S_{2k} has no 2-cycle of the form $x_py_qx_p$. The following result by Bang-Jensen and Yeo [120] verifies Conjecture 13.10.2 for 2-arc-strong tournaments.

Theorem 13.10.3 *Let D be a 2-arc-strong semicomplete digraph, on n vertices. Then D has two arc-disjoint spanning strong subdigraphs, if and only if it is not isomorphic to S_4.* □

We first verify Conjecture 13.10.2 in the case when T has a non-trivial $\lambda(T)$-cut, that is, there is a set $S \subset V(T)$ such that $d^+(S) = k$ and $\min\{|S|, |V(T) - S|\} \geq 2$. In this case, as we show below, the desired decomposition exists even in the case of semicomplete digraphs, except for one semicomplete digraph on four vertices.

We shall use two well-known results in this section. The first one was known under the name "The Evans conjecture" and originally dealt with partially completed Latin squares, but it can easily be restated as follows:

Theorem 13.10.4 *[825] Let B be a complete bipartite graph, with n vertices in each partite set, and let R be a set of edges in B, such that $|R| \leq n - 1$. Then we can decompose $E(B)$ into n edge-disjoint matchings $M_1, M_2, ..., M_n$, such that $|M_i \cap R| \leq 1$ for all $i = 1, 2, ..., n$.* □

Corollary 13.10.5 *[120] Let $B = (X, Y, E)$ be a complete bipartite graph (undirected), with $|X| = t$, $|Y| = s$ and $t > s$. Let R be a set of edges in B, such that $|R| \leq s$. Then we can colour the edges of B by $|R|$ colours in such a way that all edges in R receive distinct colours and every vertex in $X \cup Y$ is incident with all $|R|$ colours.* □

Lemma 13.10.6 *[120] Let D be a semicomplete digraph which is isomorphic to S_{2k} for some $k \geq 2$. Then D contains k arc-disjoint spanning strong subdigraphs except when $k = 2$, in which case D has no such decomposition.*

Proof: Exercise 13.20. □

Lemma 13.10.7 *Let D be a semicomplete digraph and let k be an integer, such that $\sum_{x \in V(D)} max\{0, k - d^+(x)\} \leq k - 1$. Then $|V(D)| \geq k + 1$.*

Proof: Let $n = |V(D)|$ and note that the following must hold: $n(n-1) \geq |A(D)| = \sum_{x \in V(D)} d^+(x) \geq nk - (k-1)$. By rearranging the terms we get that $(n-1)(n-k) \geq 1$, implying that $n \geq k + 1$. □

Theorem 13.10.8 *[120] Let $k \geq 1$ and let D be a k-arc-strong semicomplete digraph such that there exists a set $S \subset V(D)$, with $2 \leq |S| \leq |V(D)| - 2$ and $d^+(S) = k$. There exist k arc-disjoint strong spanning subdigraphs of D except if $D = S_4$.*

Proof: We may assume that D is not isomorphic to S_4 since, by Lemma 13.10.6, S_4 has no two arc-disjoint spanning strong subdigraphs.

Using an argument analogous to that in the proof of Lemma 13.10.7, we obtain that $k \leq |S| \leq n - k$ (by showing that $|S| \geq k$ and $|V(D) - S| \geq k$, respectively). If $|S| = |V - S| = 2$, then D contains S_4 as a proper spanning subdigraph and it is easy to check that adding any arc to S_4 will result in a digraph with two arc-disjoint strong spanning subdigraphs. Hence we may assume that $n \geq 5$. By reversing all arcs if necessary, we may assume that $|V - S| \geq |S|$.

Let $e_i = x_i y_i$, $i = 1, 2, \ldots, k$, be the k arcs from S to $V(D) - S$ and let $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$. Note that we may have $|X| < k$ or $|Y| < k$ or both. If $|S| = k$, then $X = S$ as no vertex of S can have more than $k-1$ out-neighbours in S. Hence, by Lemma 13.10.6 and the remark above, we may assume that $|V - S| > |S|$ if $|S| = k$. By Corollary 13.10.5 (with $R = \{e_1, e_2, \ldots, e_k\}$), we can colour all arcs between S and $V(D) - S$ with k colours, such that the arcs from S to $V(D) - S$ get different colours and every vertex in V is incident with arcs of all k colours. Note that if $|V - S| = |S| > k$, this follows from Theorem 13.10.4.

Assume, without loss of generality, that the arc $x_i y_i$ is coloured with colour i, and let F_i contain all arcs between S and $V(D) - S$ of colour i.

By Theorem 9.3.1, $D\langle V(D) - S\rangle$ contains k arc-disjoint out-branchings $B^+_{y_1}, B^+_{y_2}, \ldots, B^+_{y_k}$, such that $B^+_{y_i}$ is rooted at y_i, for $i = 1, 2, \ldots, k$ (consider k arc-disjoint out-branchings rooted at any vertex $s \in S$ in D. Each of these must contain exactly one of the arcs $e_1, e_2, \ldots, e_k$. Thus the out-branching that contains the arc e_i must contain an out-branching from y_i in $D\langle V(D) - S\rangle$). Analogously, there exist k arc-disjoint in-branchings $B^-_{x_1}, B^-_{x_2}, \ldots, B^-_{x_k}$, in $D\langle S\rangle$, such that $B^-_{x_i}$ is rooted at x_i, for $i = 1, 2, \ldots, k$. Let $H_i = B^-_{x_i} \cup B^+_{y_i} \cup F_i$, for $i = 1, 2, \ldots, k$. Clearly $H_1, H_2, \ldots, H_k$ are arc-disjoint and spanning. Each H_i is furthermore strong: by the construction of the colouring, every vertex in V is incident to an arc of colour i, every vertex in $V(D) - S - y_i$ has an arc in H_i into S, and hence every vertex in V can reach y_i (via $B^-_{x_i}$ and the arc $x_i y_i$) and every vertex in $S - x_i$ has an arc in H_i from $V(B^+_{y_i})$, implying that all vertices of V can be reached by y_i. This completes the proof. □

The theorem below verifies Conjecture 13.10.2 in the case when T has no vertex of in- or out-degree smaller than $37k$. Given a digraph $D = (V, A)$ we use the notation $d_X(v)$, where $X \subseteq A$, to denote the number of arcs in X which have v as either their tail or head (that is, it is the degree of v in the underlying graph of $D\langle X\rangle$). We need the following technical lemma.

Lemma 13.10.9 *[107] Let T be a tournament and R a subset of $A(T)$. Suppose that $\sum_{\{u\in V(T):d_R(u)>c\}}[d_R(u)-c]\leq q$ and that z is a vertex such that*

$$d_T^+(u)\leq d_T^+(z)+\gamma \quad \text{for all } u\in V(T). \tag{13.8}$$

Let W be the set of vertices which are not reachable from z by a directed path in $D=T-R$. Then $|W|\leq 2c+2d_R(z)+2\gamma-1+\sqrt{2q}$. □

Theorem 13.10.10 *Let T be a k-arc-strong tournament, with $\delta^0(T)\geq 37k$. Then there exist k arc-disjoint spanning strong subdigraphs in T.*

Proof: Let $T=(V,A)$ be a k-arc-strong tournament on n vertices, with $\delta^0(T)\geq 37k$. Let $v_1,v_2,\ldots,v_n$ be an ordering of the vertices of T such that $d^+(v_1)\leq d^+(v_2)\leq\ldots\leq d^+(v_n)$. Note that since T is a tournament this ordering also satisfies $d^-(v_1)\geq d^-(v_2)\geq\ldots\geq d^-(v_n)$. Let $X=\{v_{n-k+1},v_{n-k+2},\ldots,v_n\}$ and $Y=\{v_1,v_2,\ldots,v_k\}$.

Since T is k-arc-strong, it follows from Exercise 5.15 that there are k arc-disjoint paths $P_1,P_2,\ldots,P_k$ from Y to X such that all end-vertices of these paths are disjoint. Let $y_1,y_2,\ldots,y_k$ and $x_1,x_2,\ldots,x_k$ be distinct vertices which are chosen such that $X=\{x_1,x_2,\ldots,x_k\}$, $Y=\{y_1,y_2,\ldots,y_k\}$ and P_i is a (y_i,x_i)-path for $i=1,2,\ldots,k$. Note that each P_i may contain several vertices from $X\cup Y-\{x_i,y_i\}$.

Let $c_i=\gamma_i=2k-2$ for all $i\in[k]$ and define $q_1,q_2,\ldots,q_{k+1}$ recursively as follows:

$$\begin{aligned} q_1&=0,\\ q_i&=q_{i-1}+2(2k-2+\sqrt{2q_{i-1}}),\quad i=2,3,\ldots,k+1. \end{aligned} \tag{13.9}$$

Note that we have $q_1<q_2<\ldots<q_{k+1}$ and it is not difficult to show by induction on i that $q_i<16k(i-1)$ (Exercise 13.38). In particular, we have $q_{k+1}<16k^2$.

We will now construct arc-disjoint strong subdigraphs $H_1,H_2,\ldots,H_k$ of T (in that order). Given $H_1,H_2,\ldots,H_{i-1}$ ($i\in[k-1]$), we define the following sets $L_i,R_i\subseteq A$ and $Z_i,W_i\subseteq V$.

- $L_i=A(H_1)\cup A(H_2)\cup\ldots\cup A(H_{i-1})$.
- $R_i=L_i\cup A(P_{i+1})\cup A(P_{i+2})\cup\ldots\cup A(P_k)$.
- $Z_i=\{z\in V|d_{L_i}(z)\geq 10k\}$.
- $W_i=X\cup Y\cup Z_i-\{x_i,y_i\}$.

Define the digraphs D_i,D_i^* and D_i^{**} as follows: $D_i=T-R_i$, $D_i^*=D_i-W_i$ and $D_i^{**}=D_i^*\cup V(P_i)\cup A(P_i)$.

Assume that we have already found arc-disjoint strong[5] subdigraphs $H_1,H_2,\ldots,H_{i-1}$ so that the following holds.

[5] Note that these subdigraphs may not be spanning in T.

(a) $\sum_{j=1}^{i-1} \sum_{d_{H_j}(u)>2} [d_{H_j}(u) - 2] \leq q_i$.
(b) $\sum_{d_{R_i}(u)>c_i} [d_{R_i}(u) - c_i] \leq q_i$.
(c) $|Z_i| \leq 2k$ and $|W_i| \leq 4k - 2$.
(d) $d_{R_i}(x_i), d_{R_i}(y_i) \leq 2k - 2$.

Claim: D_i^{**} is strongly connected.

Proof of Claim: Let Q be all the vertices in D_i^* which cannot be reached by x_i. By Lemma 13.10.9 (with $\gamma = 4k - 2$) we get that

$$\begin{aligned} |Q| &\leq 2c_i + 2d_{R_i}(x_i) + 2(4k-2) - 1 + \sqrt{2q_i} \\ &\leq (4k-4) + (4k-4) + (8k-4) - 1 + 6k \\ &= 22k - 9. \end{aligned} \tag{13.10}$$

Note that D_i^* contains no vertex from Z_i (neither x_i nor y_i belongs to Z_i by (d)). Let $r \in V(D_i^*)$ be arbitrary. Since $r \notin Z_i$ we have

$$d^-_{D_i^*}(r) \geq 37k - 10k - k - (4k-2) \tag{13.11}$$

$$= 22k + 2. \tag{13.12}$$

This follows from the fact that there are at least 37k arcs into r in T and as we have used at most $10k$ of them in the H_j's so far (as $r \notin Z_i$) at most k in the P_j's and at most $4k-2$ into vertices in W_i (by (c)). It follows from (13.10) and (13.11) that r can be reached from x_i (as if r could not be reached, then $N^-(r)$ could not be reached from x_i). Analogously we get that all vertices in D_i^* can reach y_i in D_i^*. Since $D_i^{**} = D_i^* \cup V(P_i) \cup A(P_i)$ it follows that D_i^{**} is strong, which proves the claim. □

By Corollary 12.1.5 (with c_i and q_i, in the place of c and q), we can find a strong spanning subdigraph H_i of D_i^{**} with $|A(H_i)| \leq |V(H_i)| + c_i + \sqrt{2q_i}$.

We can now prove (a)-(d) by induction on i. Suppose first that $i = 1$. Then (a) holds vacuously and since $L_1 = \emptyset$ and $R_1 = A(P_2) \cup \ldots \cup A(P_k)$ it follows that (c) and (d) hold. Finally, as no vertex is incident to more than two arcs on each P_j, $j = 1, 2, \ldots, k$ and $c_1 = 2k - 2$, (b) also holds.

Suppose now that (a) and (b) hold for some $i < k$. We will now show that (a) and (b) hold for $i+1$. By the construction of H_i above we have $|A(H_i)| \leq |V(H_i)| + c_i + \sqrt{2q_i}$. Since every vertex in H_i has degree at least two (in the undirected sense) this implies that $\sum_{d_{H_i}(u)>2} [d_{H_i}(u) - 2] \leq 2(c_i + \sqrt{2q_i})$. By the recursive definition of q_{i+1} we have $q_{i+1} - q_i = 2(2k - 2 + \sqrt{2q_i}) = 2(c_i + \sqrt{2q_i})$. Now we see that (a) holds for $i+1$. To see that (b) holds given (a), it suffices to observe that every vertex has degree at least 2 in every H_j constructed so far and at most 2 in each P_t, $t \in [k]$. Hence every vertex u contributing to the sum in (b) contributes with at least the same amount to the sum in (a).

Note that every vertex in Z_{i+1} must contribute at least $10k - c_{i+1}$ to the sum in (b), implying that $q_{i+1} \geq |Z_{i+1}|(8k+2)$. As we have seen that $q_{i+1} < 16k^2$, this implies that $|Z_{i+1}| < 2k$, which was the first part of (c). The second part of (c) follows immediately from this and the definition of W_i.

In order to prove that (d) holds for $i+1$, we note that if x_{i+1} or y_{i+1} are used in any subdigraph $H_j \in \{H_1, H_2, \ldots, H_{i-1}\}$, then it must be because it lies on the corresponding path, P_j, in which case it will have degree at most 2 in H_j. This implies that the degrees of x_{i+1} and y_{i+1} are at most 2 in each of the subdigraphs $H_1, H_2, \ldots, H_{i-1}, P_{i+1}, P_{i+2}, \ldots, P_k$, implying (d).

We have now constructed $H_1, H_2, \ldots, H_k$. Let $H^* = V(H_1) \cap V(H_2) \cap \ldots \cap V(H_k)$ and $W^* = X \cup Y \cup \{z | d_{R^*}(z) \geq 10k\}$, where $R^* = A(H_1) \cup A(H_2) \cup \ldots \cup A(H_k)$. Note that $V(T) - H^* \subseteq W^*$, as $W_i \subseteq W^*$. Let $w \in W^*$ be arbitrary, and assume that $w \notin Z_i$ but $w \in Z_{i+1}$. By the construction of H_j, $j \geq i+1$, this means that w is incident to at most two arcs in each H_j. By the construction of H_i we have $|A(H_i)| \leq |V(H_i)| + c_i + \sqrt{2q_i}$. This implies that $d_{H_i}(w) \leq 2 + 2c_i + 2\sqrt{2q_i}$ (consider any ear decomposition of H_i). Now we see that

$$\begin{aligned} d_{R^*}(w) &\leq d_{L_i}(w) + d_{H_i}(w) + \sum_{j=i+1}^{k} d_{H_j}(w) \\ &\leq 10k + (2 + 2c_i + 2\sqrt{2q_i}) + 2(k-i) \\ &\leq 10k + (16k+2) + 2k \\ &\leq 28k + 2 \leq 30k. \end{aligned}$$

Furthermore $|W^*| \leq 4k$ by a similar argument as when we proved (c) above (using that $q_{k+1} \leq 16k^2$). Since $\delta^0(T) \geq 37k$ this implies that every vertex in W^* has at least $3k$ arcs into H^* and $3k$ arcs from H^*. Therefore it is not difficult to connect every vertex in W^* to every H_i, which it does not already belong to, by disjoint arcs (one in each direction). This gives us the desired arc-disjoint spanning strong subdigraphs. □

The following two conjectures represent successive weakenings of Conjecture 13.10.2.

Conjecture 13.10.11 *[120] Let k, s and t be natural numbers such that $k = s + t$. Then every k-arc-strong tournament contains arc-disjoint spanning subdigraphs D_1, D_2 such that D_1 is s-arc-strong and D_2 is t-arc-strong.*

Conjecture 13.10.12 *[120] Every k-arc-strong tournament contains a spanning strong subdigraph H such that $T - A(H)$ is $(k-1)$-arc-strong.*

Thomassen proved [857, Theorem 4.2] that every 2-arc-strong tournament T contains a hamiltonian path P such that $T - A(P)$ is strong. It is interesting

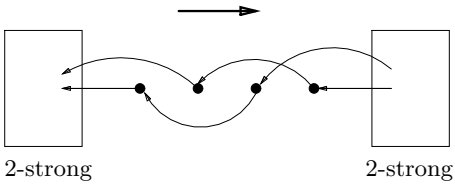


Figure 13.4 An infinite family of 2-strong tournaments such that the deletion of the arcs of any hamiltonian cycle leaves a non-strong digraph. The first and the last box symbolizes arbitrary 2-strong tournaments and the fat arc indicates that except for the 6 arcs shown from right to left all other arcs go from left to right [120].

to note that we cannot replace hamiltonian path by hamiltonian cycle above, as shown by the infinite class of 2-(arc-)strong tournaments in Figure 13.4.

Conjecture 13.10.13 *[120] Except for finitely many exceptions for each k, every k-arc-strong semicomplete digraph can be decomposed in k arc-disjoint spanning strong subdigraphs.*

We have already seen (Lemma 13.10.6) that the 2-strong semicomplete digraph S_4 cannot be decomposed into two strong spanning digraphs.

The following conjecture by Bang-Jensen and Yeo would immediately imply Thomassen's conjecture on arc-disjoint in- and out-branchings in highly arc-strong digraphs (Conjecture 9.6.9).

Conjecture 13.10.14 *[120] There exists a natural number K such that every K-arc-strong digraph contains two arc-disjoint strong spanning subdigraphs.*

13.11 The Directed Path-Partition Conjecture

The majority of this section is based on the paper [114] by Bang-Jensen, Nielsen and Yeo. Recall that given a digraph D, $\mathrm{lp}(D)$ denotes the order of a longest path in D. The Gallai-Roy-Vitaver theorem (Theorem 11.3.1) states that the chromatic number of (the underlying graph of) a digraph D is at most $\mathrm{lp}(D)$. In 1983 Laborde, Payan and Xuong posed the following conjecture which extends this theorem in a natural way.

Conjecture 13.11.1 *[633] Every digraph D contains an independent set X, such that $\mathrm{lp}(D - X) < \mathrm{lp}(D)$.*

The corresponding statement for undirected graphs is easily seen to be true (just take any maximal independent set). Conjecture 13.11.1 seems very difficult, however, and only a few partial results have been obtained. Clearly,

if the digraph has a kernel, then removing any kernel will decrease the order of every longest path. Havet [507] verified the conjecture for digraphs with independence number at most two. The following, more general, conjecture is also due to Laborde et al.

Conjecture 13.11.2 (Path Partition conjecture) *[633] For every digraph D and every choice of positive integers, ℓ_1, ℓ_2, such that $\mathrm{lp}(D) = \ell_1 + \ell_2$, there exists a partition of D into two digraphs, D_1 and D_2, such that $\mathrm{lp}(D_i) \leq \ell_i$, for $i = 1, 2$.*

A seemingly stronger version of Conjecture 13.11.2 is stated in [169]. Bondy attributes it to Laborde et al. [633] although only the undirected version of Conjecture 13.11.2 is explicitly mentioned there.

Conjecture 13.11.3 *[169] For every digraph D and every choice of positive integers, ℓ_1, ℓ_2, such that $\mathrm{lp}(D) = \ell_1 + \ell_2$, there exists a partition of D into two digraphs, D_1 and D_2, such that $\mathrm{lp}(D_i) = \ell_i$, for $i = 1, 2$.*

All three conjectures above are very difficult to attack for general digraphs, since very little can be said about the structure of longest paths in general digraphs. They are all trivial for connected locally semicomplete digraphs as every such digraph has a hamiltonian path (Theorem 6.3.3). The proof of Conjecture 13.11.2 is even more trivial in the case of bipartite digraphs, since we may simply let D_1 and D_2 be the arc-less digraphs induced by the two independent sets of an arbitrary bipartition. Already for quasi-transitive digraphs the conjectures seem non-trivial.

Theorem 13.11.4 *[114] Let D be a digraph which is either extended semicomplete or locally in-semicomplete. For every choice of positive integers, ℓ_1, ℓ_2, such that $\mathrm{lp}(D) = \ell_1 + \ell_2$, there exists a partition of D into two digraphs, D_1 and D_2, such that $\mathrm{lp}(D_i) = \ell_i$, for $i = 1, 2$.* □

Below we give a proof of an extension of Conjecture 13.11.2 for quasi-transitive digraphs. We will use the following two lemmas whose proofs are left to the reader as Exercises 13.21 and 13.22.

Lemma 13.11.5 *[114] Let $D = S[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_s}]$, where S is a semicomplete digraph and let $l_{i,k}$ denote the maximum number of vertices of $\overline{K}_{n_i}$ that can be covered by a k-path subdigraph in D. Then every maximum k-path subdigraph in D covers exactly $l_{i,k}$ vertices of E_i, for every $i \in [s]$.* □

Denote by $\mathrm{lp}_k(D)$ the maximum number of vertices in a k-path subdigraph of D. A set of k disjoint paths in D with $\mathrm{lp}_k(D)$ vertices is called a **maximum** k-path subdigraph of D.

Lemma 13.11.6 *[114] Let $D = S[Q_1, Q_2, \ldots, Q_s]$, where S is a strong semicomplete digraph and each Q_i is either a single vertex or a non-strong quasi-transitive digraph. For every $k \in \{1, 2, \ldots, |V(D)|\}$ and $i \in [s]$, there exists*

an integer, $v_{i,k}$, such that every maximum k-path subdigraph, P_k, of D satisfies $|V(Q_i) \cap V(P_k)| = v_{i,k}$ and no k-path subdigraph of D contains more than $v_{i,k}$ vertices of Q_i. □

Theorem 13.11.7 *[114] Let D be a quasi-transitive digraph or a strong extended semicomplete digraph, and let q be any positive integer. Then there exists a partition X, Y of $V(D)$ such that the following holds:*

(i) $\mathrm{lp}(D\langle X\rangle) \leq q$,
(ii) $\mathrm{lp}_k(D\langle Y\rangle) \leq \mathrm{lp}_k(D) - q$, *for all* $k = 1, 2, 3, ..., |V(Y)|$, *provided* $\mathrm{lp}_k(D) - q \geq 0$.

Proof: We prove the theorem by induction on $|V(D)|$. For the base case, when $|V(D)| = 1$, the claim is trivially true.

Suppose first that D is strong. By Theorem 2.7.5 and the definition of an extended semicomplete digraph, we may let $D = S[Q_1, Q_2, ..., Q_s]$, where S is a strong semicomplete digraph and each Q_i is either a non-strong quasi-transitive digraph (in particular, it could be an independent set of vertices) or a single vertex, for every $i \in [s]$. Let $v_{i,k}$ be defined as in Lemma 13.11.6 and assume without loss of generality that $v_{i,1} < |Q_i|$, for all $i \in [l]$ and $v_{j,1} = |Q_j|$, for $j \in \{l+1, l+2, \ldots, s\}$ (i.e., there is no path in D containing all the vertices of Q_i, for any $i \in [l]$, but there is a path in D containing all the vertices of $Q_{l+1} \cup Q_{l+2} \cup \ldots \cup Q_s$). We may assume that $q < v_{1,1} + v_{2,1} + \ldots + v_{s,1}$, since otherwise $(X, Y) = (V(D), \emptyset)$ is the desired partition. Now define r and γ_r such that the following holds:

$$\gamma_r = v_{1,1} + v_{2,1} + \ldots + v_{r-1,1} < q \leq v_{1,1} + v_{2,1} + \ldots + v_{r,1}.$$

We now consider the cases $r \leq l$ and $r > l$ separately and assume first that $r \leq l$. Let $q' = q - \gamma_r > 0$ and use our induction hypothesis to partition Q_r into (X_r, Y_r) such that $\mathrm{lp}(Q_r\langle X_r\rangle) \leq q'$ and $\mathrm{lp}_k(Q_r\langle Y_r\rangle) \leq \mathrm{lp}_k(Q_r) - q'$, for all $k = 1, 2, 3, ..., |V(Y_r)|$. Let $X = V(Q_1) \cup V(Q_2) \cup \ldots \cup V(Q_{r-1}) \cup X_r$ and let $Y = Y_r \cup V(Q_{r+1}) \cup V(Q_{r+2}) \cup \ldots \cup V(Q_s)$. We will now show that X and Y fulfill the conditions (i) and (ii) in the theorem.

Denote the vertices of S by $V(S) = [s]$, where i has been expanded to Q_i in D. We first show that $S_l = S\langle\{1, 2, \ldots, l\}\rangle$ is acyclic. Indeed, assume that $C = c_1 c_2 \ldots c_z c_1$ is a cycle in S_l, and without loss of generality assume that $|V(Q_{c_1})| \leq |V(Q_{c_j})|$ for all $j = 2, 3, \ldots, z$. It is not difficult to see (by going around the cycle, C, $|V(Q_{c_1})|$ times) that there exists a cycle (and, hence, a path) in D which contains all the vertices in Q_{c_1}, and hence it follows from Lemma 13.11.6 that $v_{c_1,1} = |Q_{c_1}|$, which contradicts the definition of l. Therefore S_l is acyclic and hence, if $i \leq l$, no path in D visits Q_i more than once. This implies the following:

$$\begin{aligned}\mathrm{lp}(D\langle X\rangle) &= \mathrm{lp}(Q_1) + \mathrm{lp}(Q_2) + \ldots + \mathrm{lp}(Q_{r-1}) + \mathrm{lp}(Q_r\langle X_r\rangle)\\ &\leq \gamma_r + q'\\ &= q,\end{aligned}$$

where the first equality follows from the fact that $S\langle\{1,2,\ldots,r\}\rangle$ is semicomplete and hence traceable. Let $k \in \{1,2,3,...,|V(Y)|\}$ be arbitrary, let W_k be a maximum k-path subdigraph of $D\langle Y\rangle$ and assume that $W_k \cap Q_r$ consists of $b \geq 0$ paths. Then the intersection of Q_r with any maximum k-path subdigraph of D consists of at least b paths, and hence $v_{r,k} \geq \mathrm{lp}_b(Q_r)$ (where we define $\mathrm{lp}_0(\cdot) = 0$). If $b > 0$, we may assume that $q' < \mathrm{lp}(Q_r)$, which implies that $\mathrm{lp}_b(Q_r) - q' > 0$; hence, by our induction hypothesis, $\mathrm{lp}_b(Q_r\langle Y_r\rangle) \leq \mathrm{lp}_b(Q_r) - q' \leq v_{r,k} - q'$. Thus, for any value of b, we have:

$$\begin{aligned}\mathrm{lp}_k(D\langle Y\rangle) &\leq \mathrm{lp}_b(Q_r\langle Y_r\rangle) + v_{r+1,k} + v_{r+2,k} + \ldots + v_{s,k}\\ &\leq v_{r,k} + v_{r+1,k} + \ldots + v_{s,k} - q'\\ &= v_{1,1} + v_{2,1} + \ldots + v_{r-1,1} + v_{r,k} + v_{r+1,k} + \ldots + v_{s,k} - q\\ &\leq v_{1,k} + v_{2,k} + \ldots + v_{r-1,k} + v_{r,k} + v_{r+1,k} + \ldots + v_{s,k} - q\\ &= \mathrm{lp}_k(D) - q.\end{aligned}$$

Suppose now that $r > l$. Let $q' = q - \gamma_r > 0$ and let X_r be any subset of q' vertices from Q_r. Let $Y_r = V(Q_r) - X_r$, and define X and Y as we did in the previous case. By Lemma 13.11.6:

$$\begin{aligned}\mathrm{lp}(D\langle X\rangle) &\leq v_{1,1} + v_{2,1} + \ldots + v_{r-1,1} + |X_r|\\ &= \gamma_r + (q - \gamma_r).\end{aligned}$$

Therefore (i) holds; we will now prove (ii). Let $k \in \{1,2,3,...,|V(Y)|\}$ be arbitrary and note that $v_{r,k} = |Q_r|$. This implies the following:

$$\begin{aligned}\mathrm{lp}_k(D\langle Y\rangle) &\leq |Y_r| + v_{r+1,k} + v_{r+2,k} + \ldots + v_{s,k}\\ &= (v_{r,k} - q') + v_{r+1,k} + \ldots + v_{s,k}\\ &= v_{1,1} + v_{2,1} + \ldots + v_{r-1,1} + v_{r,k} + v_{r+1,k} + \ldots + v_{s,k} - q\\ &\leq v_{1,k} + v_{2,k} + \ldots + v_{r-1,k} + v_{r,k} + v_{r+1,k} + \ldots + v_{s,k} - q\\ &= \mathrm{lp}_k(D) - q,\end{aligned}$$

which completes the case when D is strong.

Now suppose D is a non-strong quasi-transitive digraph. By Theorem 2.7.5, there is a transitive oriented graph T and strong quasi-transitive digraphs $H_1, H_2, \ldots, H_t$ such that $D = T[H_1, H_2, \ldots, H_t]$. Define p_i^{in} as the maximum number of vertices on a path in $D - V(H_i)$, such that the terminal vertex on the path has an arc into H_i. Define p_i^{end} to be the maximum number of vertices on a path in D, such that the terminal vertex on the path belongs to H_i. We will place vertices of D into X and Y as follows:

(a) If $p_i^{end} \leq q$: Put $V(H_i)$ into X.
(b) If $p_i^{in} \geq q$: Put $V(H_i)$ into Y.
(c) If $p_i^{in} < q < p_i^{end}$: Let $q_i' = q - p_i^{in}$ and use our induction hypothesis to partition H_i into (X_i, Y_i) such that $\mathrm{lp}(H_i\langle X_i\rangle) \leq q_i'$ and $\mathrm{lp}_k(H_i\langle Y_i\rangle) \leq$

$\mathrm{lp}_k(H_i) - q_i'$, for all $k = 1, 2, 3, ..., |V(Y_i)|$. Put X_i into X and put Y_i into Y.

The above defines our partition, so we will now show that (i) and (ii) hold. Let P be a longest path in $D\langle X\rangle$, and assume that the terminal vertex belongs to H_i. If H_i was considered in (a), then clearly $|V(P)| \leq p_i^{end} \leq q$ and if H_i was considered in (c), then $|V(P)| \leq p_i^{in} + q_i' = q$ (since, for every j such that $H_i \to H_j$, we have $H_j \subseteq Y$).

Let $k \in \{1, 2, \ldots, |V(Y)|\}$ be arbitrary and let W_k be a k-path subdigraph of $D\langle Y\rangle$, such that $|V(W_k)| = \mathrm{lp}_k(D\langle Y\rangle)$. Let P be any path in W_k and assume that P starts in the vertex $x \in V(H_i)$. If H_i was considered in (b), then there is a path P' in D, such that $|V(P')| \geq q$ and the terminal vertex in P' dominates x. However, merging P and P' into one path and considering this path together with the $k-1$ paths in $W_k - P$, we see that $\mathrm{lp}_k(D) \geq \mathrm{lp}_k(D\langle Y\rangle) + q$. Therefore we may assume that H_i was considered in (c).

Suppose that b paths of W_k start in H_i. By our induction hypothesis we know that $\mathrm{lp}_b(H_i\langle Y_i\rangle) \leq \mathrm{lp}_b(H_i) - q_i'$. We can obtain a k-path subdigraph of D by substituting the b paths in $H_i\langle Y_i\rangle$ by b paths in H_i and prepending a path of order p_i^{in} to one of the paths (as we did above). This implies that

$$\begin{aligned}\mathrm{lp}_k(D) &\geq p_i^{in} + \mathrm{lp}_b(H_i) + (|W_k| - \mathrm{lp}_b(H_i\langle Y_i\rangle))\\ &\geq p_i^{in} + |W_k| + q_i'\\ &= |W_k| + q\\ &= \mathrm{lp}_k(D\langle Y\rangle) + q,\end{aligned}$$

which completes the proof. □

The proof of the following corollary is left to the reader as Exercise 13.24. The case when D is strong is covered in Theorem 13.11.7.

Corollary 13.11.8 *[114] Let D be an extended semicomplete digraph and let q be any positive integer. Then there exists a partition X, Y of $V(D)$ such that $\mathrm{lp}(D\langle X\rangle) \leq q$ and $\mathrm{lp}_k(D\langle Y\rangle) \leq \mathrm{lp}_k(D) - q$, for $k = 1, 2, 3, ..., |V(Y)|$, provided $\mathrm{lp}_k(D) - q \geq 0$.* □

13.12 Miscellaneous Topics

13.12.1 Maximum One-Way Cuts and Covering by One-Way Cuts

Recall that a directed cut is a set of arcs of the form $(X, V - X)$ (that is, the set of all arcs from X to $V - X$) where $X, V - X \neq \emptyset$ and $d^-(X) = 0$. If we drop the last condition, we speak about a **one-way cut**. Clearly every

non-trivial partition $Y, V-Y$ of the vertex set of a digraph gives rise to two one-way cuts, one from Y to $V-Y$ and one from $V-Y$ to Y.

It is an easy fact that every undirected graph has a bipartition containing at least half of its edges (the obvious greedy algorithm will produce such a subgraph). Edwards [289] improved this bound by showing that one can always find a bipartite subgraph with at least $\frac{m}{2}+\sqrt{\frac{m}{8}+\frac{1}{64}}-\frac{1}{8}$ edges in any graph with m edges (with equality for complete graphs of odd order). This immediately implies that every digraph with m arcs has a one-way cut of size at least $\frac{m}{4}+\sqrt{\frac{m}{32}+\frac{1}{256}}-\frac{1}{16}$. Regular tournaments show that the bound can be achieved, so if we want to ensure the existence of larger one-ways cuts, we must make some assumptions on the digraphs.

In [20] Alon, Bollobàs, Gyàrfàs, Lehel and Scott studied lower bounds on the maximum one-way cut in acyclic digraphs. They proved that $\frac{1}{4}$ is still the best fraction of the total number of arcs one can guarantee in a one-way cut, although one can get more arcs in a one-way cut than for arbitrary digraphs. Let $h(m)$ denote the minimum, over all acyclic digraphs D with m arcs, of the size of a maximum one-way cut in D.

Theorem 13.12.1 *[20] For $m \geq 1$, $\frac{m}{4}+\Omega(m^{3/5}) \leq h(m) = \frac{m}{4}+O(m^{4/5})$.* □

Theorem 13.12.2 *[20] The minimum number of one-way cuts needed to cover the arcs of the complete digraph on n vertices is equal to*

$$c(n) = \min\left\{k : \binom{k}{\lfloor k/2 \rfloor} \geq n\right\} = \log_2 n + \frac{1}{2}\log_2\log_2 n + O(1).$$

Proof: Given a collection $\mathcal{S}$ of one-way cuts $(X_1, Y_1), \ldots, (X_p, Y_p)$ of the complete digraph with vertex set V, we associate with each $v \in V$ the set $S_v = \{i : v \in X_i\}$. Now it is easy to see that $\mathcal{S}$ covers all arcs in the complete digraph with vertex set V if and only if $\{S_v\}_{v\in V}$ forms an antichain. Thus the bound in the theorem follows from Sperner's Lemma. □

Problem 13.12.3 *[20] What is the smallest $d(k)$ such that the arcs of every acyclic digraph D with $\delta^+(D) \leq k$ can be covered by $d(k)$ one-way cuts?*

Theorem 13.12.4 *[20] Let D be a digraph with m arcs and $\delta^+(D) \leq k$. Then D has a one-way cut of size at least $\frac{k+1}{4k+2}m$.*

Proof: Let G be the underlying multigraph of D. By Lemma 11.3.3, $\chi(G) \leq 2k+1$ so let us pick a $(2k+1)$-colouring of G. By identifying the vertices in each colour class of G and keeping multiple edges, we obtain a weighted copy K of K_{2k+1}. Now take a random partition of $V(K)$ into two sets of sizes k

and $k+1$, respectively, chosen uniformly from the set of all such partitions. The expected weight of the resulting cut is $\frac{k+1}{2k+1}m$. Thus K has a cut of weight $w \geq \frac{k+1}{2k+1}m$. In G this cut corresponds to a cut $[X,Y]$ with w edges, implying that, back in D, either (X,Y) or (Y,X) is a one-way cut whose weight is at least $\frac{k+1}{4k+2}m$. □

13.12.2 Acyclic Decompositions of Digraphs

By an **acyclic arc-decomposition** of a digraph $D = (V,A)$ we mean a partition of A into sets $A_1, A_2, \ldots, A_s$, $s \geq 2$, such that the subdigraph induced by A_i is acyclic for all $i \in [s]$. Wood proved the following result which is best possible.

Theorem 13.12.5 *[907] For every integer $s \geq 2$, every digraph $D = (V,A)$ has an acyclic arc-decomposition $A_1, A_2, \ldots, A_s$ such that $d^+_{A_i}(v) \leq \lceil \frac{d^+(v)}{s-1} \rceil$ for all $v \in V$ and $i \in [s]$.* □

Instead of decomposing the arc set of a digraph into acyclic subdigraphs we may also consider decompositions of the vertex set such that each subset of the vertices in the decomposition induces an acyclic digraph. The complete digraph shows that when considering vertex partitions, we must assume something about the structure of the digraph in order to guarantee the existence of a partition into k vertex induced subdigraphs all of which are acyclic. According to Thomassé the following conjecture was posed independently by Neumann-Lara and Škrekovski.

Conjecture 13.12.6 *If D is an orientation of a planar graph, then there is a partition of $V(D)$ into $\{X_1, X_2\}$ so that $D\langle X_i \rangle$ is acyclic for $i = 1, 2$.*

13.12.3 Decomposing Tournaments into Strong Subtournaments

Since every strong tournament has a hamiltonian cycle, a tournament T contains a 2-cycle factor if and only if T can be partitioned into two strong subtournaments. Thomassen posed the following problem which generalizes the problem of the existence of a 2-cycle factor in a tournament.

Problem 13.12.7 (Thomassen) *[773] Is it true that for all natural numbers r, s, there exists a natural number $f(r,s)$ with the following property: except for finitely many exceptions for each r, s, every $f(r,s)$-strong tournament T can be partitioned into an r-strong tournament T_1 and an s-strong tournament T_2?*

It follows from Theorem 13.7.1 that $f(1,1) = 2$. It is worth noticing that the problem of determining the analogue $f'(1,1)$ of $f(1,1)$ for semicomplete digraphs is open. Since every 3-strong semicomplete digraph contains a spanning 2-strong tournament (Proposition 11.10.5), we obtain that $2 \leq f'(1,1) \leq 3$ holds for semicomplete digraphs.

The next conjecture by Bang-Jensen, Guo and Yeo goes further than Problem 13.12.7. It may be seen as a first step towards studying partitions into subtournaments containing prescribed vertices in highly connected tournaments.

Conjecture 13.12.8 *[83] For all natural numbers r, s there exists a natural number $g(r,s)$ such that the following is true with no more than finitely many exceptions for each choice of r, s: for every tournament T which is $g(r,s)$-strong and every choice of distinct vertices $x, y \in V(T)$, there exist vertex-disjoint subtournaments T_x, T_y of T such that $V(T) = V(T_x) \cup V(T_y)$, T_x is r-strong, T_y is s-strong and $x \in V(T_x)$, $y \in V(T_y)$.*

Note that it is easy to decide in polynomial time whether a tournament T contains two disjoint cycles C_x and C_y such that $x \in V(C_x)$ and $y \in V(C_y)$. This follows from the fact that, by Moon's theorem, every strongly connected tournament is vertex-pancyclic. Hence C_x and C_y exist if and only if T contains disjoint 3-cycles, one containing x and the other y. It follows from this that every 4-strong tournament contains cycles C_x, C_y as above. Bang-Jensen, Guo and Yeo proved that this already holds for 3-strong tournaments and an infinite family of 2-strong counter-examples was given [83]. Hence $g(1,1) = 3$.

13.12.4 Decomposing Digraphs under Degree Constraints

Stiebitz proved [834] that the vertex set of every undirected graph of minimum degree $(s+t+1)$ can be decomposed into two sets which induce graphs of minimum degree at least s and t, respectively. He also posed the following problem at the Midsummer Combinatorial Workshop in Prague in 1995.

Problem 13.12.9 *[833] Does there exist a function $f(s,t)$ such that every digraph $D = (V, A)$ with $\delta^+(D) \geq f(s,t)$ contains two digraphs $D_1 = (V_1, A_1)$ and $D_2 = (V_2, A_2)$ with $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$ such that $\delta^+(D_1) \geq s$ and $\delta^+(D_2) \geq t$?*

The only known value of f for positive s, t is $f(1,1) = 3$. The second power of an odd cycle shows that $f(1,1) > 2$. To show that $f(1,1) \leq 3$ we use the result of Thomassen that every digraph of minimum out-degree 3 has a pair of vertex-disjoint cycles (see Exercise 13.4). Let D have $\delta^+(D) \geq 3$ and let X_1, X_2 be vertex sets of two disjoint cycles in D. Now define V_1 to be X_1 plus all those vertices of $V - \{X_1 \cup X_2\}$ that can reach X_1 by a directed path in $D\langle V - X_2\rangle$ and $V_2 = V - V_1$. It is easy to check that this gives the desired partition.

Problem 13.12.9 seems very hard and already the existence of $f(2,1)$ is open. See also the paper [17] where Alon poses the problem whether $f(2,2)$ exists.

A theorem of Lovász [651] asserts that if G is an undirected graph with maximum degree Δ and $d_1 \geq d_2 \geq \ldots \geq d_k$ are non-negative integers such that $d_1 + d_2 + \ldots + d_k + k - 1 = \Delta$, then the vertex set of G can be partitioned into $V_1, V_2, \ldots, V_k$ so that the subgraph induced by V_i has maximum degree at most d_i. For $k = 2$ this result implies that vertex set of every graph of maximum degree $2d + 1$ can be partitioned into two sets, none of which induces a graph with maximum degree more than d.

The following argument by Alon [17] shows that no such result can hold for directed graphs: By a result of Thomassen (see also Section 8.3), for every k there exists a digraph $D_k = (V, A)$ with $d^+(v) = k$ for every vertex $v \in V$ and D_k has no even cycle. Now consider an arbitrary partition of V into non-empty sets V_1, V_2. We claim that some vertex in V_i has all its out-neighbours inside V_i for $i = 1$ or 2. Suppose this is not the case and let v_1 be an arbitrary vertex in V_1. Then v_1 has an out-neighbour $v_2 \in V_2$ which in turn has an out-neighbour $v_3 \in V_1$ and so on. Since $|V|$ is finite we must reach a point when $v_i = v_j$ and now $v_i v_{i+1} \ldots v_{j-1} v_i$ is an even cycle in D_k, a contradiction.

Alon also showed that if we allow partitions into three or more sets, then one can always reduce the maximum degree of the resulting digraphs.

Theorem 13.12.10 *[19] Let $d_1, d_2, \ldots, d_k, \Delta$, $k \geq 3$, be non-negative integers such that $d_1 + d_2 + \ldots d_k + k - 1 = 2\Delta$. Then the vertex set of any directed graph D with $\Delta^+(D) = \Delta$ can be partitioned into k subsets $V_1, V_2, \ldots, V_k$ such that $\Delta^+(D\langle V_i \rangle) \leq d_i$ for all $i \in [k]$.* □

13.13 Exercises

13.1. Construct a strong tournament on 5 vertices such that one must delete at least 3 arcs to obtain an acyclic digraph.

13.2. (+) Let T be the tournament on 15 vertices described in Section 13.2. Prove that every feedback arc set of T has at least 39 arcs.

13.3. For every $n \geq 3$, construct a digraph of minimum out-degree 2 not having two disjoint cycles.

13.4. (+) **Minimum out-degree three implies two disjoint cycles.** Prove that every digraph D with $\delta^+(D) \geq 3$ has a pair of vertex-disjoint cycles. Hint: use Lemma 13.3.8 (Thomassen [858]).

13.5. Prove Corollary 13.3.5 using Theorem 13.3.4. Hint: first observe that every digraph D with $\delta^+(D) \geq k$ has at least $\frac{k}{64}$ vertex-disjoint cycles. Remove the arcs of these and continue recursively.

13.6. (−) Prove Theorem 13.4.2. Hint: use Exercise 13.17.

13.7. Prove Lemma 13.5.1. Hint: start by extracting paths by going backwards from leaves.

13.8. **Scheduling airplanes.** An airport has a certain number of runways that can be used for landing of airplanes. How would you schedule airplanes to use the minimum number of the runways (in order to possibly have some spare ones permanently ready for emergency landings) if every use of a runway can be determined as a fixed time interval ?

13.9. (−) Show by examples that properties (1) and (2) of Lemma 13.5.9 need not hold for arbitrary acyclic digraphs.

13.10. **Non-hamiltonian 2-strong digraphs with** $\alpha = 2$. Construct such a digraph. Hint take two 2-strong tournaments on five vertices and join them appropriately by arcs [156].
13.11. Prove Lemma 13.6.3. Hint: use the proof of Theorem 6.7.1.
13.12. Prove Lemma 13.6.5.
13.13. Complete the proof of Theorem 13.8.1.
13.14. **Heaviest cycle subdigraphs in digraphs.** Describe an $O(n^3)$ algorithm to find, in a digraph with non-negative weights on the arcs, a cycle subdigraph of maximum weight. Hint: use the same approach as in the proof of Theorem 13.8.1.
13.15. **Finding a minimum cycle factor in extended semicomplete and semicomplete bipartite digraphs**. Describe polynomial algorithms for the minimum cycle factor problem in the case of digraphs from one of these classes.
13.16. (+) Prove Theorem 13.8.3 directly from Theorem 4.8.2. Show that your proof implies the existence of an algorithm, which given a k-strong digraph D and a subset $X \subseteq V(D)$, either finds a collection of disjoint cycles covering all the vertices of X, or an independent set $X' \subseteq X$ of size more than k.
13.17. **Every regular directed multigraph has a cycle factor.** Prove this claim.
13.18. Prove that every 2-arc-strong semicomplete digraph H has three distinct vertices q_1, q_2, q_3 such that $H - q_i$ is strong for $i = 1, 2, 3$.
13.19. (−) Prove that every regular tournament is arc-3-cyclic. Show that this is not always true for regular semicomplete digraphs.
13.20. (+) Prove Lemma 13.10.6. Hint: the cases $k = 2, 3, 5$ can be handled separately. Apply Theorem 13.4.3 to get, for each $k \neq 2, 3, 5$, a decomposition of K^*_{k+1} into arc-disjoint hamiltonian paths $P_1, P_2, \ldots, P_k$ such that P_i starts in x_i and ends in $x_{\pi(i)}$ for all $i \in [k]$ where the vertex set of $\overleftrightarrow{K}_k$ is $\{x_1, x_2, \ldots, x_k\}$ and π is a permutation of $[k]$. Show that these paths can be extended to the desired spanning strong subdigraphs.
13.21. Derive Lemma 13.11.5 from Theorem 6.6.8.
13.22. Use Lemma 13.11.5 to prove Lemma 13.11.6.
13.23. Recall that $\mathrm{lc}(D)$ denotes the length of a longest cycle in D. Prove the following result:

Proposition 13.13.1 *[114] For every digraph D and every choice of positive integers, ℓ_1, ℓ_2, such that $\mathrm{lp}(D) = \ell_1 + \ell_2$, there exists a partition of D into two digraphs, D_1 and D_2, such that $\mathrm{lc}(D_i) \leq \ell_i$, for $i = 1, 2$.*

13.24. Prove Corollary 13.11.8.
13.25. **Finding disjoint cycles is hard.** Prove that the following problem is $\mathcal{NP}$-complete. Given a digraph D and an integer k, decide whether D has at least k disjoint cycles. Hint: use a reduction from the 3-dimensional matching problem. (Given three sets X^1, X^2, X^3 of the same cardinality n and a subset R of $X^1 \times X^2 \times X^3$, decide whether the elements of every X^i can be labelled $x^i_1, x^i_2, \ldots, x^i_n$ so that $(x^1_j, x^2_j, x^3_j) \in R$ for each $j \in [n]$. This problem is $\mathcal{NP}$-complete, see Gary and Johnson [393].) In the reduction you may utilize the gadget L given in Figure 13.5. We start from the digraph G on vertices $X^1 \cup X^2 \cup X^3$ and with no arcs. For each $(x, y, z) \in R$, we add L to G. Prove that the resulting digraph has $n + 2|R|$ cycles (all of which are 3-cycles) if and only if there exists the required labelling of the elements in X^1, X^2 and X^3 (A. Yeo, personal communication).
13.26. Prove Theorem 13.4.10.

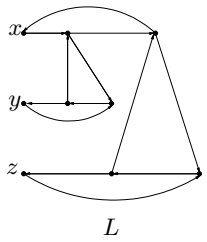


Figure 13.5 The gadget for Exercise 13.25.

13.27. Prove Lemma 13.5.9. Hint: the only acyclic extended semicomplete digraphs are extensions of transitive tournaments.
13.28. Show that Theorem 13.4.1 follows from Theorem 1.7.2.
13.29. Prove that the arcs of $\overleftrightarrow{K}_6$ cannot be decomposed into 5 hamiltonian cycles.
13.30. Show that there is only one 2-strong tournament on 7 vertices which has no 2-cycle factor.
13.31. Prove that the digraph in Figure 13.2 has no hamiltonian path.
13.32. Show that if $v_1, v_2, \ldots, v_n$ is an ordering of a coherent cyclic order and there is an arc between v_i and v_{i+1}, then $v_i \rightarrow v_{i+1}$.
13.33. Show by an example that the cyclic independence number of a digraph may depend on which cyclic order $\mathcal{O}$ we choose.
13.34. Describe a linear algorithm which, given a digraph $D = (V, A)$ an ordering E of V and an arc $a \in A$, decides whether $i_E(a) > 1$. Hint: how can you find an equivalent order E', so that a will be a backward arc in E'?
13.35. Describe a polynomial algorithm which, given a digraph $D = (V, A)$ an ordering E of V and an arc $a \in A$, finds the index $i_E(a)$ of the arcs a with respect to the ordering E.
13.36. **(+) Finding a maximum cyclic independent set in polynomial time**. Show how to find, given a strong digraph D and a coherent cyclic order $\mathcal{O}$ of D, a cyclic independent set (with respect to $\mathcal{O}$) of maximum size). Hint: consider what happens if you start the proof of Theorem 13.9.6 from a set S which is cyclic independent with respect to $\mathcal{O}$ but $|S| < \alpha_{\mathcal{O}}(D)$. Argue that in the resulting transitive digraph you can find, in polynomial time, an independent set of size larger than $|S|$ and from that obtain a new larger cyclic independent with respect to $\mathcal{O}$.
13.37. Let D_r be the digraph which is defined at the end of Subsection 13.9. Show that every strong spanning subdigraph of D_r has cyclomatic number at least $2r - 1$. Next show that every cyclic spanning subdigraph of D_r with cyclomatic number r is an r-cycle factor in which all cycles are 4-cycles.
13.38. Let q_i, $i = 1, 2, \ldots, k+1$, be defined as in (13.9). Prove that $q_i < 16k(i-1)$ for all $i \in [k+1]$.
13.39. **Covering a digraph of bounded maximum out-degree by few one-way cuts.** Prove the following where $c = c(n)$ is the function in Theorem 13.12.2.

Proposition 13.13.2 *[20] Let D be a digraph with $\delta^+(D) \leq k$. Then the arcs of D can be covered by at most $c(2k+1)$ one-way cuts.*

Hint: use Lemma 11.3.3 to bound the chromatic number of D by $2k+1$ and show how to extend a covering of an arc-weighted copy of $\overleftrightarrow{K}_{2k+1}$ by r cuts to a covering of D by the same number of cuts.

14. Increasing Connectivity

In this chapter we discuss the important problem of increasing the (arc)-strong connectivity of a given directed (multi)graph by a number of different operations. These include adding new arcs, reversing existing arcs and de-orienting arcs. In Section 14.1 we introduce the operation of splitting off a pair of arcs incident with a vertex. We prove Mader's splitting theorem which allows one to give inductive proofs for several important results on directed multigraphs. In Section 14.2, using Mader's theorem, we describe a solution, due to Frank, for the problem of finding a minimum cardinality set of new arcs to add to a directed multigraph such that the result is a k-arc-strong directed multigraph. In Section 14.3 we describe a solution by Frank and Jordán of the analogous problem for vertex-strong connectivity.

Another way of increasing the arc/vertex-strong connectivity of a directed multigraph is by reversing the orientation of certain arcs. In Sections 14.4 and 14.5 we discuss this approach. In Section 14.6 we consider the variation of the augmentation problem where parallel arcs are not allowed and a new arc can only be added if the opposite arc is already present. This is equivalent to deorienting (replacing the arc by a 2-cycle) some of the arcs that are not contained in a 2-cycle. Finally we briefly discuss a number of miscellaneous results such as increasing the arc-strong connectivity of a bipartite directed multigraph while preserving bipartiteness and increasing the arc-strong connectivity of a directed hypergraph.

14.1 The Splitting Off Operation

In Frank's proof of Menger's theorem in Section 5.4, we saw how one could apply the idea of replacing two arcs incident to some vertex by one new arc and then apply induction. In this section we shall see yet another indication that this type of operation can be very useful. We consider a directed multigraph D with a special vertex s. We always assume that

$$d_D^+(s) = d_D^-(s). \tag{14.1}$$

To emphasize that s is a special vertex we specify D as $D = (V + s, A)$ or $D = (V + s, E \cup F)$ where F is the set of arcs with one end-vertex in s

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_14,

($s \notin V$ and $E \cap F = \emptyset$). Furthermore we will assume that the local arc-strong connectivity between every pair x, y of vertices in V is at least k. By Menger's theorem this is equivalent to

$$d^+(U), d^-(U) \geq k \quad \text{for all} \quad \emptyset \neq U \subset V. \tag{14.2}$$

Whenever a digraph $D = (V + s, A)$ satisfies (14.2) for some k we say that D is ***k*-arc-strong in** V.

We consider the operation of replacing a pair (us, sv) of arcs incident with s by one new arc uv. The operation of performing this replacement is called **splitting off** or just **splitting** the pair (us, sv) and the resulting directed multigraph is denoted by D_{uv}. The splitting of a pair (us, sv) is **admissible** if (14.2) holds in D_{uv}. If this is the case, we will also say that the pair (us, sv) is an **admissible pair** (or an **admissible splitting**).

Recall the definition of a k-(in/out)-critical set of vertices from Section 5.6. The following useful lemma is due to Frank:

Lemma 14.1.1 *[342] If X and Y are intersecting k-critical sets, then one of the following holds:*

(a) $X \cup Y$ is k-critical,
(b) $Y - X$ is k-critical and $d(X \cap Y, V + s - (X \cup Y)) = 0$.

Proof: We consider three cases:
Case 1: $X \cup Y \neq V$ and X, Y are either both k-out-critical or both k-in-critical.

Assume that X, Y are both k-out-critical. It follows from (14.2) that $d^+(X \cup Y), d^+(X \cap Y) \geq k$. Using the submodularity of d^+_D, we obtain

$$\begin{aligned} k + k &= d^+(X) + d^+(Y) \\ &\geq d^+(X \cup Y) + d^+(X \cap Y) \\ &\geq k + k, \end{aligned} \tag{14.3}$$

and from this we get that $X \cup Y$ is k-critical and hence (a) holds. The same conclusion is reached if X, Y are both k-in-critical.
Case 2: $X \cup Y = V$ and X, Y are either both k-out-critical or both k-in-critical.

We will assume that X, Y are both k-out-critical, the proof is analogous in the other case. Let $S = V + s - X$ and $T = V + s - Y$. Then $d^-(S) = d^-(T) = k$ and $S \cap T = \{s\}$. Since $S - T = Y - X$ and $T - S = X - Y$, we get from (14.2) that $d^-(S - T), d^-(T - S) \geq k$. Since $d^-(s) = d^+(s)$, we can apply (5.3) and obtain

$$\begin{aligned} k + k &= d^-(S) + d^-(T) \\ &= d^-(S - T) + d^-(T - S) + d(S \cap T, V + s - (S \cup T)) \\ &\geq k + k + d(V - S, T), \end{aligned} \tag{14.4}$$

implying that $Y-X = S-T$ is k-in-critical and that $d(S\cap T, V+s-(S\cup T)) = 0$. Since $X \cap Y = V + s - (S \cup T)$ and $V + s - (X \cup Y) = \{s\} = S \cap T$ we also see that $d(X \cap Y, V + s - (X \cup Y)) = 0$. Thus (b) holds.

Case 3: One of X, Y is k-in-critical and the other is k-out-critical.

We consider the case when X is k-in-critical and Y is k-out-critical, the other case is analogous. Let $Z = V+s-X$. Then we have $d^+(Y) = d^+(Z) = k$, $Y\cap Z = Y-X$ and $Y\cup Z = V+s-(X-Y)$. Hence $d^+(Y\cap Z) = d^+(Y-X) \geq k$ and $d^+(Y \cup Z) = d^-(V + s - (Y \cup Z)) = d^-(X - Y) \geq k$. Now we can apply (5.2) and we get

$$\begin{aligned} k + k &= d^+(Y) + d^+(Z) \\ &= d^+(Y \cap Z) + d^+(Y \cup Z) + d(Y, Z) \\ &\geq k + k + d(Y, Z), \end{aligned} \tag{14.5}$$

implying that $d^+(Y - X) = d^+(Y \cap Z) = k$ and that $d(Y, Z) = 0$. Since $Z - Y = V + s - (X \cup Y)$ and $Y - Z = X \cap Y$, the last equality shows that $d(X \cap Y, V + s - (X \cup Y)) = 0$. Thus (b) holds. □

We are now ready to prove the following important result by Mader.

Theorem 14.1.2 (Mader's directed splitting theorem) *[669] Suppose that $D = (V + s, E \cup F)$ satisfies (14.2) and that $d^+(s) = d^-(s)$. Then for every arc sv there is an arc us such that the pair (us, sv) is an admissible splitting.*

Proof: The proof we give is due to Frank [342]. First note that a pair (us, sv) can be split off preserving (14.2) if and only if there is no k-critical set which contains both u and v. Hence if there is no k-critical set containing v, then we are done. If X and Y are intersecting k-critical sets containing v, then only alternative (i) can hold in Lemma 14.1.1, because the existence of the arc sv implies that $d(V + s - (X \cup Y), X \cap Y) \geq 1$. Hence the union T of all k-critical sets containing v is also k-critical. If we can find an in-neighbour u of s in $V - T$, then we are done, since by the choice of T, there is no k-critical set which contains u and v. So suppose that all in-neighbours of s are in T. If T is k-out-critical, then

$$\begin{aligned} d^-(V - T) &= d^+(T) - d^+(T, s) + d^+(s, V - T) \\ &\leq k - (d^-(s) - d^+(s) + 1) \\ &= k - 1, \end{aligned}$$

since s has no in-neighbour in $V - T$ and sv is an arc from s to T (we also used $d^-(s) = d^+(s)$). This contradicts (14.2) so we cannot have that T is k-out-critical. But if T is k-in-critical, then

$$\begin{aligned} d^+(V - T) = d^-(T + s) &= d^-(T) - d^+(s, T) + d^+(V - T, s) \\ &\leq k - 1 + 0 < k, \end{aligned}$$

a contradiction again. Hence we have shown that (us, sv) is an admissible pair and the proof is complete. □

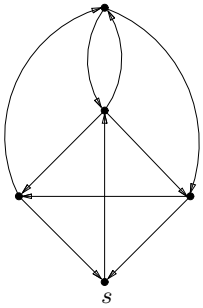


Figure 14.1 A digraph $D = (V + s, A)$ which is 2-arc-strong in V and has no admissible splitting at s. Note that $d^-(s) = 2 \neq 1 = d^+(s)$.

Note that the assumption that $d^-(s) = d^+(s)$ in Theorem 14.1.2 cannot be removed. Figure 14.1 shows an example of a digraph $D = (V + s, A)$ with no admissible splitting at s.

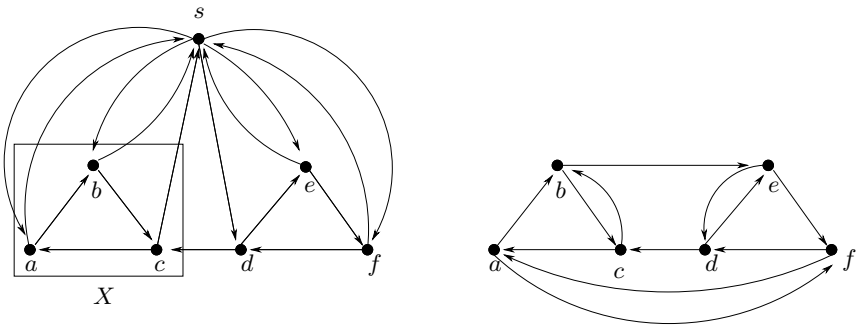


Figure 14.2 A digraph $D = (V + s, A)$ which is 2-arc-strong in V. A complete splitting of the arcs is shown in the right figure after removal of s. The set X shows that we cannot split off both of the pairs $(as, sb), (cs, sa)$, since that would leave X with out-degree one.

Corollary 14.1.3 *Suppose that $D = (V + s, E + F)$ satisfies (14.2) and that $d^+(s) = d^-(s)$. Then there exists a pairing $((u_1s, sv_1), \ldots, (u_rs, sv_r))$, $r = d^-(s)$, of the arcs entering s with the arcs leaving s such that replacing all arcs incident with s by the arcs $u_1v_1, \ldots, u_rv_r$ and then deleting s, we obtain a k-arc-strong directed multigraph D'.* □

See Figure 14.2 for an example of a complete splitting in a digraph.

Frank and Jackson showed that for eulerian directed multigraphs one can get a stronger result. Namely, it is possible to split off all arcs incident with the special vertex s in such a way that all local arc-strong connectivities within V are preserved.

Theorem 14.1.4 *[341, 556] Let $D = (V + s, A)$ be an eulerian directed multigraph. Then for every arc $us \in A$ there exists an arc $sv \in A$ such that $\lambda_{D_{uv}}(x, y) = \lambda_D(x, y)$ for all $x, y \in V$.* □

A similar result concerning local connectivity preserving splittings holds for general undirected graphs. This very powerful result was proved by Mader [668]. Such a similarity between eulerian digraphs and general undirected graphs with respect to certain properties seems to be quite common. To say it popularly: Eulerian digraphs often behave like undirected graphs. For another example of this phenomenon see Section 10.7.2.

Bang-Jensen, Frank and Jackson showed that it is possible to give a common generalization of Theorem 14.1.4 and Mader's directed splitting theorem (Theorem 14.1.2) to mixed graphs. Since the statement of this result is rather technical, we refer the interested reader to the paper [78].

It was pointed out by Enni in [295] that Theorem 14.1.4 cannot be extended to arbitrary digraphs, not even if one only wants to preserve the minimum of $\lambda(x, y)$ and $\lambda(y, x)$. For two other generalizations of Theorem 14.1.2 see the papers [836] by Su and [372] by Gabow and Jordán.

14.2 Increasing the Arc-Strong Connectivity Optimally

We will consider the ARC-CONNECTIVITY AUGMENTATION PROBLEM: Given a directed multigraph $D = (V, E)$ and a natural number k such that D is not k-arc-strong; find a minimum cardinality set of new arcs F to add to D such that the resulting directed multigraph $D' = (V, E \cup F)$ is k-arc-strong. This D' is called an **optimal augmentation** of D. We will present a solution to this problem due to Frank [342]. Frank solved the problem by supplying a min-max formula for the minimum number of new arcs as well as a polynomial algorithm to find such a minimum set of new arcs. First let us make the simple observation that such a set F indeed exists, since we may just add k parallel arcs in both directions between a fixed vertex $v \in V$ and all other vertices in V (it is easy to see that the resulting directed multigraph will be k-arc-strong).

Definition 14.2.1 *Let $D = (V, A)$ be a directed multigraph. Then $\gamma_k(D)$ is the smallest integer γ such that*

$$\sum_{X_i \in \mathcal{F}} (k - d^-(X_i)) \leq \gamma \text{ and}$$
$$\sum_{X_i \in \mathcal{F}} (k - d^+(X_i)) \leq \gamma,$$

for every subpartition $\mathcal{F} = \{X_1, \ldots, X_t\}$ *of* V *with* $\emptyset \subset X_i \subset V$, $\forall\, i \in [t]$.

We call $\gamma_k(D)$ the **subpartition lower bound for arc-strong connectivity**. By Menger's theorem, D is k-arc-strong if and only if $\gamma_k(D) \leq 0$. Indeed, if D is k-arc-strong, then $d^+(X), d^-(X) \geq k$ holds for all proper non-empty subsets of V and hence we see that $\gamma_k(D) \leq 0$. Conversely, if D is not k-arc-strong, then let X be a set with $d^-(X) < k$. Take $\mathcal{F} = \{X\}$, then we see that $\gamma_k(D) \geq k - d^-(X) > 0$.

Lemma 14.2.2 *[342] Let* $D = (V, A)$ *be a directed multigraph and let* k *be a positive integer such that* $\gamma_k(D) > 0$. *Then* D *can be extended to a new directed multigraph* $D' = (V + s, A \cup F)$, *where* F *consists of* $\gamma_k(D)$ *arcs whose head is* s *and* $\gamma_k(D)$ *arcs of whose tail is* s *such that (14.2) holds in* D'.

Proof: We will show that, starting from D, it is possible to add $\gamma_k(D)$ arcs from V to s so that the resulting graph satisfies

$$d^+(X) \geq k \text{ for all } \emptyset \neq X \subset V. \tag{14.6}$$

Then it will follow analogously (by considering the converse of D) that it is also possible to add $\gamma_k(D)$ new arcs from s to V so that the resulting graph satisfies

$$d^-(X) \geq k \text{ for all } \emptyset \neq X \subset V. \tag{14.7}$$

First add k parallel arcs from v to s for every $v \in V$. This will certainly make the resulting directed multigraph satisfy (14.6). Now delete as many new arcs as possible until removing any further arc would result in a digraph where (14.6) no longer holds (that is, every remaining new arc vs leaves a k-out-critical set). Let $\tilde{D}$ denote the current directed multigraph after this deletion phase and let S be the set of vertices v which still have an arc to s in $\tilde{D}$. Let $\mathcal{F} = \{X_1, \ldots, X_r\}$ be a family of k-out-critical sets such that every $v \in S$ is contained in some member X_i of $\mathcal{F}$ and assume that $\mathcal{F}$ has as few members as possible with respect to this property. Clearly this choice implies that either $\mathcal{F}$ is a subpartition of V, or there is a pair of intersecting sets X_i, X_j in $\mathcal{F}$.

Case 1: $\mathcal{F}$ is a subpartition of V.

Then we have

$$\begin{aligned} kr &= \sum_{i=1}^{r} d^+_{\tilde{D}}(X_i) \\ &= \sum_{i=1}^{r} (d^+_D(X_i) + d^+_{\tilde{D}}(X_i, s)) \\ &= \sum_{i=1}^{r} d^+_D(X_i) + d^-_{\tilde{D}}(s), \end{aligned}$$

implying that $d^-_{\tilde{D}}(s) = \sum_{i=1}^{r}(k - d^+_D(X_i)) \leq \gamma_k(D)$, by the definition of $\gamma_k(D)$.

Case 2: Some pair $X_i, X_j \in \mathcal{F}$ is intersecting.

If X_i, X_j are crossing, then the submodularity of $d^+_{\tilde{D}}$ and (14.2) imply that $X_i \cup X_j$ is also k-out-critical and hence we could replace the two sets X_i, X_j by the set $X_i \cup X_j$ in $\mathcal{F}$, contradicting the choice of $\mathcal{F}$. Hence we must have $X_i \cup X_j = V$ and $\mathcal{F} = \{X_1, X_2\}$, where without loss of generality $i = 1, j = 2$. Let $X = V - X_1 = X_2 - X_1$ and $Y = V - X_2 = X_1 - X_2$. Then $d^-_D(X) = d^+_D(X_1)$ and $d^-_D(Y) = d^+_D(X_2)$ and hence we get

$$\begin{aligned} \gamma_k(D) &\geq (k - d^-_D(X)) + (k - d^-_D(Y)) \\ &= k - d^+_D(X_1) + k - d^+_D(X_2) \\ &\geq k - d^+_{\tilde{D}}(X_1) + k - d^+_{\tilde{D}}(X_2) + d^-_{\tilde{D}}(s) \\ &= d^-_{\tilde{D}}(s), \end{aligned}$$

since X_1, X_2 are k-out-critical in $\tilde{D}$. Thus $d^-_{\tilde{D}}(s) \leq \gamma_k(D)$ as claimed. □

Theorem 14.2.3 (Frank's arc-connectivity augmentation theorem) *[342] Let $D = (V, A)$ be a directed multigraph and k a positive integer such that $\gamma_k(D) > 0$. The minimum number of new arcs that must be added to D in order to give a k-arc-strong directed multigraph $D' = (V, A \cup F)$ equals $\gamma_k(D)$.*

Proof: To see that we must use at least $\gamma_k(D)$ arcs, it suffices to observe that if X and Y are disjoint sets, then no new arc can increase the out-degree (in-degree) of both sets. Hence a subpartition $\mathcal{F}$ realizing the value of γ_k in Definition 14.2.1 is a certificate that we must use at least $\gamma_k(D)$ new arcs.

To prove the other direction, we use Mader's splitting theorem and Lemma 14.2.2. According to this lemma we can extend D to a new directed multigraph $\tilde{D}$ by adding a new vertex s and $\gamma_k(D)$ arcs from V to s and from s to V. Note that we may not need $\gamma_k(D)$ arcs in both directions, but we will need it in one of the directions by our remark in the beginning of the proof.

In the case where fewer arcs are needed, say from V to s, we add arbitrary arcs from V to s so that the resulting number becomes $\gamma_k(D)$.

Now it follows from Corollary 14.1.3 that all arcs incident with s can be split off without violating (14.2). This means that if we remove s, then the resulting directed multigraph D' is k-arc-strong. □

See Figure 14.3 for an example illustrating the theorem.

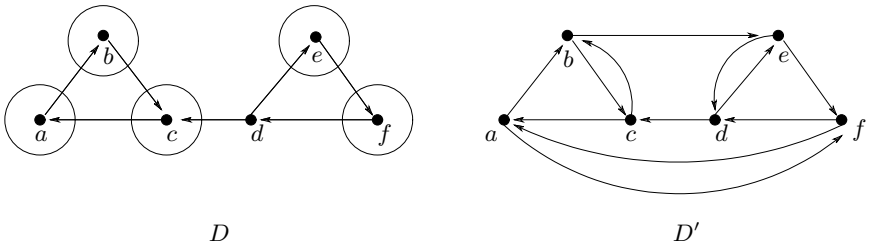


Figure 14.3 A digraph D with $\gamma_2(D) = 5$. The big circles indicate a subpartition which realizes $\gamma_2(D)$. The right part of the figure shows an optimal 2-arc-strong augmentation D' of D obtained by adding 5 new arcs. Compare this with Figure 14.2. Here the digraph in the right part is the same as the augmented digraph D'.

The reader may have noticed that in the proof of Lemma 14.2.2, we never specified exactly how we obtained the minimal set of arcs from V to s so that (14.6) held. The proof is valid for every such set of arcs that is minimal with respect to deletion of arcs. This means in particular that we can use a greedy approach to find such a set of arcs starting from the configuration with k parallel arcs from every vertex $v \in V$ to s. This gives rise to the following algorithm, by Frank [342], for augmenting the arc-strong connectivity optimally to k for any directed multigraph D which is not already k-arc-strong:

Frank's arc-strong connectivity augmentation algorithm
Input: A directed multigraph $D = (V, A)$ and a natural number k such that $\gamma_k(D) > 0$.
Output: A k-arc-strong optimal augmentation D^* of D.

1. Let $v_1, v_2, \ldots, v_n$ be a fixed ordering of V and let s be a new vertex.
2. For each $i \in [n]$, add k parallel arcs from v_i to s and from s to v_i.
3. Starting from $i := 1$, remove as many arcs from v_i to s as possible without violating (14.6). If $i < n$, then let $i := i + 1$ and repeat this step;
   Let γ^- denote the number of remaining arcs from V to s in the resulting digraph.
4. Starting from $i := 1$, remove as many arcs from s to v_i as possible without violating (14.7). If $i < n$, then $i := i + 1$ and repeat this step;

Let γ^+ denote the number of remaining arcs from s to V in the resulting directed multigraph.

5. Let $\gamma = \max\{\gamma^-, \gamma^+\}$. If $\gamma^- < \gamma^+$, then add $\gamma^+ - \gamma^-$ arcs from v_1 to s; if $\gamma^+ < \gamma^-$, then add $\gamma^- - \gamma^+$ arcs from s to v_1.
6. Let D' denote the current directed multigraph. In D' we have $d^-_{D'}(s) = d^+_{D'}(s)$ and (14.2) holds. Split off all arcs incident with s in D' by applying Theorem 14.1.2 γ times. Let D^* denote the resulting directed multigraph.
7. Return D^*.

Using flows, this algorithm can be implemented as a polynomial algorithm for augmenting the arc-strong connectivity of a given directed multigraph [342]. See Exercises 14.1 and 14.3.

Frank [342] pointed out that his algorithm also works for the following problem called the ARC-STRONG CONNECTIVITY AUGMENTATION PROBLEM WITH VERTEX-WEIGHTS. Here there are two weights $c_{in}(v), c_{out}(v)$ on every vertex v and the cost of adding an arc from u to v is equal to $c_{out}(u)+c_{in}(v)$. The only change needed in the algorithm above is that now the ordering of the vertices should be so that $c_{out}(v_1) \geq c_{out}(v_2) \geq \ldots \geq c_{out}(v_n)$ when we delete arcs into s and we use another ordering (if necessary) with the same property with respect to c_{in} when we delete arcs out of s. The reason why this greedy approach works is outlined in [342] and comes from the fact that a certain polymatroidal structure is present [342, 357].

If instead we allow weights on the arcs and ask for a minimum weight (rather than just minimum cardinality) set of new arcs to add to D in order to obtain a k-arc-strong directed multigraph D', then we have the WEIGHTED ARC-STRONG CONNECTIVITY AUGMENTATION PROBLEM.

Theorem 14.2.4 *The weighted arc-strong connectivity augmentation problem is $\mathcal{NP}$-hard.*

Proof: Let $D = (V, A)$ be a digraph on n vertices $V = [n]$. Define weights $c(ij)$ on the arcs of the complete digraph $\overleftrightarrow{K}_n$ with vertex set V as follows:

$$c(ij) = \begin{cases} 1 & \text{if } ij \in A \\ 2 & \text{if } ij \notin A. \end{cases} \tag{14.8}$$

Let $D_0 = (V, \emptyset)$ (that is, the digraph on V with no arcs). Since every vertex of a strong digraph is the tail of at least one arc, we need at least n arcs to make D_0 strong. Now it is easy to see that D_0 can be made strongly connected using arcs with total weight at most n if and only if D has a Hamilton cycle. Thus we have reduced the $\mathcal{NP}$-hard Hamilton cycle problem to the weighted arc-strong connectivity augmentation problem. Clearly our reduction can be carried out in polynomial time. □

We complete this section with an interesting result by Cheng and Jordán. It implies that the so-called **successive augmentation property** holds for arc-strong connectivity.

Theorem 14.2.5 *[206] Let D be a directed multigraph with $\lambda(D) = \ell$. Then there exists an infinite sequence $D = D_0, D_1, D_2, \ldots$ of directed multigraphs such that, for every $i \geq 0$, $A(D_i) \subset A(D_{i+1})$, $V(D_i) = V(D)$ and D_i is an optimal $(\ell + i)$-arc-strong augmentation of D.* □

It is shown by an example in [206] that a similar property does not hold for the vertex-strong connectivity augmentation problem which we consider below.

14.3 Increasing the Vertex-Strong Connectivity Optimally

We now turn to the VERTEX-STRONG CONNECTIVITY AUGMENTATION PROBLEM: given a digraph $D = (V, A)$ on at least $k + 1$ vertices, find a smallest set F of new arcs for which $D' = (V, A \cup F)$ is k-strong.

When it comes to studying vertex-strong connectivity, multiple arcs play no role and hence we shall always consider digraphs (knowing that our results extend to directed multigraphs). In particular, in this section we have $d^+_D(v) = |N^+_D(v)|$ for every vertex v.

Let us first observe that, even if we do not allow multiple arcs, we cannot bound the number of arcs we need to add to make a digraph D k-strong by some function of $\gamma_k(D)$ (recall Definition 14.2.1). To see this, it suffices to note that there are k-arc-strong digraphs which are not k-strong and one can construct such digraphs where the number of new arcs one needs to add in order to obtain a k-strong superdigraph is arbitrarily high (see Exercise 14.6).

Suppose X is a non-empty set of vertices in a digraph D such that $N^+[X] \neq V$ and $|N^+(X)| < k$ (recall that $N^+[X] = X \cup N^+(X)$). Then it follows from Menger's theorem that D is not k-strong because the set $N^+(X)$ separates every vertex in X from every vertex in $V - N^+[X]$. Furthermore, in order to obtain a k-strong digraph by adding arcs to D we must add at least $k - |N^+(X)|$ new arcs with tail in X and head in $V - N^+(X)$.

Similarly to the definition of $\gamma_k(D)$ in Definition 14.2.1, we can define $\gamma^*_k(D)$ as follows:

Definition 14.3.1 *Let $D = (V, A)$ be a directed graph. Then $\gamma^*_k(D)$ is the smallest integer γ such that*

$$\sum_{X \in \mathcal{F}^-} (k - |N^-(X)|) \leq \gamma \text{ and}$$
$$\sum_{X \in \mathcal{F}^+} (k - |N^+(X)|) \leq \gamma,$$

for every choice of subpartitions $\mathcal{F}^-, \mathcal{F}^+$ *of* V *with the property that every* $X \in \mathcal{F}^-$ *satisfies* $N^-[X] \neq V$ *and every* $X \in \mathcal{F}^+$ *satisfies* $N^+[X] \neq V$.

As with arc-strong connectivity it is not hard to see that $\gamma_k^*(D)$ is a lower bound for the number of new arcs we must add to D to obtain a k-strong digraph. This follows from the fact that the sets in $\mathcal{F}^-$ are disjoint and hence no new arc can increase the in-neighbourhoods (out-neighbourhoods) of two sets from $\mathcal{F}^-$ ($\mathcal{F}^+$). We call the number $\gamma_k^*(D)$ the **subpartition lower bound for vertex-strong connectivity**.

The ***k*-strong augmentation number** of D, denoted $a_k(D)$, is the minimum number of new arcs that must be added to a digraph $D = (V, A)$ in order to obtain a k-strong digraph. It is easy to see that $a_k(D)$ is well-defined provided that D has at least $k + 1$ vertices.

14.3.1 One-Way Pairs

First we point out that for vertex-strong connectivity augmentation, the subpartition lower bound is no longer sufficient, that is, it may not be possible to make D k-strong by adding $\gamma_k^*(D)$ arcs. An example illustrating this is given in Figure 14.4(a). Here $k = 2$ and it is not difficult to check that $\gamma_k^*(D) = 2$. However, it is not possible to make D 2-strong by adding just two new arcs. In order to explain this, we need a few new definitions due to Frank and Jordán [354]. Let X, Y be disjoint non-empty proper subsets of V. The ordered pair (X, Y) is a **one-way pair** in $D = (V, A)$ if D has no arc with tail in X and head in Y (that is, $Y \Rightarrow X$). For such a pair (X, Y) we refer to X (Y) as the **tail** (**head**) of the pair. Let $h(X, Y) = |V - X - Y|$. The **deficiency** of a one-way pair (X, Y) with respect to k-strong connectivity is

$$\eta_k(X, Y) = \max\{0, k - h(X, Y)\}. \tag{14.9}$$

For instance, if $N^+[X] \neq V$, then the pair $(X, V - N^+[X])$ is a one-way pair with deficiency $\eta_k(X, V - N^+[X]) = \max\{0, k - |N^+(X)|\}$. One-way pairs are closely related to k-strong connectivity.

Lemma 14.3.2 *[354] A digraph* $D = (V, A)$ *is* k*-strong if and only if we have* $h(X, Y) \geq k$ *for every one-way pair* (X, Y) *in* D.

Proof: Suppose first that D is k-strong. By Corollary 5.4.2, there are k internally disjoint (s, t)-paths for every choice of distinct vertices $s, t \in V$. Now let (X, Y) be a one-way pair and take $s \in X, t \in Y$. For every collection of the k internally disjoint paths from s to t, each such path must use a vertex in $V - X - Y$ and hence $h(X, Y) \geq k$. Conversely, assume that $h(X, Y) \geq k$ for every one-way pair (X, Y). Let S be a minimal separator of D. By the definition of a separator, $V - S$ can be divided into two sets X, Y so that there is no arc from X to Y in $D - S$ (namely, let s, t be separated by S

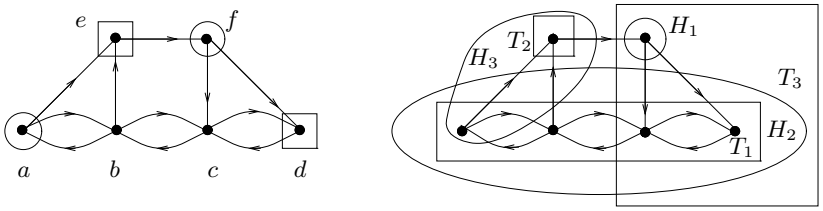


Figure 14.4 An example, due to Jordán [576, Figure 3.9.1], showing that the subpartition lower bound is not always attainable. The desired connectivity is $k = 2$ and the value $\gamma_2^*(D)$ is 2 and it is realized by the subpartitions $\{\{d\}, \{e\}\}, \{\{a\}, \{f\}\}$, respectively (see (a)). Part (b) shows three pairwise independent one-way pairs $(T_1, H_1), (T_2, H_2), (T_3, H_3)$ (tails are indicated by boxes). This shows that $a_2(D) \geq 3$. In fact, $a_2(D) = 3$, since adding the arcs af, ed, da will result in a 2-strong digraph.

and let X denote those vertices that can be reached from s in $D - S$ and $Y = V - X - S$). Thus (X, Y) is a one-way pair and $h(X, Y) = |S|$ showing that $|S| \geq k$ and hence D is k-strong. □

Two one-way pairs $(X, Y), (X', Y')$ are **independent** if either their heads or their tails are disjoint. Hence one-way pairs that contribute to the sums in Definition 14.3.1 are always independent since either all heads or all tails are disjoint for those pairs. As we saw in Figure 14.4, the sum of deficiencies over one-way pairs for which either all tails are disjoint or all heads are disjoint does not always provide the right lower bound for the number of new arcs needed in order to make the digraph k-strong.

By Lemma 14.3.2, in order to obtain a k-strong superdigraph of D, we must add enough new arcs to cover all one-way pairs with $\eta_k(X, Y) > 0$ (we must add at least $\eta_k(X, Y)$ arcs from X to Y). Clearly, if $(X, Y), (X', Y')$ are independent one-way pairs, then no new edge can decrease both $\eta_k(X, Y)$ and $\eta_k(X', Y')$. This shows that if $\mathcal{F}$ is any family of pairwise independent one-way pairs in D, then we must add at least

$$\eta_k(\mathcal{F}) = \sum_{(X,Y)\in\mathcal{F}} \eta_k(X, Y) \tag{14.10}$$

new arcs to D in order to obtain a k-strong digraph. We call the number $\eta_k(\mathcal{F})$ the **deficiency** of $\mathcal{F}$. Now consider Figure 14.4(b). Here we have indicated one-way pairs (T_i, H_i), $i = 1, 2, 3$. These are pairwise independent and have total deficiency 3. Thus it follows from our arguments above that we need at least 3 new arcs to make D k-strong. In fact, 3 arcs are sufficient in this case as pointed out in the caption of the figure.

14.3.2 Optimal k-Strong Augmentation

The following theorem, due to Frank and Jordán, shows that the maximum deficiency over families of independent one-way pairs gives the right lower bound for the vertex-strong connectivity augmentation problem.

Theorem 14.3.3 (The Frank-Jordán vertex-connectivity augmentation theorem) *[354] For every digraph D on at least $k+1$ vertices we have*

$$a_k(D) = \max_{\mathcal{F}}\{\eta_k(\mathcal{F})\}, \tag{14.11}$$

where $\mathcal{F}$ is a family of independent one-way pairs in D. □

Frank and Jordán also gave a polynomial algorithm for finding an optimal augmenting set of size $a_k(D)$. Their algorithm was based on the ellipsoid method[1] and hence is not a combinatorial algorithm. Recently [885] Végh and Benczúr found such an algorithm.

Theorem 14.3.4 *[885] There exists a combinatorial $O(n^7)$ algorithm which, given a digraph $D = (V, A)$ on n vertices and a natural number k, finds a minimum cardinality set F of new arcs to add to D so that the resulting graph is k-strong.* □

Earlier Frank [355] had given a combinatorial polynomial algorithm found for fixed k. It is beyond the scope of this book to describe any of these algorithms here. The combinatorial algorithms in [355] and [885] rely on a detailed study of the structure of one-way pairs. We refer to the proof of Lemma 5.6.10 for an example of a proof that uses the structure of one-way pairs.

Although we may have $a_k(D) > \gamma_k^*(D)$ as we saw in Figure 14.4, Frank and Jordán proved (see below) that the difference cannot be arbitrary large. A family $\mathcal{F}$ of independent one-way pairs is **subpartition-type** if either all the tails in $\mathcal{F}$ are pairwise disjoint, or all the heads in $\mathcal{F}$ are pairwise disjoint. It is easy to see that if $\mathcal{F}$ is subpartition-type, then $\eta_k(\mathcal{F}) \leq \gamma_k^*(D)$.

Proposition 14.3.5 *[355] For any digraph $D = (V, A)$ and any target connectivity k there exists a family $\mathcal{F}$ of independent one-way pairs such that the deficiency, $\eta_k(\mathcal{F})$, of $\mathcal{F}$ equals $a_k(D)$ and $\mathcal{F}$ is either subpartition-type or the disjoint union of two families of subpartition-type. Thus $a_k(D) \leq 2\gamma_k^*(D)$.* □

The next result shows that if we need to add many arcs to D (in terms of k) to make it k-strong, then the subpartition lower bound is attainable.

[1] For a thorough treatment of the ellipsoid method and its consequences for Combinatorial Optimization, see the book [430] by Grötschel, Lovász and Schrijver.

Proposition 14.3.6 *[355] If $\mathcal{F}$ is a family of independent one-way pairs and $\eta_k(\mathcal{F}) \geq 2k^2 - 1$, then $\mathcal{F}$ is subpartition-type. Hence if $a_k(D) \geq 2k^2 - 1$, then $\gamma_k^*(D) = a_k(D)$.* □

Now let us consider the special case of the vertex-strong connectivity augmentation problem when we want to increase $\kappa(D)$ from k to $k+1$. The following result is due to Frank and Jordán:

Theorem 14.3.7 *[355] If $\kappa(D) = k$ and $a_{k+1}(D) \geq 2k+2$, then $a_{k+1}(D) = \gamma_{k+1}^*(D)$.* □

Frank and Jordán also showed that when we augment the connectivity by just one, then we can restrict the structure of the set of new arcs.

Theorem 14.3.8 *[354] If $\kappa(D) = k$, then D can be optimally augmented to a $(k+1)$-strong digraph by adding disjoint cycles and paths. In particular, if D is a k-strong and k-regular digraph, then there are disjoint cycles covering V whose addition to D gives a $(k+1)$-strong and $(k+1)$-regular digraph.* □

It is instructive to compare this result with Theorem 5.6.11.

14.3.3 Special Classes of Digraphs

For general digraphs one cannot say much about the structure of families of independent one-way pairs, but as we are going to see, there are (non-trivial) classes of digraphs for which nice structure can be found and hence a good estimate on the value of $a_k(D)$ can be given. The first result, due to Masuzawa, Hagihara and Tokura, deals with in-branchings.

Theorem 14.3.9 *[686] Let $B = (V, A)$ be an in-branching. Then $a_k(B)$ is given by $a_k(B) = \sum_{v \in V} \max\{0, k - d^-(v)\}$.* □

The proof of this result in [686] is long, but Frank and Jordán found a short proof based on Theorem 14.3.3, see [355].

For an arbitrary digraph we define the numbers ${\eta_k}^-(D), {\eta_k}^+(D)$ by

$$\eta_k^-(D) = \sum_{v \in V} \max\{0, k - d^-(v)\}, \tag{14.12}$$

$$\eta_k^+(D) = \sum_{v \in V} \max\{0, k - d^+(v)\}. \tag{14.13}$$

Frank made the following conjecture, which would imply that we have $a_k(D) = \gamma_k(D)$ for every acyclic digraph D:

Conjecture 14.3.10 *[345] For any acyclic digraph D on at least $k+1$ vertices $a_k(D) = \max\{{\eta_k}^-(D), {\eta_k}^+(D)\}$.*

A partial result was obtained by Frank and Jordán in [355].

Lemma 14.3.11 *[355] Let $D = (V, A)$ be an acyclic digraph for which $a_k(D) = \gamma_k^*(D)$. Then $a_k(D) = \max\{{\eta_k}^-(D), {\eta_k}^+(D)\}$.*

Proof: Since $a_k(D) = \gamma_k^*(D)$, there exists some family $\mathcal{F}$ of independent one-way pairs with $\eta_k(\mathcal{F}) = a_k(D)$ such that all tails, or all heads, in $\mathcal{F}$ are pairwise disjoint. By considering the converse of D if necessary, we may assume that the tails $\{T_1, \ldots, T_t\}$ of $\mathcal{F}$ are pairwise disjoint.

Because D is acyclic, the subgraph induced by T_i is acyclic for each $i \in [t]$. Hence each T_i contains a vertex x_i of out-degree zero in $D\langle T_i\rangle$. Thus $N^+(x_i) \subseteq N^+(T_i)$ and hence $k - d^+(x_i) \geq k - |N^+(T_i)| \geq k - h(T_i, H_i)$ for each $i \in [t]$. Now we obtain

$$\begin{aligned} a_k(D) &\geq {\eta_k}^+(D) \\ &\geq \sum_{i=1}^{t}(k - d^+(x_i)) \\ &\geq \sum_{i=1}^{t}(k - h(T_i, H_i)) \\ &\geq a_k(D), \end{aligned}$$

showing that $a_k(D) = {\eta_k}^+(D)$. □

Bang-Jensen made the following conjecture at a meeting in Budapest in 1994:

Conjecture 14.3.12 *For every semicomplete digraph D on at least $k + 1$ vertices*

$$a_k(D) \leq \frac{k(k+1)}{2}.$$

If true, this would be the best possible since a transitive tournament T on $n \geq k + 1$ vertices needs this many arcs. To see this it suffices to observe that if $v_1, v_2, \ldots, v_n$ is the unique acyclic ordering of the vertices in T, then the first k vertices need $k, k-1, \ldots, 2, 1$ new arcs entering them in order to satisfy the condition that the in-degree is at least k. It is not difficult to check (Exercise 14.5) that one can always make a transitive tournament k-strong by adding $\frac{k(k+1)}{2}$ new arcs. The following partial result follows from the work of Frank and Jordán [355]:

Proposition 14.3.13 *For every semicomplete digraph D on at least $k + 1$ vertices we have $a_k(D) \leq k^2$.*

Proof: We prove this by showing that if D is an r-strong semicomplete digraph which has at least $r + 2$ vertices, then we need at most $2r + 1$ new arcs to make it $(r + 1)$-strong. This will imply that we need at most k^2 arcs to make any semicomplete digraph k-strong.

Suppose first that D is not strongly connected. Since every semicomplete digraph has a Hamilton path (by Theorem 1.4.2), it follows that we can make D strong by adding one arc.

Suppose now that $r \geq 1$ and that there is some r-strong semicomplete digraph D for which we need at least $2r+2$ arcs to obtain an $(r+1)$-strong semicomplete digraph from D. Thus $a_{r+1}(D) \geq 2r+2$ and then we conclude from Theorem 14.3.7 that $a_{r+1}(D) = \gamma^*_{r+1}(D)$. Hence, by the definition of $\gamma^*_{r+1}(D)$, there exist $2r+2$ pairwise disjoint sets $X_1, X_2, \ldots, X_{2r+2}$, such that either each of these has $|N^+(X_i)| = r$ or each has $|N^-(X_i)| = r$. By considering the converse of D if necessary, we may assume that $|N^+(X_i)| = r$ for each X_i. Let X' be obtained by taking one vertex x_i from each X_i and let $D' = D\langle X' \rangle$. Since D' is semicomplete and has $2r+2$ vertices, it is easy to see that some x_i has at least $r+1$ out-neighbours in D'. However, each of these contributes to $|N^+_D(X_i)|$, a contradiction. □

14.4 Arc Reversals and Vertex-Strong Connectivity

Suppose now that we want to increase the vertex-strong connectivity of a digraph by reorienting arcs rather than adding new ones. This gives rise to the following problem.

Problem 14.4.1 *Given a natural number k and a digraph $D = (V, A)$ on at least $k+1$ vertices, find a minimum set $F \subset A$ of arcs in D such that the digraph D' obtained from D by reversing every arc in F is k-strong.*

If such a subset exists, then we let $r_k(D) = |F|$, where F is a minimum cardinality subset of A, whose reversal makes the resulting digraph k-strong. Otherwise we let $r_k(D) = \infty$.

We saw in Section 13.1 that, using submodular flows, we can find $r_1(D)$ in polynomial time for any digraph. For arbitrary digraphs and $k \geq 2$ it is not clear how we can decide whether such a reversal even exists, let alone find an optimal one (unless we try all possibilities which clearly requires exponential time). Indeed, this seems to be a very difficult problem (see also Conjecture 11.6.2). Clearly,

$$a_k(D) \leq r_k(D), \tag{14.14}$$

since, instead of reversing in D, we may add exactly those new arcs we would obtain by reversing and keep the original ones.

The next lemma, whose proof is left to the reader as Exercise 14.10, shows that for semicomplete digraphs D, $r_k(D)$ is bounded by a function depending only on k.

Lemma 14.4.2 *[109] If a semicomplete digraph D has at least $2k+1$ vertices, then $r_k(D) \leq \binom{4k-2}{2}$.* □

Bang-Jensen and Jordán showed that, somewhat surprisingly, as soon as the number of vertices in the given semicomplete digraph D is sufficiently high (depending only on k), the minimum number of arcs in D we need to reverse in order to achieve a k-strong semicomplete digraph equals the minimum number of new arcs we need to add to D to obtain a k-strong semicomplete digraph.

Theorem 14.4.3 *[109] If D is a semicomplete digraph on $n \geq 3k-1$ vertices, then $a_k(D) = r_k(D)$.* □

The idea is to show that $r_k(D) \leq a_k(D)$, by demonstrating that a certain optimal augmenting set F of D has the property that if we reverse the existing (opposite) arcs of F in D, then we obtain a k-strong semicomplete digraph. As we point out later, even for semicomplete digraphs, it is by no means the case that just an arbitrary optimal augmenting set will have this property. It was shown in [109] that $3k-1$ is the best possible for semicomplete digraphs. However, in the case when D is tournament, the question as to whether or not the bound is best possible was left open and the following conjecture was implicitly formulated.

Conjecture 14.4.4 *[109] For every tournament D on at least $2k+1$ vertices, we have $a_k(D) = r_k(D)$.*

One may argue that perhaps if we restrict ourselves to only adding arcs between adjacent vertices, then we could have $a_k(D) = r_k(D)$ for arbitrary digraphs D, provided both numbers are finite and the number of vertices in D is large enough. This is not true, however, as can be seen from the digraph D in Figure 14.5. It is not difficult to see that $a_2(D) = 1$ and that any arc whose addition to D results in a 2-strong digraph has tail x and head in $T_2 \cup z$ (in particular, $D + xz$ is 2-strong). On the other hand, it is also easy to see that $r_2(D) \geq 2$ (Exercise 14.15). This example can be modified to work for any $k \geq 1$ (Exercise 14.16).

If we add arcs to the digraph D described above without increasing the number of out-neighbours of x and of z, we can construct a semicomplete digraph D' of any given size for which xz is an optimal augmentation but reversing xz does not make D' 2-strong. This construction for $k = 2$ and similar constructions for higher connectivity show that even for semicomplete digraphs we cannot reverse along an arbitrary optimal augmenting set for $k \geq 2$.

The following conjecture, which implies Conjecture 14.3.12, was made by Bang-Jensen at a meeting in Budapest in 1994. Again the transitive tournament on $n \geq 2k+1$ vertices shows that the bound would be best possible if true.

Conjecture 14.4.5 *For every tournament T on n vertices and every positive integer k such that $n \geq 2k+1$ we have $r_k(T) \leq \frac{k(k+1)}{2}$.*

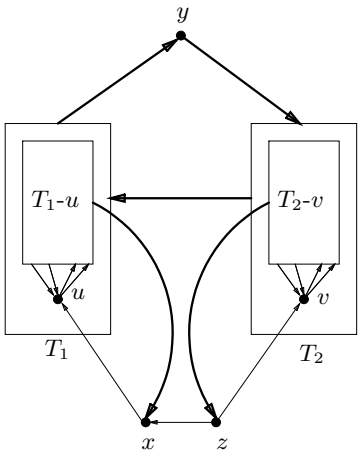


Figure 14.5 A digraph with $a_2(D) = 1$ and $r_2(D) = 2$. The digraphs T_1 and T_2 are 2-strong. Bold arcs between sets of vertices indicate that all arcs between these sets are present and have the direction shown.

We complete this section with the following useful observation, which we use in Section 11.10.

Lemma 14.4.6 *[66, 435] Let $D = (V, A)$ be a k-strong digraph and let xy be an arc of D. If D has at least $(k+1)$-internally disjoint (x, y)-paths each of length at least 2, then the digraph D' obtained from D by replacing the arc xy by the arc yx (or just deleting xy if $yx \in A$) is k-strong. Furthermore, if D' is not $(k+1)$-strong, then every minimum separating set S' of D' is also separating in D.*

Proof: Suppose that D' is not $(k+1)$-strong. Let S' be a minimum separator of D'. Then $|S'| \leq k$ and there is some pair a, b of vertices separated by S' in D'. It follows from the assumption on $\kappa(x, y)$ that either $S' \cap \{x, y\} \neq \emptyset$ or S' does not separate x, y. From this we get that $\{a, b\} \neq \{x, y\}$ and that a, b are also separated by S' in D. This shows that every minimum separating set of D' is also separating in D. Since D is k-strong we have $|S'| = k$ and hence D' is k-strong. □

14.5 Arc-Reversals and Arc-Strong Connectivity

Let $r_k^{deg}(D)$ be the minimum number of arcs one needs to reverse in a directed multigraph D in order to obtain a directed multigraph D' with $\delta^0(D') \geq k$. If no such reversal exists, we let $r_k^{deg}(D) = \infty$. Analogously define $r_k^{arc}(D)$ to be the minimum number of arcs one needs to reverse in D in order to obtain a k-arc-strong directed multigraph. By Theorem 11.5.3, $r_k^{arc}(D) < \infty$ precisely

when $UMG(D)$ is $2k$-edge-connected. We saw in Section 11.8.4 that we can determine $r_k^{arc}(D)$ in polynomial time using submodular flows.

For an arbitrary directed multigraph D the value of $r_k^{arc}(D)$ may depend on the number of vertices of D. We prove below that for tournaments $r_k^{arc}(T)$ is always bounded by a quadratic function of k. It follows from our proofs that one can determine $r_k^{arc}(T)$ for an arbitrary tournament T using standard minimum cost flows instead of the more complicated submodular flows. The results in this section are from [121] by Bang-Jensen and Yeo.

14.5.1 Determining $r_k^{deg}(D)$ Efficiently

We start by observing that the problem of determining $r_k^{deg}(D)$ and finding an optimal set of arcs to reverse can be solved using flows in networks for any given directed multigraph D.

Let $D = (V, A)$ be an arbitrary directed multigraph and let $N = (V, A, l \equiv 0, u \equiv 1)$ be the corresponding flow network in which every arc has capacity one and lower bound zero. Starting from any directed multigraph D' which is obtained from D by reversing some arcs, we can define an integer flow x in N by taking $x_{ij} = 1$ precisely if the arc ij is reversed when going from D to D'. It is easy to see that we may also go the other way. Hence we may study reversals of arcs in D through integer flows in N. Given an integer flow x in N, let D' be obtained from D by reversing those arcs ij that have $x_{ij} = 1$. The in-degree of a vertex i in D' is given by $d^-_{D'}(i) = d^-_D(i) + \sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji}$. Hence, in order for D' to have $\delta^0(D') \geq k$ we must have

$$k \leq d^-_D(i) + \sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji} \leq d^+_D(i) + d^-_D(i) - k, \tag{14.15}$$

where the last inequality ensures that $d^+_{D'}(i) \geq k$. The condition above is equivalent to requiring that the flow x satisfies $0 \leq x_{ij} \leq 1$ and

$$d^+_D(i) - k \geq \sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji} \geq k - d^-_D(i). \tag{14.16}$$

This is just a feasibility problem for flows and hence can be solved in polynomial time using any algorithm for finding a maximum flow in a network (see Exercise 4.32). By introducing the cost 1 on every arc and solving a minimum cost flow problem, in polynomial time, we can determine $r_k^{deg}(D)$ and find an optimal reversing set, or determine that $r_k^{deg}(D) = \infty$ which corresponds to the case when there is no feasible flow in N.

Note that if D has 2-cycles, then the optimal reversal may involve the creation of parallel arcs. If D has no multiple arcs and we insist that the reversal does not create multiple arcs, we can exclude the reversal of arcs in 2-cycles by letting $l_{ij} = u_{ij} = 0$ for every arc which is part of a 2-cycle.

The following result by Bang-Jensen and Yeo shows that for tournaments r_k^{deg} depends only on k. The result below is best possible since for all $n \geq 2k+1$ we have $r_k^{deg}(TT_n) = k(k+1)/2$.

Theorem 14.5.1 *[121] If T is a tournament, with $|V(T)| \geq 2k+1$, then $r_k^{deg}(T) \leq k(k+1)/2$.* □

14.5.2 Reversals of Arcs to Achieve High Arc-Strong Connectivity in Tournaments

The next result by Bang-Jensen and Yeo shows that in the case of tournaments, the numbers r_k^{deg} and r_k^{arc} are closely related.

Theorem 14.5.2 *[121] For every tournament T with $|V(T)| = n \geq 2k+1$ we have*

$$r_k^{arc}(T) = \max\{k - \lambda(T), r_k^{deg}(T)\}.$$

In particular, if $r_k^{deg}(T) \geq k - \lambda(T)$, then $r_k^{arc}(T) = r_k^{deg}(T)$.

Proof: Recall that for $X, Y \subseteq V(T)$ we denote by $|(X,Y)|$ the number of arcs with tail in X and head in Y. Let $q = \max\{k - \lambda(T), r_k^{deg}(T)\}$ and let T' be a tournament obtained from T by reversing at most q arcs, such that the following holds:

(i) $\delta^0(T') \geq k$.
(ii) $\sum_{x \in V(T')} (d^+(x))^2$ is minimum.

Note that there exists such a T', by the definition of q. If T' is k-arc-strong, then we are done, so assume that $\lambda(T') < k$. Let S be chosen such that $|(S, V-S)| = \lambda(T')$ in T', and such that $|S|$ is minimum among all subsets S' with $|(S', V-S')| = \lambda(T')$. As $\delta^+(T'), \delta^-(T') \geq k$ and $\lambda(T') < k$, we have $2 \leq |S| \leq |V(T')| - 2$.

If there exists a vertex $x \in S$, with $|(S,x)| \leq |(x, V-S)|$, then $S' = S - x$ is a contradiction against the choice of S. Therefore $|(S,x)| > |(x, V-S)|$, which implies that $d^+_{T'}(x) \leq |S| - 2$, for all $x \in S$. The minimality of $|(S, V-S)|$ implies that $|(S,y)| \leq |(y, V-S)|$ and hence we get $d^+_{T'}(y) \geq |S|$, for all $y \in V - S$. If $|S| \leq 2k$, then

$$\sum_{x \in S} d^+_{T'}(x) = |(S,S)| + |(S, V-S)| < |S|(|S|-1)/2 + k \leq k(|S|-1) + k = k|S|,$$

which is a contradiction to $\delta^+(T') \geq k$. Therefore $|S| \geq 2k+1$. Analogously we can prove that $|V - S| \geq 2k+1$. This implies that for all $x \in S$ we have $d^-_{T'}(x) \geq n - 1 - (|S|-2) \geq |V-S| + 1 \geq 2k+2 > k+1$ and for all $y \in V - S$ we have $d^+_{T'}(y) \geq |S| \geq 2k+1 > k+1$.

Note that reversing any arc yx which goes from $V - S$ to S in T' will maintain (i) and decrease (ii) as $d^+_{T'}(y) \geq d^+_{T'}(x) + 2$. Hence it follows from the

fact that $|S|, |V-S| \geq 2k+1$ (implying that T' contains arcs from $V-S$ to S) that we have reversed exactly q arcs in order to obtain T' and furthermore every arc from $V-S$ to S in T' also goes from $V-S$ to S in T (otherwise we could improve (ii) by not reversing such an arc originally). Let R denote the arcs in T' which have an opposite direction to what they had in T (i.e., R are the arcs that have been reversed, and $|R| = q$). We will now show that all arcs in R go from S to $V-S$ in T'. It follows from the remark above that there is no $(V-S, S)$-arc in R.

If there exists an (S, S)-arc in R, then let vu be such an arc (i.e., $uv \in A(T)$). As $|V-S| \geq 2k+1$ and $|(S, V-S)_{T'}| < k$ (the number of arcs from S to $V-S$ in T'), there exists a vertex w in $V-S$, with $wv \in A(T')$. Now consider the tournament T'', obtained from T' by reversing vu and wv. Note that T'' also has q arcs reversed compared to T (uv is reversed back again). Compared to T' we see that all degrees stay the same except that $d^+(w)$ decreases by one and $d^-(u)$ decreases by one. Therefore we still have $\delta^+(T''), \delta^-(T'') \geq k$, and we obtain a contradiction against (ii).

If there exists a $(V-S, V-S)$-arc in R, we analogously obtain a contradiction. Therefore all arcs in R are $(S, V-S)$-arcs.

Since we have reversed q arcs, there are at least $\lambda(T) + q \geq k$ arcs in T' from S to $V-S$, contradicting the assumption that T' has fewer than k such arcs. □

Note that the proof above can be turned into a polynomial algorithm for finding a set of q arcs whose reversal makes T k-arc-strong using just flows instead of the more complicated submodular flows (as we mentioned in the introduction, one can determine $r_k^{arc}(D)$ for an arbitrary digraph D using minimum cost submodular flows). We leave the details to the interested reader.

Combining Theorem 14.5.1 with Theorem 14.5.2, we obtain the following upper bound on $r_k^{arc}(T)$ which provides support to Conjecture 14.4.5. Recall again that the transitive tournaments show that this is best possible.

Corollary 14.5.3 *[121] For every tournament T with $|V(T)| = n \geq 2k+1$ we have $r_k^{arc}(T) \leq k(k+1)/2$.* □

For general digraphs D there need not be any close relation between the numbers $r_k^{deg}(D)$ and $r_k^{arc}(D)$. For example, let $D = P_t[C_3, \ldots, C_3]$ be the digraph obtained by substituting a 3-cycle for each vertex of a directed path P_t on t vertices. Then it is easy to see that $r_2^{deg}(D) = 6$ and $r_2^{arc}(D)$ is proportional to t and hence can be made much larger than $r_2^{deg}(D)$ by increasing t.

14.6 Increasing Connectivity by Deorienting Arcs

It is not difficult to find examples of tournaments T for which $a_k(T) > r_k^{arc}(T)$ and $r_k^{arc}(T) < \infty$ since adding parallel arcs may increase the arc-strong

connectivity. On the other hand, parallel arcs have no effect on vertex-strong connectivity and in fact we saw in Theorem 14.4.3 that when $n \geq 3k-1$ we have $a_k(D) = r_k(D)$ for every semicomplete digraph. Recall also Conjecture 14.4.4.

We now consider the operation of deorienting an arc. Let xy be an ordinary[2] arc of a digraph D. By **deorienting** xy we mean the operation which adds the arc yx to D. Clearly, deorienting arcs is equivalent to adding new arcs with the restriction that we can only add an arc which is opposite to an existing arc and we cannot create parallel arcs. Hence we may view deorienting arcs as a restricted version of the arc addition operation.

Let $deor_k^{deg}(D)$ denote the minimum number of arcs we need to deorient in D in order to obtain a digraph D' with $\delta^0(D') \geq k$. Clearly $deor_k^{deg}(D) < \infty$ if and only if each vertex of D has degree at least k in $UG(D)$ and $deor_k^{deg}(D) \leq r_k^{deg}(D)$ for every oriented graph. If D contains directed 2-cycles, this may not hold. See, e.g., Figure 14.6. In Exercise 14.18 the reader is asked to give a polynomial algorithm for finding $deor_k^{deg}(D)$ for an arbitrary digraph D. For tournaments we have the following characterization by Bang-Jensen and Yeo.

Theorem 14.6.1 *[121] Let T be a tournament on at least $2k+1$ vertices. Then $deor_k^{deg}(T) = r_k^{deg}(T)$. In particular, $deor_k^{deg}(T) \leq k(k+1)/2$.* □

Analogously define $deor_k^{arc}(D)$ to be the minimum number of arcs one needs to deorient in D in order to obtain a k-arc-strong digraph. It is easy to see that $deor_k^{arc}(D) < \infty$ if and only if $UG(D)$ is k-edge-connected. Furthermore, if D is an oriented graph (in particular, if D is a tournament), then we have $deor_k^{arc}(D) \leq r_k^{arc}(D)$ since instead of reversing an optimal set A' of arcs we may deorient these arcs and obtain a digraph with minimum semidegree at least k. Figure 14.6 shows that the inequality above may not hold when D contains 2-cycles. The following is a corollary of Theorem 13.1.2.

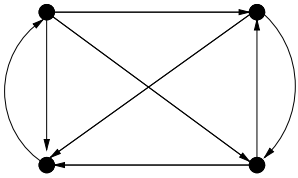


Figure 14.6 A digraph D for which $1 = r_2^{arc}(D) < deor_2^{arc}(D) = 2$.

Theorem 14.6.2 *Let D be a non-strong digraph for which $UG(D)$ is 2-edge-connected. Then $deor_1^{arc}(D) = r_1^{arc}(D)$.* □

[2] Recall that this means that D does not contain the arc yx.

When $k \geq 2$ and D is an arbitrary digraph we do not know how to determine $deor_k^{arc}(D)$ efficiently, but as we show below, this is possible when D is a tournament.

One might expect that $deor_k^{arc}(D) < r_k^{arc}(D)$ for most oriented graphs. The next result shows that for tournaments the two numbers are equal and hence, with respect to increasing the arc-strong connectivity of a tournament, there is no gain from deorienting arcs rather than reversing arcs.

Theorem 14.6.3 *[121] For every tournament T on at least $2k+1$ vertices we have $deor_k^{arc}(T) = r_k^{arc}(T)$.*

Proof: We saw in Theorem 14.5.2 that $r_k^{arc}(T) = \max\{k - \lambda(T), r_k^{deg}(T)\}$. If $r_k^{arc}(T) = r_k^{deg}(T)$, then, by Theorem 14.6.1, we have

$$\begin{aligned} deor_k^{arc}(T) &\leq r_k^{arc}(T) \\ &= r_k^{deg}(T) \\ &= deor_k^{deg}(T) \\ &\leq deor_k^{arc}(T), \end{aligned}$$

implying that $deor_k^{arc}(T) = r_k^{arc}(T)$. So we may assume that $r_k^{arc}(T) = k - \lambda(T)$. Now the claim follows from the easy fact that $deor_k^{arc}(T) \geq k - \lambda(T)$. □

We argued in Section 14.5.2 that we can find, in polynomial time, a set of arcs $A' \subset A(T)$ of size $r_k^{arc}(T)$ in a tournament T such that reversing the arcs of A' results in a k-arc-strong tournament. Thus it follows from Theorem 14.6.3 that, in polynomial time, we can determine $deor_k^{arc}(T)$ and find a set of $deor_k^{arc}(T)$ arcs to deorient such that the resulting semicomplete digraph is k-arc-strong. One optimal set of arcs to deorient is simply a set that would form an optimal reversal.

Problem 14.6.4 *[121] Is there a polynomial algorithm which, given a digraph D and positive integer k, determines $deor_k^{arc}(D)$?*

As we saw above the answer is yes if $k = 1$ or if D is a tournament but we do not even know whether there exists a polynomial algorithm for general oriented graphs when $k = 2$. Even the case of semicomplete digraphs is open. Recall that $deor_k^{deg}(D)$ can be calculated in polynomial time via flows (Exercise 14.18).

Problem 14.6.5 *[121] Is there a polynomial algorithm which, given an oriented graph D and positive integer k, determines $deor_k^{arc}(D)$?*

Analogously to the definition of $deor_k^{arc}(D)$ we may define $deor_k(D)$ to denote the minimum number of arcs we need to deorient in D in order to obtain a k-strong digraph. Clearly $deor_k(D) < \infty$ precisely when $UG(D)$ is k-connected. Notice that when D is a semicomplete digraph we have $deor_k(D) = a_k(D)$ and hence the results in Section 14.4 apply.

Problem 14.6.6 *Determine classes of digraphs for which $deor_k(D)$ can be determined efficiently.*

Problem 14.6.7 *Is there a polynomial algorithm for deciding whether a digraph D satisfies $deor_k^{arc}(D) = r_k^{arc}(D)$?*

Problem 14.6.8 *Characterize those oriented graphs D for which one has $deor_k^{arc}(D) = r_k^{arc}(D)$.*

14.7 Miscellaneous Topics

14.7.1 Increasing Arc-Strong Connectivity of a Bipartite Digraph

The deorienting operation above gives rise to a special type of augmentation problems in which we are not just interested in increasing the arc-strong connectivity of a digraph by adding as few arcs as possible, but we also have a structure requirement on the possible arcs that may be added. Along the same line, Gabow and Jordán [372] considered the following situation which generalizes the problem of increasing the arc-strong connectivity of a bipartite digraph while preserving bipartiteness. We are given a directed graph $D = (V, A)$, a bipartition $V = X_1 \cup X_2$ of V and a positive integer k. The goal is to find a smallest set F of arcs for which $D' = (V, A \cup F)$ is k-arc-strong and $F \subseteq (X_1, X_2) \cup (X_2, X_1)$. For D as above and $i = 1, 2$ define $\gamma_k^+(D, X_i)$ and $\gamma_k^-(D, X_i)$ as follows:

- $\gamma_k^+(D, X_i) = \max\{\sum_{j=1}^p (k - d^+(U_j)) : U_1, \ldots, U_p$ is a subpartition of $X_i\}$,
- $\gamma_k^-(D, X_i) = \max\{\sum_{j=1}^p (k - d^-(U_j)) : U_1, \ldots, U_p$ is a subpartition of $X_i\}$.

In Section 14.2 we proved that the minimum number of new arcs we need to add to D in order to obtain a k-arc-strong directed multigraph $D' = (V, A \cup F)$ is exactly $\gamma_k(D)$. In the bipartition constrained augmentation problem we may need more than $\gamma_k(D)$ arcs. See Figure 14.7 for an example from [371].

Let $\Phi_{(k,X_1,X_2)}(D) = \max\{\gamma_k(D), \gamma_k^+(D, X_1) + \gamma_k^-(D, X_1), \gamma_k^+(D, X_2) + \gamma_k^-(D, X_2)\}$ and denote by $OPT_{X_1,X_2}^k(D)$ the minimum number of new arcs we must add to D in order to obtain a k-arc-strong bipartite digraph.

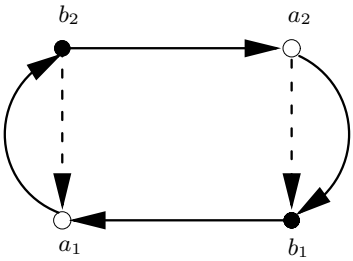


Figure 14.7 A k-arc-strong bipartite multigraph. The full arcs have multiplicity k and the dotted arcs have multiplicity one. By adding the arcs a_1a_2, b_1b_2 we can increase the arc-strong connectivity by one. However, if the resulting digraph must remain bipartite, then we need to add the three arcs b_2a_2, b_1a_1, a_1b_2 [372].

To see that $OPT^k_{X_1,X_2}(D) \geq \Phi_{(k,X_1,X_2)}(D)$, it suffices to observe that no single arc can decrease the deficiency of more than one deficient set which lies entirely inside X_1 or X_2.

In the digraph of Figure 14.7 we have

$$3 = OPT^{k+1}_{X_1,X_2}(D) > \Phi_{(k+1,X_1,X_2)}(D) = 2,$$

showing that equality does not always hold. Let $H = (V = X_1 \cup X_2, A)$ with $X_1 = \{a_1, a_2\}$, $X_2 = \{b_1, b_2\}$ where A consists of k copies of each of the arcs a_1b_1, b_2a_2. It is easy to see that $3k = OPT^k_{X_1,X_2}(H) > \Phi_{(k,X_1,X_2)}(H) = 2k$, so we may have $OPT^k_{X_1,X_2}(D) \geq \Phi_{(k,X_1,X_2)}(D) + k$. Using a (non-trivial) modification of Frank's algorithm from Section 14.2, Gabow and Jordán found a polynomial algorithm which adds at most k arcs more than the optimum, thereby showing that $OPT^k_{X_1,X_2}(D) \leq \Phi_{(k,X_1,X_2)}(D) + k$ always holds.

14.7.2 Augmenting Arc-Strong Connectivity in Directed Hypergraphs

One can define a directed hypergraph in many ways. Below we follow Frank et al. [356] and give a definition that straightforwardly generalizes the notion of a directed graph. To make it more clear what is going on, we use the name star hypergraph [116] for this kind of orientation of a hypergraph. A **star hypergraph** H^* is a hypergraph $H = (V, \mathcal{E})$ together with a function $h : \mathcal{E} \to V$ that associates one vertex $h(a) \in a$ to each hyperedge $a \in \mathcal{E}$. We write $H^* = (V, A)$, where $A = (\mathcal{E}, h)$, and call $h(a)$ the **head** of a. For each of the definitions below let $H^* = (V, A)$ be a star hypergraph. We always denote by $H = (V, \mathcal{E})$ the underlying hypergraph of H^*, that is, the hypergraph we obtain by ignoring the orientation (thus $\mathcal{E}$ and A contain the same edges as subsets of V). By an **arc** of H^* we always mean a hyperedge with a designated head. The arc a **enters** a set $X \subset V$ if $a \cap (V - X) \neq \emptyset$ and $h(a) \in X$. Similarly, a **leaves** X if $h(a) \notin X$ and $a \cap X \neq \emptyset$. The **in-degree** of

$X \subset V$, $d^-(X)$, is the number of arcs that enter X and the **out-degree** of X, $d^+(X)$, is the number of arcs that leave X. Note that, as for usual digraphs, we have $d^-(X) = d^+(V - X)$. Note also that an arc a may contribute to the out-degree of up to $|a| - 1$ sets in a partition $\mathcal{P}$ of V but only to the in-degree of at most one set in $\mathcal{P}$.

A **path** in H^* is a sequence $P = v_1, a_1, v_2, a_2, v_3, a_3, \ldots, a_{k-1}, v_k$ such that $v_i \in V$, for $i \in [k]$, all v_i are distinct, $a_j \in A$ for $j \in [k-1]$, $h(a_i) = v_{i+1}$ and $v_i \in a_i$ for $i \in [k-1]$. We call a path P as above an (s,t)**-path** if $s = v_1$ and $t = v_k$.

It is not difficult (Exercise 14.20) to extend Menger's theorem to star hypergraphs: there exist k-arc-disjoint (s,t)-paths in H^* if and only if $d^-(X) \geq k$ for every $X \subset V$ with $s \notin X$ and $t \in X$. We say that a star hypergraph $H^* = (V, A)$ is k-arc-strong if $d^-(X) \geq k$ for all $\emptyset \neq X \subset V$.

In [140] Berg, Jackson and Jordán consider the problem of augmenting the arc-strong connectivity of a star hypergraph by adding new hyperarcs (again with a designated head for each new hyperarc) of size at most t for some $t \geq 2$ (all new hyperarcs must have size at most t). Their main result is the following theorem which is easily seen to generalize Theorem 14.2.3.

Theorem 14.7.1 *[140] Let $H^* = (V, A)$ be a star hypergraph and γ a non-negative integer. Then H^* can be made k-arc-strong by adding γ hyperarcs of size at most t if and only if*

$$\gamma \geq \sum_{i=1}^{r}(k - d^-(X_i)) \qquad \text{and} \qquad (t-1)\gamma \geq \sum_{i=1}^{r}(k - d^+(X_i)) \quad (14.17)$$

holds for every subpartition $\{X_1, X_2, \ldots, X_r\}$ of V. □

One of the main tools in the proof of Theorem 14.7.1 is a generalization of the splitting off operation from Section 14.1 in which the special vertex s is incident only to arcs of size 2 and we may split off a set of t arcs $u_1s, u_2s, \ldots, u_ts$ entering s with one arc sv leaving s and replace these arcs by one new hyperarc $\{u_1, u_2, \ldots, u_t, v\}$ with head v.

It is briefly indicated in [140] how one can obtain a polynomial algorithm for finding an optimal augmentation of a star hypergraph to a k-arc-strong star hypergraph by adding hyperarcs of size at most t. One of the main tools is again minimum cut calculations in a suitable network via max flow calculations.

14.7.3 Weighted Versions of Local Arc-Connectivity Problems

In [47] Arkin, Hassin and Shahar studied weighted versions of the three operations adding, reversing and deorienting arcs, which we have considered above, applied to problems concerning local arc-connectivity.

In [47] the deorientation operation was replaced by the more general operation of complementing an arc. By **complementing** an arc xy in a digraph

$D = (V, A)$ we mean the operation of adding a new arc from y to x to the digraph, thereby possibly creating parallel arcs if $\mu_D(y, x) \geq 1$. Contrary to the situation for the deorientation operation, where we may have $deor_k^{arc}(D) > r_k^{arc}(D)$ (see Figure 14.6), we never need to complement more than $r_k^{arc}(D)$ arcs in D to obtain a k-arc-strong directed multigraph, as we may just complement the set of arcs in an optimal reversal.

In [47] many problems concerning local connectivities were considered for each of the three operations of adding, reversing or complementing arcs. In the first and third case, the possible new arcs are weighted and in the case of arc reversal the original arcs are weighted. That is, we are given a weight function either on $V \times V - A$ or on A (in the case of reversal) and the goal is to find a cheapest set of arcs to either add or reverse so that the given local connectivity requirement is satisfied. One can reduce the complementing problem to an addition problem with weights, simply by giving infinite cost to all new arcs xy for which there is no copy of the arc yx in the input digraph. This shows that for the weighted versions the complement version is no harder than the addition version.

The simplest problem of the type above is when the input is a digraph $D = (V, A)$, two vertices s, t and a weight function on the arcs and the goal is to add, complement or reverse arcs of minimum cost so that the resulting digraph has k arc-disjoint (s, t)-paths. All three variants can be solved by reducing to minimum cost flow problems (Exercises 14.21 and 14.22).

When we want to guarantee k-arc-disjoint paths from a given vertex s to every other vertex (or equivalently, by Edmonds' branching theorem, ensure the existence of k arc-disjoint out-branchings from s) by adding a cheapest set of new arcs or reversing a minimum cost subset of the original arcs these problems can still be solved in polynomial time by using an algorithm for weighted matroid intersection and submodular flows, respectively (Exercises 9.5 and 14.23). When we only want k arc-disjoint paths from s to a subset $T \subseteq V$, all three variants (adding, complementing and reversing) become $\mathcal{NP}$-hard [47]. Arkin et al. show that both the reversal and the complementing variant of the following problem (which they call the S TO S PROBLEM) are $\mathcal{NP}$-hard[3]: Given a digraph $D = (V, A)$ and a subset $S \subseteq V$; find a cheapest set of arcs to reverse (or complement) so that the resulting digraph contains an (s, t)-path for every choice of distinct vertices $s, t \in S$.

Problem 14.7.2 *[47] Is there a polynomial algorithm for the S to S problem when $|S| = k$, k fixed (not part of the input) and the operation is either addition, complement or reversal?*

The problem is open already for $k = 2$ in the case of reversal and for $k \geq 3$ for addition and complement. For $k = 2$ there is a polynomial algorithm in case of addition (and hence complement) [47].

[3] The addition version is also $\mathcal{NP}$-hard [342].

14.8 Exercises

14.1. **(+) Greedy deletion of arcs in Frank's algorithm**. Show how to implement Steps 2 and 3 of Frank's algorithm in Section 14.2 by using flows to find the maximum number of arcs that can be deleted for each vertex v_i (Frank [342]). Hint: let t be a vertex of $V - v_i$, identify s and t to one vertex t' and then calculate $\lambda(v_i, t')$ in the resulting directed multigraph. Do this for all $t \in V - v_i$ and let ρ be the smallest of the numbers calculated. Using Menger's theorem, show that we may delete precisely $\min\{\mu(v_i, s), \rho - k\}$ arcs from v_i to s without violating (14.2).

14.2. Perform Frank's algorithm on the digraph in Figure 14.8 when the goal is to obtain a 2-arc-strong directed multigraph.

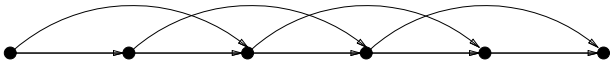


Figure 14.8 A directed graph H.

14.3. **(+) Finding an admissible split**. Show that Step 5 of Frank's algorithm in Section 14.2 can be implemented using flows. That is, show how to decide if a given splitting (us, sv) preserves k-arc-strong connectivity in V (Frank [342]). Hint: we need to decide if there is a set $U \subset V$ such that $u, v \in U$ and $d^+(U) = k$ or $d^-(U) = k$. This can be done using flows in a way similar to that outlined in the hint to Exercise 14.1.

14.4. **(+) Augmenting rooted arc-strong connectivity.** Consider the problem of finding the minimum number of new edges to add to a directed multigraph so that the new directed multigraph has $\boldsymbol{k}$ arc-disjoint out-branchings at $\boldsymbol{s}$. Show how to reduce this problem to the general k-arc-connectivity augmentation problem. Try to derive a min-max formula for the optimal number of new arcs.

14.5. **Augmenting acyclic tournaments to $\boldsymbol{k}$-strong connectivity**. Prove that an acyclic tournament on $n \geq k + 1$ vertices can be made k-strong by adding $\frac{k(k+1)}{2}$ arcs. Hint: use Exercise 5.10.

14.6. **(+)** Let $D = \vec{C}_n[\overleftrightarrow{K}_k, I_1, \overleftrightarrow{K}_k, \ldots, \overleftrightarrow{K}_k, I_1]$, where I_1 denotes the digraph which is just an isolated vertex and n is an even number. Prove that $\gamma_k(D) = k$. Try to determine $a_k(D)$.

14.7. **Obtaining new $\boldsymbol{k}$-arc-strong directed multigraphs by adding new vertices**. Let D be a k-arc-strong directed multigraph, let x be a new vertex and let D' be obtained from D and x by adding k arcs from x to arbitrary vertices of D and k arcs from arbitrary vertices of D to x. Prove that D' is k-arc-strong.

14.8. **Obtaining new $\boldsymbol{k}$-strong digraphs by adding vertices**. Let D be a k-strong digraph, let x be a new vertex and let D' be obtained from D and x by adding k arcs from x to distinct vertices of D and k arcs from distinct vertices of D to x. Prove that D' is k-strong.

14.9. **Vertices with high in- and out-degree in semicomplete digraphs.** Prove that every semicomplete digraph on at least $4k-1$ vertices has a vertex x with $d^+(x), d^-(x) \geq k$. Show that this is the best possible.

14.10. Prove Lemma 14.4.2. Hint: use Exercises 14.8 and 14.9 to reduce the problem to one for a tournament on at most $4k-2$ vertices.

14.11. **Bi-submodularity of the function $h(X,Y)$ on one-way pairs**. Let $D=(V,A)$ be a digraph. Recall that a pair (X,Y), where $X,Y \subset V$, is a one-way pair if there are no edges from X to Y and that $h(X,Y)$ is defined by $h(X,Y) := |V-(X \cup Y)|$. Prove that the function $h(X,Y)$ is **bi-submodular**, i.e., for every choice of one-way pairs $(X,Y),(X',Y')$ the following holds:

$$h(X,Y)+h(X',Y') \geq h(X \cup X', Y \cap Y') + h(X \cap X', Y \cup Y').$$

Hint: consider the contribution of a vertex $v \in V$ to each side of the inequality.

14.12. Let D be a digraph that is k-strong but not $(k+1)$-strong. Call a one-way pair (X,Y) **critical** if $h(X,Y)$=k. By Lemma 14.3.2, the family

$$\mathcal{F} = \{(X,Y) : (X,Y) \text{ is a critical one-way pair}\}$$

is non-empty. Prove that $\mathcal{F}$ is a crossing family of pairs of sets, i.e., if $(X,Y),(X',Y') \in \mathcal{F}$ satisfy $X \cap X' \neq \emptyset$ and $Y \cap Y' \neq \emptyset$, then $(X \cup X', Y \cap Y'), (X \cap X', Y \cup Y') \in \mathcal{F}$. Hint: use Exercise 14.11.

14.13. Let H be the digraph in Figure 14.8. Determine $a_2(H)$ and a set of $a_2(H)$ arcs whose addition to H results in a 2-strong digraph. Use one-way pairs to verify optimality.

14.14. Let D be a digraph with $\kappa(D)=k$ and suppose that $\gamma^*_{k+1}(D) = 2k+1$. Prove that $a_{k+1}(D) = \gamma^*_{k+1}(D)$.

14.15. Let D be the digraph illustrated in Figure 14.5. Prove that $r_2(D) \geq 2$.

14.16. Generalize the example in Figure 14.5 to obtain an infinite set of digraphs $\mathcal{D} = \{D_1, D_2, \dots\}$ such that $r_k(D_k) > a_k(D_k)$, $k = 1, 2, \dots$.

14.17. Let T be the tournament on 7 vertices shown in Figure 14.9. Show that $r_2(T) = 1$ and that $r_2(T-v) = 3$.

14.18. **Calculating $deor_k^{deg}(D)$.** Show how to calculate $deor_k^{deg}(D)$ via flows. Hint: let

$$def_k^{deg}(D) = \sum_{v \in V} \min\{0, k-d^-(v)\} + \sum_{v \in V} \min\{0, k-d^+(v)\}$$

and show how to use flows to find a maximum set of arcs A' so that deorienting these arcs decreases $def_k^{deg}(D)$ by $2|A'|$. Then use this observation to solve the full problem.

14.19. Give examples of tournaments T for which $a_k(T) < r_k^{arc}(T) < \infty$.

14.20. **Menger's theorem for star hypergraphs.** Prove Menger's theorem for star hypergraphs. Hint: follow the second proof of Menger's theorem in Section 5.4.

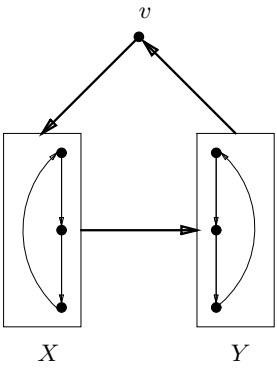


Figure 14.9 A strong tournament T on 7 vertices. The fat arcs indicate that all arcs between the sets indicated have the directions shown.

14.21. Show how to formulate the problem of finding a minimum cost set of new arcs to add to a digraph $D = (V, A)$ so that the resulting digraph has k arc-disjoint (s, t)-paths as a minimum cost flow problem.

14.22. Show how to solve the same problem as above when the operation is arc-reversal. Hint: use minimum cost flows again.

14.23. Let $D = (V, A)$ be a digraph, let s be a specified vertex of D and let ω be a non-negative weight function on A. The goal is to find a minimum cost set of arcs to reverse so that the resulting digraph contains k-arc-disjoint out-branchings rooted in s. Show how to formulate this problem as a submodular flow problem [47].

14.24. ***$\mathcal{NP}$*-hardness of the unweighted *s* to *T* reversal problem**. Show that the problem of deciding, for a given digraph $D = (V, A)$, a vertex s and a subset $T \subset V - s$, the minimum number of arcs one needs to reverse in D in order to get a digraph in which s can reach every vertex of T. Hint: show how to reduce the Set Covering Problem[4] to the problem above [47].

14.25. Use the result of the previous exercise to show that the following problem is $\mathcal{NP}$-hard: Given a digraph $D = (V, A)$ and a subset $S \subseteq V$ find a minimum cardinality subset of arcs in A so that after reversing these arcs the resulting digraph has an (s, t)-path for every choice of $s, t \in S$ [47].

[4] Given $S = \{a_1, a_2, \ldots, a_n\}$, and integer K and a family $\mathcal{F} = \{X_1, X_2, \ldots, X_m\}$ of S. Does $\mathcal{F}$ contain a subcollection of at most K subsets so that these cover every element of S at least once?

15. Feedback Sets and Vertex Orderings

In the previous chapters, we considered various properties of cycles in digraphs. In this chapter, we consider the question of how few vertices have to be deleted to get rid of all cycles. We will also consider the arc version of this question (i.e., to destroy all cycles we delete arcs rather than vertices). The last question leads us to the notion of an 'optimal' ordering of a digraph which is of interest in various applications. We will see that the minimum number of vertices (arcs) required to destroy all cycles is related to the maximum number of vertex-disjoint (arc-disjoint) cycles in the digraph under consideration. We will also consider the question of decreasing the number of cycles by reversing arcs.

In a digraph D, a set S of vertices (arcs) is a **feedback vertex set** (an **feedback arc set**) if $D - S$ is acyclic. The minimum number of elements in a feedback vertex (arc) set of D is denoted by $\boldsymbol{\tau_0(D)}$ ($\boldsymbol{\tau_1(D)}$). Notice that the parameters $\tau_0(D)$ and $\tau_1(D)$ have several practical applications, one of the most important is testing electronic circuits (see Leiserson and Saxe [639]). An electronic circuit can be modelled by a directed graph by letting each (boolean) gate correspond to a vertex and the wires into each gate be modelled by arcs into the vertex corresponding to that gate. Finding a small set of arcs whose removal makes the resulting digraph acyclic can help reduce the hardware overhead needed for testing the circuit using so-called scan registers (see Kunzmann and Wunderlich [631]).

Let $\pi = v_1, v_2, \ldots, v_n$ be an ordering of the vertices of a digraph D. An arc $v_iv_j \in A(D)$ is called **forward (backward) with respect to** π if $i < j$ ($i > j$); π is **τ-optimal** (or, simply **optimal**) if the number of backward arcs with respect to π is minimum among all orderings of vertices of D. The following proposition is folklore; we leave its proof as Exercise 15.1.

Proposition 15.0.1 *Let D be a digraph and let $S \subseteq A(D)$. Then the following claims are equivalents:*

(a) S is a feedback arc set of minimum size;
(b) There exists a τ-optimal ordering π of $V(D)$ such that S is the set of backward arcs with respect to π;
(c) $D - S$ is an acyclic subdigraph of D of maximum size. □

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_15,

Recall that for a digraph D, $\nu_0(D)$ ($\nu_1(D)$) denotes the maximum number of vertex-disjoint (arc-disjoint) cycles in D (see Section 13.3).

In Section 15.1 we state two conjectures on feedback arc sets, consider partial results and relations to other conjectures. Section 15.2 is devoted to optimal orderings of vertices in tournaments. Complexity of the feedback set problems is discussed in Section 15.3. In Section 15.4 we study relations between the parameters ν_0 and ν_1, on the one hand, and parameters τ_0 and τ_1, on the other hand. We state the famous Younger's conjecture and give an overview of the proof of this conjecture by Reed, Robertson, Seymour and Thomas. In Section 15.5 we give a proof of Seymour's Second Neighbourhood Conjecture for tournaments based on special orderings of vertices of tournaments. The proof is due to Havet and Thomassé. Section 15.6 is devoted to Ádám's conjecture on decreasing the number of cycles in a digraph by arc reversals.

15.1 Two Conjectures on Feedback Arc Sets

For a digraph D, let $\bar{m}(D)$ be the number of edges in the complement of $UG(D)$. Chudnovsky, Seymour and Sullivan proved the following upper bound for $\tau_1(D)$ for digraphs D with girth at least 4.

Theorem 15.1.1 *[217] If an oriented graph D has no 3-cycle, then $\tau_1(D) \leq \bar{m}(D)$.*

Proof: The proof is by induction on $n = |V(D)|$. Clearly, the theorem holds for $n \leq 2$ and assume that $n \geq 3$. For a vertex v of D, let $f(v)$ denote the number of induced paths of length 2 starting at v and let $g(v)$ denote the number of induced paths of length 2 in which v is the middle vertex. Since $\sum_{v \in V(D)} f(v) = \sum_{v \in V(D)} g(v)$, there is a vertex v such that $f(v) \leq g(v)$. Choose such a vertex v. Let $D_1 = D\langle N^+(v)\rangle$ and let $D_2 = D - N^+[v]$. Observe that $\bar{m}(D) \geq \bar{m}(D_1) + \bar{m}(D_2) + g(v)$. By the induction hypothesis, $\tau_1(D_1) \leq \bar{m}(D_1)$ and $\tau_1(D_2) \leq \bar{m}(D_2)$; for $i = 1, 2$ let X_i be a feedback arc set of D_i of minimum size. Let $X_3 = \{ab \in A(D) : \ a \in N^+(v),\ b \notin N^+(v)\}$; clearly $|X_3| = f(v)$. Since D has no 3-cycle, no arc of X_3 has its head in $N^-(v)$. Thus, $D - X$ is acyclic, where $X = X_1 \cup X_2 \cup X_3$. Hence

$$\begin{aligned} \tau_1(D) \leq |X| = |X_1| + |X_2| + |X_3| = \\ \tau_1(D) + \tau_2(D) + f(v) \leq \\ \bar{m}(D_1) + \bar{m}(D_2) + g(v) \leq \bar{m}(D). \end{aligned}$$

□

Chudnovsky, Seymour and Sullivan believed that the bound in Theorem 15.1.1 is not sharp and suggested the following:

Conjecture 15.1.2 *[217] If an oriented graph D has no 3-cycle, then we have $\tau_1(D) \leq \frac{1}{2}\bar{m}(D)$.*

Chudnovsky, Seymour and Sullivan [217] proved the conjecture for two cases: (a) $V(D)$ can be partitioned into two subsets Y and Z such that $D\langle Y\rangle$ and $D\langle Z\rangle$ are tournaments, (b) the vertices of D can be arranged in a circle such that if distinct vertices u, v, w are in clockwise order and $u \rightarrow w$, then $u \rightarrow v$ and $v \rightarrow w$. Kostochka and Stiebitz (see [217]) proved that in any minimal counterexample to the conjecture every vertex has at least 3 non-neighbours, and the conjecture holds for all oriented graphs of order at most 8.

The following conjecture appears to be very strong. According to Sullivan [837] who attributes it to Lichiardopol, the conjecture implies the Hoang-Reed, Caccetta-Häggkvist, Bermond-Thomassen and Thomassé conjectures (Conjectures 8.4.5, 8.4.1, 13.3.3 and 6.5.8, respectively).

Conjecture 15.1.3 *Every digraph D has some minimal feedback arc set that contains a path of length $\delta^+(D)$.*

15.2 Optimal Orderings in Tournaments

To illustrate the definitions above and to gain some understanding of difficulties in studying the feedback sets, let us consider the class of tournaments.

For a tournament T, let $\boldsymbol{\beta(T)}$ be the maximum number of arcs in an acyclic subdigraph of T. Fixing an arbitrary ordering $u_1, \ldots, u_n$ of vertices in T, we see that the number of forward arcs plus the number of backward arcs equals $\binom{n}{2}$. By replacing the ordering $u_1, u_2, \ldots, u_n$ by $u_n, u_{n-1}, \ldots, u_1$ if needed, we obtain that $\beta(T) \geq n(n-1)/4$. One may guess that we can always find an acyclic subdigraph of T of size exceeding $n(n-1)/4$ by a significant number, say, $\epsilon n(n-1)/4$, where ϵ is an absolute positive constant not depending on n. However, this is not true due to the following result due to Erdős and Moon [298]. Its proof is a modification of the original proof suggested by N. Alon.

Theorem 15.2.1 *For every $n \geq 3$, there exists a tournament T of order n such that $\beta(T) \leq n(n-1)/4 + \frac{1}{2}\sqrt{n^3 \log_e n}$.*

Proof: Consider a random tournament T_n on vertices $[n]$, i.e., a tournament chosen randomly from the set of all tournaments on $[n]$. Observe that for every pair $i \neq j \in [n]$, $ij \in A(T_n)$ with probability $1/2$.

For every pair $i < j \in [n]$, define the random variable x_{ij} by

$$x_{ij} = \begin{cases} +1 \text{ if } ij \in A(T_n) \\ -1 \text{ otherwise.} \end{cases}$$

Let $N = \binom{n}{2}$. With respect to the ordering $\pi = 1, 2, \ldots, n$, the number of forward arcs minus the number of backward arcs equals

$$S_N = \sum_{1 \le i < j \le n} x_{ij}.$$

Then, $E_\pi = \{|S_N| > a\}$ denotes the event that, in one of the two orderings $\pi = \pi(1), \pi(2), \ldots, \pi(n) (= 1, 2, \ldots, n)$ and $\pi^* = \pi(n), \pi(n-1), \ldots, \pi(1) (= n, n-1, \ldots, 1)$, the number of forward arcs exceeds $n(n-1)/4 + a/2$. On the other hand, S_N is the sum of $\binom{n}{2}$ random independent variables taking values $+1$ and -1, each with probability $1/2$. By Corollary A.1.2 in [29],

$$\text{Prob}(|S_N| > a) \le 2e^{-a^2/(2N)}, \tag{15.1}$$

for every positive number a.

Observe that the event E that for at least one permutation of $1, 2, \ldots, n$, the number of forward arcs exceeds $n(n-1)/4 + a/2$ equals the union of the events E_ν for all permutations of $1, 2, \ldots, n$, whose total number is $n!$. Put $a = \sqrt{n^3 \log_e n}$. Applying (15.1) we obtain

$$\begin{aligned} \text{Prob}(E) &\le 2n! \exp(-n \log_e n) \\ &\le 2n! n^{-n} < 1 \end{aligned}$$

for every $n \ge 3$. This means that with positive probability the event E does not hold, i.e., for every permutation of $1, 2, \ldots, n$, the number of forward arcs does not exceed $n(n-1)/4 + \sqrt{n^3 \log_e n}/2$. By the definition of T_n, it follows that there exists a tournament of order n with the above-mentioned property. □

A slightly better result was obtained by de la Vega in [254] who proved that $\sqrt{\log_e n}$ in the inequality of Theorem 15.2.1 can be replaced by a constant. Alon and Spencer [29] describe an explicit construction of tournaments T for which $\beta(T) \le n(n-1)/4 + O(n^{3/2} \log n)$.

15.3 Complexity of the Feedback Set Problems

In Subsection 15.3.1 we consider $\mathcal{NP}$-completeness results including a proof of a conjecture by Bang-Jensen and Thomassen that the problem of finding a smallest feedback arc set in a tournament is $\mathcal{NP}$-complete. In Subsection 15.3.2 we prove that the smallest feedback arc set problem is polynomial time solvable for planar digraphs. Approximation algorithms for the feedback arc problem are discussed in Section 15.3.3. Finally, in Subsection 15.3.4 we consider a recent result by Chen, Liu, Lu, O'Sullivan and Razgon [205] that the problem of finding a smallest feedback vertex set in a digraph is fixed-parameter tractable.

15.3.1 $\mathcal{NP}$-Completeness Results

The following result can be proved similarly to Proposition 13.3.1.

Proposition 15.3.1 *For every digraph D there exist digraphs D' and D'' such that $\tau_0(D) = \tau_1(D')$ and $\tau_1(D) = \tau_0(D'')$. The digraphs D' and D'' can be constructed from D in polynomial time.* □

This proposition implies that the following problems are of the same complexity (up to a polynomial factor).

FVS Problem: Given an integer k and a digraph D, is $\tau_0(D) \leq k$?
FAS Problem: Given an integer k and a digraph D, is $\tau_1(D) \leq k$?

Karp [585] was the first to prove the following theorem:

Theorem 15.3.2 *The FAS problem is $\mathcal{NP}$-complete.* □

Gavril [396] proved that the FAS problem remains $\mathcal{NP}$-complete even for digraphs D with $\Delta^0(D) \leq 3$ or line digraphs. Proposition 15.3.1 and Theorem 15.3.2 imply immediately that the FVS problem is $\mathcal{NP}$-complete. The FVS problem remains $\mathcal{NP}$-complete for digraphs D with $\Delta^0(D) \leq 2$, planar digraphs D with $\Delta^0(D) \leq 3$ (see Garey and Johnson [393]) and for line digraphs (see Gavril [396]). This problem, unlike the FAS problem, is $\mathcal{NP}$-complete even for undirected graphs [393].

Bang-Jensen and Thomassen [118] and Speckenmeyer [831] proved the following:

Theorem 15.3.3 *The FVS problem is $\mathcal{NP}$-complete for tournaments.* □

Bang-Jensen and Thomassen [118] conjectured that the FAS problem is also $\mathcal{NP}$-complete for tournaments. Interestingly, this conjecture was independently solved by several groups of researchers almost at the same time, see Alon [18], Charbit, Thomassé and Yeo [197] and Conitzer [227]. In fact, Ailon, Charikar and Newman [15] were the first to find a randomized reduction from FAS on digraphs to FAS on tournaments and the approaches of [18] and [197] can be viewed as deterministic variations of this randomized reduction.

In the remainder of this subsection, we present the proof from [197] that FAS for tournaments is $\mathcal{NP}$-complete.

Recall that the Cauchy-Schwarz inequality is often formulated as follows: $(u, v) \leq \sqrt{(u, u)(v, v)}$, where u and v are n-dimensional vectors and $(x, y) = \sum_{i=1}^{n} x_i y_i$ when $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$.

Let z be a positive integer and let $k = 2^z$. We denote by A the $k \times k$ matrix whose rows and columns are indexed by the subsets F_i of $\{1, 2, \ldots, z\}$ (in any order) and whose entries are $a_{ij} = (-1)^{|F_i \cap F_j|}$.

Lemma 15.3.4 *For any subset J of r columns of A, we have*

$$\sum_{i=1}^{k} |\sum_{j\in J} a_{ij}| \leq k\sqrt{r}.$$

Proof: Let $p \neq q$. Observe that $a_{ip}a_{iq} = (-1)^{2|F_i\cap F_p\cap F_q|+|F_i\cap(F_p\Delta F_q)|}$. Since there are equal numbers of subsets F_i for which $|F_i \cap (F_p \Delta F_q)|$ is even and odd, $\sum_{i=1}^{k} a_{ip}a_{iq} = 0$ when $p \neq q$. Now by the Cauchy-Schwarz inequality,

$$\sum_{i=1}^{k} \frac{|\sum_{j\in J} a_{ij}|}{k} \leq \sqrt{\frac{\sum_{i=1}^{k}(\sum_{j\in J} a_{ij})^2}{k}} = \sqrt{\frac{\sum_{i=1}^{k}(\sum_{p\in J} a_{ip}^2 + 2\sum_{p\neq q\in J} a_{ip}a_{iq})}{k}} = \sqrt{r}.$$

□

Lemma 15.3.5 *Suppose that z is divisible by 3. Let $B = (b_{ij})$ be the matrix obtained from A by an arbitrary permutation of the columns and let q_i be defined as follows: $q_i = \max\{|\sum_{j=1}^{p} b_{ij}| : \ p \in [k]\}$. Then $\sum_{i=1}^{k} q_i \leq 2k^{5/3}$.*

Proof: Define the integers $l = k^{2/3}$ and $s = k^{1/3}$. For all $i \in [k]$ and $j \in [s]$, we let $c_i^j = |\sum_{j'=(j-1)l+1}^{jl} b_{ij'}|$. By Lemma 15.3.4 we have $\sum_{i=1}^{k} c_i^j \leq k\sqrt{l}$ for all $j \in [s]$. Therefore $\sum_{i=1}^{k}\sum_{j=1}^{s} c_i^j \leq ks\sqrt{l} = k^{5/3}$.

We now bound q_i. Assume that p is defined so that $q_i = |\sum_{j'=1}^{p} b_{ij'}|$. Let j be such that $(j-1)l \leq p < jl$. Note that $q_i \leq c_i^1 + c_i^2 + \ldots + c_i^{j-1} + l$ since $|\sum_{i=(j-1)l+1}^{p} b_{ij}| \leq l$. Thus $\sum_{i=1}^{k} q_i \leq (\sum_{i=1}^{k}\sum_{j=1}^{s} c_i^j) + kl \leq 2k^{5/3} \leq l$. □

We leave the proof of the following lemma as Exercise 15.2.

Lemma 15.3.6 *Every column of A has $k/2$ positive entries, except the empty set column which has k.* □

Theorem 15.3.7 *Let z be divisible by 3. There exists a bipartite tournament G_k in which partite sets both have k vertices ($|V(G_k)| = 2k$) and $\tau_1(G_k) \geq \frac{k^2}{2} - 2k^{5/3}$. Furthermore, we can construct G_k in polynomial time.*

Proof: Let the partite sets of G_k be $\{r_1, r_2, \ldots, r_k\}$ and $\{s_1, s_2, \ldots, s_k\}$, respectively. Now add an arc from r_i to s_j if $a_{ij} = -1$ in A, and add an arc from s_j to r_i if $a_{ij} = 1$ in A. This clearly defines a bipartite tournament, which can be constructed in polynomial time. Let π be a τ-optimal ordering of $V(G_k)$. Without loss of generality, we may assume that the order of the s_j's in π is $s_1, s_2, \ldots, s_k$. Let $i \in [k]$ be arbitrary and define p such that $s_1, s_2, \ldots, s_p$ come before r_i in π and $s_{p+1}, s_{p+2}, \ldots, s_k$ come after r_i in π. Let m_i denote the number of positive entries of A in row i and note that the number of backward arcs incident with r_i is as follows:

$$|\{a_{ij} : a_{ij} = -1, j \leq p\}| + |\{a_{ij} : a_{ij} = 1, j > p\}| =$$
$$|\{a_{ij} : a_{ij} = -1, j \leq p\}| + (m_i - |\{a_{ij} : a_{ij} = 1, j \leq p\}|) =$$
$$m_i - \sum_{j=1}^{p} a_{ij}.$$

Thus, by Lemmas 15.3.5 and 15.3.6, $\tau_1(G_k) \geq \frac{k(k+1)}{2} - 2k^{5/3} > \frac{k^2}{2} - 2k^{5/3}$. □

Theorem 15.3.8 *The FAS problem for tournaments is $\mathcal{NP}$-complete.*

Proof: We will use a reduction from FAS for arbitrary digraphs to FAS for tournaments. Let D be an arbitrary digraph with $V(D) = \{v_1, v_2, \ldots, v_n\}$. We may assume that D has no cycles of length two, as deleting such a cycle decreases the minimum feedback arc set by exactly one. Set $k = 2^{6\lceil 1+log_2 n\rceil}$. Note that $k = O(n^6)$ and $k \geq 64n^6$. Let G_k be the bipartite tournament built in Theorem 15.3.7 with partite sets $\{r_1, r_2, \ldots, r_k\}$ and $\{s_1, s_2, \ldots, s_k\}$.

We now construct a tournament T with vertex set $\{w_i^j : i \in [n], j \in [k]\}$ and the arc set described below. Let $a, b \in [n]$ and $i, j \in [k]$ be arbitrary. The arc between vertices w_a^i and w_b^j in T is oriented as follows:

(a) $w_a^i w_b^j \in A(T)$ if $a = b$ and $i < j$;
(b) $w_a^i w_b^j \in A(T)$ if $v_a v_b \in A(D)$;
(c) If v_a and v_b have no arc between them in D and $a < b$, then $w_a^i w_b^j \in A(T)$ if $r_i s_j \in A(G_k)$ and $w_b^j w_a^i \in A(T)$ if $s_j r_i \in A(G_k)$.

We will now bound $\tau_1(T)$ from both above and below. Without loss of generality, assume that $v_1, v_2, \ldots, v_n$ is a τ-optimal ordering of $V(D)$. Note that Theorem 15.3.7 implies that the arcs generated by (c) above will always contribute between $(\binom{n}{2} - |A(D)|)(\frac{k^2}{2} - 2k^{5/3})$ and $(\binom{n}{2} - |A(D)|)(\frac{k^2}{2} + 2k^{5/3})$ to $\tau_1(T)$. Now consider the following ordering of the vertices in T:

$$w_1^1, w_1^2, \ldots, w_1^k, w_2^1, w_2^2, \ldots, w_2^k, \ldots, w_n^1, w_n^2, \ldots, w_n^k.$$

This ordering implies the following bound on $\tau_1(T)$:

$$\tau_1(T) \leq k^2 \tau_1(D) + \left(\binom{n}{2} - |A(D)|\right)\left(\frac{k^2}{2} + 2k^{5/3}\right).$$

In order to bound $\tau_1(T)$ from below let π be a τ-optimal ordering of the vertices in T. Let $i_1, i_2, \ldots, i_n$ be an arbitrary collection of n integers from $[k]$ and note that there are at least $\tau_1(D)$ arcs between vertices in $\{w_1^{i_1}, w_2^{i_2}, w_3^{i_3}, \ldots, w_n^{i_n}\}$ which are backward arcs in π, as this set of vertices induce a digraph isomorphic to D. By summing over all possible values of $i_1, i_2, \ldots, i_n$ we get $k^n \tau_1(D)$ backward arcs, where each arc can be counted at most k^{n-2} times. This implies the following:

$$\tau_1(T) \geq \frac{k^n \tau_1(D)}{k^{n-2}} + \left(\binom{n}{2} - |A(D)|\right)\left(\frac{k^2}{2} - 2k^{5/3}\right).$$

Since $k^{1/3} \geq 64^{1/3}n^2 = 4n^2$, the above two bounds imply the following:

$$\tau_1(D) - \frac{1}{2} < \frac{\tau_1(T)}{k^2} - \frac{1}{2}\left(\binom{n}{2} - |A(D)|\right) < \tau_1(D) + \frac{1}{2}.$$

So if we could compute $\tau_1(T)$ in polynomial time, we would also be able to compute $\tau_1(D)$. Since our reduction is polynomial, we are done. □

15.3.2 FAS for Planar Digraphs

While for arbitrary digraphs and even for tournaments the FAS problem is $\mathcal{NP}$-complete (as we saw above), for planar digraphs the situation is quite different (unless $\mathcal{P} = \mathcal{NP}$) due to the following result by Lucchesi:

Theorem 15.3.9 *[659] The FAS problem is polynomially solvable for planar digraphs.*

We give a proof of Theorem 15.3.9 below. First we need the definition of the dual of a plane directed pseudograph. Let $G = (V, E)$ be a planar pseudograph and let F be the set of faces of G (with respect to the fixed planar embedding of G). Let G^* be the pseudograph which has a vertex v_i for each face $f_i \in F$ and for every edge $e \in E$ such that e is on the boundary of faces f_i, f_j, the two vertices v_i, v_j corresponding to f_i, f_j are joined by an edge[1]. In general G^* contains parallel edges and may also contain loops. For plane directed pseudographs we can also define a dual called the **directed dual**. This is the same as above but now the orientation of the arc between v_i and v_j is always chosen such that the arc crosses the original arc e from left to right (here left means the left side when we traverse e from its tail to its head). See Figure 15.1 for an example of the dual of a directed multigraph.

If $D = (V, A)$ is a plane directed multigraph and D^* is its directed dual, then it is easy to see that D^* is also planar (Exercise 15.3). In fact, we have that $(D^*)^*$ is isomorphic to the converse of D (Exercise 15.4).

Proof of Theorem 15.3.9: Let D be a planar directed multigraph and assume that D is embedded in the plane with directed dual D^*. Clearly we may assume that $UG(D)$ is connected since otherwise we just consider each connected component separately.

We prove that the size of a minimum feedback set of D is equal to the minimum size of a dijoin of D (see the definition of a dijoin in Section 13.1). Recall from Section 13.1 that this is also the minimum number of arcs whose contraction results in a strongly connected directed multigraph.

[1] Note that if e is not part of the boundary of a facial cycle, then $f_i = f_j$ and we get a loop at v_i.

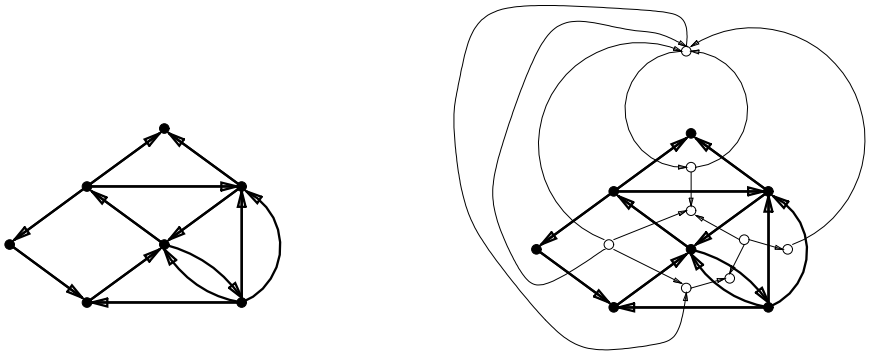


Figure 15.1 (a) A plane directed multigraph D; (b) the directed dual D^* of D drawn on top of D. White circles indicate the vertices of D^* and thin arcs are arcs of D^*. Fat arcs indicate arcs of D.

If we delete an arc a of D the effect on the dual will be the same as if we contract the corresponding dual arc a^* (the one crossing a from left to right). If C is a facial cycle of D, then the vertex v corresponding to C has all arcs directed into it or out of it (depending on whether the orientation of C is clockwise or anti-clockwise). Thus in D^* the arcs incident with v form a directed cut (recall the definition of a directed cut from Section 13.1) in D^* implying that D^* is not strong.

Conversely, if D^* is not strongly connected, then let H be an initial strong component (that is, there is no arc from $V - V(H)$ to $V(H)$ in D) of D^*. Now it is not difficult to see that the arcs of D corresponding to the directed cut $(V(H), V - V(H))$ in D^* (which is non-empty since D is connected) form a directed cycle (Exercise 15.5). Thus we have shown that D has a directed cycle if and only if D^* is not strongly connected. Furthermore, deleting arcs of D until we obtain an acyclic directed multigraph is equivalent to contracting arcs of D^* until we obtain a strong directed multigraph. This shows that the size of a minimum feedback arc set of D equals the size of a minimum directed join in D^*. Now it follows from Corollary 13.1.5 that we can find the feedback number (and a minimum feedback arc set) of D in polynomial time. □

Our arguments above imply the following:

Corollary 15.3.10 *For a planar digraph D, $\nu_1(D) = \tau_1(D)$.* □

15.3.3 Approximation Algorithms

We mentioned above that in several applications one wishes to find a minimum (cardinality) feedback arc set. From now on the following problem

will be called the FEEDBACK ARC SET (FAS) Problem [2]: given a digraph $D = (V, A)$, find an ordering $s = v_1, v_2, \ldots, v_n$ of V with minimum number of **backward arcs**.

Observe that if A' is an arbitrary feedback arc set, then by definition $D - A'$ is acyclic and hence has an acyclic ordering $v_1, v_2, \ldots, v_n$. With respect to this ordering every arc $v_i v_j \in A - A'$ satisfies $i < j$. Hence, from the algorithmic point of view, finding a minimum feedback arc set in D is equivalent to finding an ordering $u_1, u_2, \ldots, u_n$ of V which maximizes (minimizes) the number of forward arcs (backward arcs); an arc $u_i u_j$ is **forward** with respect to the above ordering if $i < j$, otherwise $u_i u_j$ is **backward**[3]. This again is easily seen to be (algorithmically) equivalent to finding an acyclic subdigraph with the maximum number of arcs in D (Exercise 15.7) . The latter problem is known as the MAXIMUM ACYCLIC SUBDIGRAPH (MAS) PROBLEM. Arora, Frieze and Kaplan [50] and Frieze and Kannan [362] designed polynomial-time approximation scheme (PTAS) for the MAS Problem restricted to dense digraphs and, in particular, to tournaments. (For a minimization problem $\mathcal{Q}$, PTAS is an algorithm which takes an instance of $\mathcal{Q}$ and a parameter $\epsilon > 0$ and, in polynomial time, produces a solution that is within a factor $1 + \epsilon$ of being optimal.)

One may also consider weighted versions of the problems above. Each arc is assigned a non-negative real-valued weight and the goal is to find a feedback arc set of minimum total weight (respectively, an acyclic subdigraph of maximum weight). The weighted version of the acyclic subdigraph problem is known as the LINEAR ORDERING PROBLEM. It arises naturally in the study of interactions between various sectors of an economical system (see Reinelt [775] and also Funke and Reinelt [367] and Grötschel, Jünger and Reinelt [428]).

For the Linear Ordering Problem there is a very easy way to obtain an ordering which achieves at least half of the optimum value of an ordering. The proof of the following proposition is an easy exercise (Exercise 15.8).

Proposition 15.3.11 *Given any weighted digraph $D = (V, A, w)$, in time $O(m)$ one can find an acyclic subdigraph $D' = (V, A')$ of D such that $w(A') \geq w(A)/2$.* □

This proposition implies that there exists a polynomial 2-approximation algorithm for the Linear Ordering Problem, since $w(A)/2 \leq w(A') \leq w(A_o) \leq w(A)$, where $w(A_o)$ is the optimum weight.

Note that although the Linear Ordering Problem and the FAS Problem are equivalent problems from the algorithmic point of view, the approximation algorithm above cannot be used as a 2-approximation algorithm for the FAS Problem as well. The reason is that the optimal ordering may have all or almost all arcs in the right direction (implying the number τ_1 is close to

[2] This is the optimization version of the decision FAS problem considered earlier.
[3] Clearly, the set of backward arcs forms a feedback arc set.

zero) whereas the ordering above may still have as little as half the arcs in the right direction. In fact, approximating the number τ_1 seems to be very difficult and so far no c-approximation algorithm is known for any constant c. The following is the best known approximation guarantee for the FAS Problem. It is due to Seymour [811].

Theorem 15.3.12 *[811] There exists an $O(\log n \log\log n)$ approximation algorithm for the FAS Problem.* □

Another approximation algorithm for a generalization of the FAS Problem (as well as the Feedback Vertex Set Problem) is described by Even, Naor, Schieber and Sudan [304]. Kenyon-Mathieu and Schudy [591] obtained a polynomial-time approximation scheme (PTAS) for the FAS Problem restricted to tournaments. Notice that a PTAS is not possible for the FAS Problem on all digraphs which follows from an in-approximability result of Dinur and Safra [265].

15.3.4 Fixed-Parameter Tractability Results

It has been a challenging open problem to decide whether the Parameterized FVS and FAS Problems are fixed-parameter tractable (FPT). See Section 18.4 for definitions related to FPT and the survey paper [482] by Gutin and Yeo for the history of the problem prior to its solution by Chen, Liu, Lu, O'Sullivan and Razgon [205]. Given a digraph D and a parameter k, the Parameterized FVS (FAS) Problem is the problem of checking whether $\tau_0(D) \leq k$ ($\tau_1(D) \leq k$). By Proposition 15.3.1, if one of the two parameterized problems is FPT, then so is the other one. This allows us to conclude that FAS is FPT after proving that the parameterized FVS is FPT.

Chen, Liu, Lu, O'Sullivan and Razgon proved the following:

Theorem 15.3.13 *The* Parameterized FVS Problem *is FPT.* □

The rest of the section is based on the paper [765] by Razgon. We provide some ideas behind the proof of Theorem 15.3.13. Let D be an acyclic digraph, and let $X = \{x_1, x_2, \ldots, x_l\}$ and $Y = \{y_1, y_2, \ldots, y_l\}$ be a pair of disjoint sets of vertices called **terminals**. A set R of non-terminal vertices **orderly separates** X from Y if $D-R$ has no path from x_i to y_j for all i, j with $1 \leq j \leq i \leq l$. The problem of checking whether there is a set R of cardinality at most k that orderly separates X from Y is called the Parameterized Ordered Multicut Problem for Acyclic Digraphs (abbreviated POMC-AD). To keep the subsection relatively short, we will assume that there is an FPT algorithm to solve POMC-AD (such an algorithm is described by Razgon [765]) and we will prove that the Parameterized FVS is FPT using a reduction from the Parameterized FVS to POMC-AD.

The reduction is based on the principle of **iterative compression** that has been used to design non-trivial FPT algorithms for various parameterized problems. Let $v_1, v_2, \ldots, v_n$ be vertices of a digraph D. The reduction

iteratively generates a sequence $D_0, D_1, \ldots, D_n$ of digraphs, where D_0 is the empty digraph and $D_i = D\langle\{v_1, v_2, \ldots, v_i\}\rangle$ for each $i \geq 1$. For each $i \geq 0$, the reduction produces a feedback vertex set S_i of D_i with $|S_i| \leq k$ or returns 'NO' if D_i has no feedback vertex set of cardinality at most k.

The sets S_i are computed recursively. In particular, $S_0 = \emptyset$. For each S_i, $i \geq 1$, if S_{i-1} is a feedback vertex set for D_i, then we set $S_i = S_{i-1}$. Otherwise, if $|S_{i-1}| \leq k-1$, we set $S_i = S_{i-1} \cup \{v_i\}$. Finally, if S_{i-1} is not a feedback vertex set of D_i and $|S_{i-1}| = k$, then we set $S'_i = S_{i-1} \cup \{v_i\}$ and try to find a feedback vertex set S_i of D_i of cardinality smaller than $|S'_i| = k+1$. In particular, for each subset F of S'_i, we apply a procedure denoted by Replace$(D_i - F, S'_i - F)$ whose input consists of $D_i - F$ and $S'_i - F$ and output is either a feedback set F' of $D_i - F$ of cardinality smaller than $|S'_i| - |F|$ and such that $F' \cap S'_i = \emptyset$ or 'NO' if no such feedback vertex set exists. If we succeed to find at least one such F', then we set $S_i = F \cup F'$. Otherwise, we return 'NO'. In other words, we guess all possibilities of $F = S'_i \cap S_i$.

It remains to describe Replace(H, S), where $H = D_i - F$ and $S = S'_i - F$ and prove its correctness and complexity.

Replace(H, S)
Input: a digraph H and a feedback vertex set S of H; $c = |S|$.
Output: a feedback vertex set R of H with $R \cap S = \emptyset$ and $|R| < c$ or 'NO' if no such feedback vertex set R exists.

1. If H is acyclic then return $\emptyset$.
2. If $H\langle S\rangle$ is not acyclic then return 'NO'.
3. Construct $A_S := \{uv \in A(H) : v \in S\}$.
4. For each possible ordering $s_1, s_2, \ldots, s_c$ of the vertices of S do the following three steps:
   4a. For each s_i find the set T_i of vertices w of $H - S$ such that $H\langle A_S\rangle$ has a path from w to s_i;
   4b. Construct a digraph H' by adding a set $T = \{t_1, t_2, \ldots, t_c\}$ of new vertices to $H - A_S$ and by adding the arc wt_i for each $w \in T_i$ (Steps 3, 4a and 4b are illustrated in Figure 15.2);
   4c. Solve POMC-AD for the digraph H', terminals S and T and parameter $k' = c - 1$; if the solution R is not 'NO', return R.
5. Return 'NO'.

It is not difficult to see that Replace(H, S) is an FPT algorithm provided POMC-AD admits an FPT algorithm. Let us show the correctness of Replace(H, S).

Theorem 15.3.14 *If Replace(H, S) returns a set R, then R satisfies the output specifications and if 'NO' is returned, then no set R satisfying the output specifications exists.*

Proof: Assume first that Replace(H, S) returns a set R. This means that there is an ordering $s_1, s_2, \ldots, s_c$ of S such that R orderly separates S from

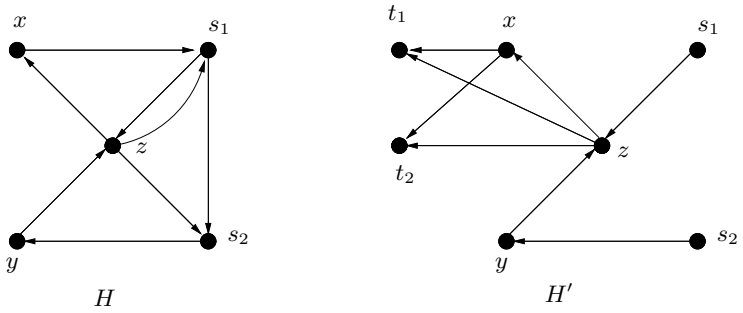


Figure 15.2 Illustrating Steps 3, 4a and 4b of Replace(H, S). We have $S = \{s_1, s_2\}$, $A_S = \{xs_1, zs_1, zs_2, s_1s_2\}$ and $T_1 = T_2 = \{x, z\}$.

T in H'. By definition, $R \subseteq V(H) \setminus S$. Suppose that R is not a feedback vertex set of H and let Z be a cycle in $H - R$.

By definition of A_S, the digraph $H - A_S$ is acyclic. Thus, Z contains arcs in A_S. Partition arcs in $A_S \cap A(Z)$ into maximal paths $P_1, P_2, \ldots, P_l$, where the ordering is according to their appearance on Z. It follows from the definition of A_S that the terminal vertex of each P_i is some vertex $s_{i_j} \in S$. By Step 2 of Replace(H, S), the initial vertex of each P_i is some $w_i \in T_{i_j}$. Since $H\langle A_S\rangle$ is acyclic, the union of the paths $P_1, P_2, \ldots, P_l$ does not form Z. Moreover, Z includes a path (in $H - R - A_S$) from s_{j_1} to a vertex of T_{j_2}, ..., a path from s_{j_l-1} to a vertex of T_{j_l} and a path from s_{j_l} to a vertex of T_{j_1}. We have $(j_1 \geq j_2) \vee \ldots \vee (j_{l-1} \geq j_l) \vee (j_l \geq j_1)$ as otherwise we get $j_1 < j_2 < \ldots < j_l < j_1$, a contradiction. Thus, $H - A_S - R$ has a path from some s_i to a vertex of T_j such that $i \geq j$. Therefore, the digraph $H' - R$ has an (s_i, t_j)-path contradicting our assumption that R orderly separates S from T in H'. Thus, we conclude that R is a feedback vertex set of H.

It remains to prove that if R is a feedback vertex set of H such that $S \cap R = \emptyset$ and $|R| \leq |S| - 1$, then R orderly separates S from T in H' for at least one ordering $s_1, s_2, \ldots, s_c$. Let R be a feedback vertex set of H such that $S \cap R = \emptyset$ and $|R| \leq |S| - 1$ and let $s_1, s_2, \ldots, s_c$ be an ordering of S. Let $t_1, t_2, \ldots, t_c$ and H' be as described in Replace(H, S). We partition the proof into two claims.

Claim 1. For each i, $H' - R$ has no (s_i, t_i)-path.

Proof: Suppose this is not true, i.e., there is an i such that P is an (s_i, t_i)-path in $H' - R$ and let w be the predecessor of t_i on P. By definition of H', $P'' = P[s_i, w]$ is a path in $H - R$. Also by definition of H', $w \in T_i$ and $H\langle A_S\rangle$ has a (w, s_i)-path P'. Observe that $w \notin R$ (as w belongs to a path in $H - R$) and all other vertices of P' are in S and, thus, not in R. Thus, P' is a path in $H - R$. The union $P' \cup P''$ is a closed walk in $H - R$ implying that $H - R$

has a cycle, a contradiction to the assumption that R is a feedback vertex set in H. □

Claim 2. Let l be an integer such that $1 \leq l \leq c$. Then there is an integer p, $1 \leq p \leq l$, such that $H' - R$ has no (s_p, t_i)-path for any $i \in [l]$.

Proof: Suppose that this claim is not true, i.e., for each integer p, $1 \leq p \leq l$, the digraph $H' - R$ has an (s_p, t_i)-path for some $i \in [l]$.

Fix an arbitrary p, $1 \leq p \leq l$. By Claim 1, $H' - R$ has no (s_p, t_p)-path. Thus, there exists an integer $z(p) \in [l]$ such that $H' - R$ has an $(s_p, t_{z(p)})$-path Q_p. Consider a sequence $p_0, p_1, \ldots, p_l$, where $p_0 = p$ and $p_i = z(p_{i-1})$, $1 \leq i \leq l$. Clearly, the sequence has at least two equal elements. Without loss of generality, we may assume that $p_0 = p_y$, where $1 \leq y \leq l$. For each $i \in \{0, 1, \ldots, y-1\}$, let $Q'_{p_i} = Q_{p_i} - t_{z(p)}$. By definition of H', Q'_{p_i} is a path in $H - R$ terminating in a vertex $w_{p_{i+1}} \in T_{p_{i+1}}$.

Let $Q''_1, Q''_2, \ldots, Q''_y$ be paths in $H\langle A_s \rangle$ such that each Q''_i is a (w_{p_i}, s_{p_i})-path. As in the proof of Claim 1, we can see that each Q''_i is a path in $H - R$. Observe that the union of $Q'_{p_0}, Q''_1, Q'_{p_1}, Q''_2, \ldots, Q'_{p_{y-1}}, Q''_y$ is a walk in $H - R$. This walk is closed since its initial vertex s_{p_0} coincides with its terminal vertex s_{p_y}. Thus, $H - R$ has a cycle, a contradiction. □

Now we will produce the desired ordering starting from our ordering $s_1, s_2, \ldots, s_c$. By Claim 2, there exists $p \in [c]$ such that s_p has no path to any t_i in H'. If $p \neq c$, we swap s_p and s_c in the ordering (here and below, if two terminals in S interchange, the corresponding terminals in T also interchange). Assume that the last $c - l$ vertices in the ordering of S have been fixed. If $l = 1$, by Claim 1, we are done. Otherwise, find p, $1 \leq p \leq l$, satisfying Claim 2. If $p \neq l$, swap s_p and s_l in the ordering. □

15.4 Disjoint Cycles Versus Feedback Sets

In this section, we study relations between the parameters ν_0 and ν_1, on the one hand, and parameters τ_0 and τ_1, on the other hand. We state the famous Younger's conjecture and present an overview of the proof of this conjecture due to Reed, Robertson, Seymour and Thomas. Some (still) open conjectures and problems are mentioned as well.

15.4.1 Relations Between Parameters ν_i and τ_i

Clearly, for every digraph D, $\nu_0(D) \leq \nu_1(D)$ and it is easy to find an infinite family of digraphs D for which the two parameters are not equal. The same is true for the parameters τ_0, τ_1. Furthermore, we obviously have $\nu_i(D) \leq \tau_i(D)$ for $i = 0, 1$. It is easy to construct an infinite family of digraphs D such that $\nu_0(D) < \tau_0(D)$ (Exercise 15.12) and thus, by Propositions 13.3.1 and 15.3.1, an infinite family of digraphs D such that $\nu_1(D) < \tau_1(D)$.

On the other hand, there are families of digraphs for which the last two inequalities become equalities. Szwarcfiter [840] described a family of digraphs D for which $\nu_0(D) = \tau_0(D)$. His family generalizes two families introduced by Frank and Gyárfás [350] and by Wang, Floyd and Soffa [898]. Szwarcfiter [840] also provides polynomial algorithms to recognize his family of digraphs and to find k-cycle factors and feedback vertex sets of cardinality k, where $k = \nu_0(D) = \tau_0(D)$. We have already seen that planar digraphs D satisfy $\nu_1(D) = \tau_1(D)$. Seymour [812] showed that the same result holds for a special family of eulerian digraphs. Another class of digraphs with the same property was considered by Ramachandran [760].

Even though not always $\nu_i(D) = \tau_i(D)$, $i = 0, 1$ (in which case $\tau_i(D)$ exceeds $\nu_i(D)$), Younger [922] conjectured that the former is bounded by a function of the latter. In other words, he conjectured that for every k, there exists a (least) natural number $\boldsymbol{t_0(k)}$ ($\boldsymbol{t_1(k)}$, respectively) such that for every digraph D the following holds: either D contains k vertex-disjoint (arc-disjoint, respectively) cycles or D has a feedback vertex (arc, respectively) set of cardinality at most $t_0(k)$ ($t_1(k)$, respectively). By Propositions 13.3.1 and 15.3.1, the validity of the 'vertex' version of Younger's conjecture implies that the 'arc' version holds and vice versa. Moreover, Propositions 13.3.1 and 15.3.1 imply that if the functions $t_0(k)$ and $t_1(k)$ exist, then they are equal (Exercise 15.13). Younger's conjecture was settled by Reed, Robertson, Seymour and Thomas [769]. We discuss their solution in the next subsection. In the rest of this subsection we consider the parameters ν_1 and τ_1 for the class of tournaments.

Even for a tournament T, the parameters $\nu_1(T)$ and $\tau_1(T)$ do not always coincide. By the proof of Theorem 15.2.1, for every $n \geq 3$ a random tournament T_n with n vertices, with probability tending to 1 as $n \to \infty$, has at least $n(n-1)/4 - \sqrt{n^3 \log_e n}/2$ arcs in a feedback arc set of T. On the other hand, it follows from a result by Chartrand, Geller and Hedetniemi [199] that T_n has at most $\lfloor \frac{n}{3} \lfloor \frac{n-1}{2} \rfloor \rfloor \leq \frac{1}{3}\binom{n}{2}$ arc-disjoint cycles (each cycle has at least three arcs). Isaak conjectured the following:

Conjecture 15.4.1 *[552] If T is a tournament which has a minimum feedback arc set A such that $T\langle A \rangle$ is a transitive subtournament of T, then $\nu_1(T)$ and $\tau_1(T)$ coincide.*

In [552] Isaak posed the following problem. Note that if the answer to the problem is yes, then this implies Conjecture 15.4.1.

Problem 15.4.2 *Suppose T is a tournament having a minimum feedback arc set which induces an acyclic digraph with a hamiltonian path. Is it true that the maximum number of arc-disjoint cycles in T equals the cardinality of a minimum feedback arc set of T?*

It is easy to see that a minimum feedback arc set of a given digraph must induce an acyclic subdigraph of D (Exercise 15.9). The next result by

Barthélémy, Hudry, Isaak, Roberts and Tesman implies that every acyclic digraph arises as a minimum feedback arc set of some tournament.

Theorem 15.4.3 *[125] Let D be an acyclic digraph. Then there exists a tournament T containing D as a subdigraph such that the arcs of D form a minimum feedback arc set in T.* □

15.4.2 Solution of Younger's Conjecture

The vertex and arc versions of Younger's conjecture were proved for various families of digraphs including the families mentioned above. McCuaig [689] proved the existence of $t_0(2)$ by characterizing **intercyclic digraphs**, i.e., digraphs D for which $\nu_0(D) \leq 1$. Moreover, he established that $t_0(2) = 3$. Reed and Shepherd [770] proved the vertex version of Younger's conjecture for planar digraphs using a result of Seymour [811]. The result of Reed and Shepherd combined with a result of Goemans and Williamson [414] implies that $t_0^{pd}(c) = O(c)$, where $t_0^{pd}(c)$ is the function $t_0(c)$ restricted to planar digraphs. Finally, Younger's conjecture was completely settled by Reed, Robertson, Seymour and Thomas [769]. In this subsection, we give a scheme of their proof. In particular, we provide a complete proof of perhaps the most interesting lemma in [769].

One of the important tools in the proof in [769] is the following well-known Ramsey Theorem [761].

Theorem 15.4.4 (Ramsey Theorem) *For all integers $q, l, r \geq 1$ there exists a (minimum) integer $R_l(r, q) \geq 0$ so that the following holds. Let Z be a set of cardinality at least $R_l(r, q)$ and let every l-subset of Z be assigned a colour from $[q]$. Then there exist an r-subset S of Z and a colour $k \in [q]$ so that every l-subset of S is of colour k.* □

Some readers may be more familiar with the graph-theoretic special case of this theorem. For every pair of natural numbers q, r there exists an integer $R_2(r, q) \geq 0$ so that every q-edge-coloured complete graph of order at least $R_2(r, q)$ has a monochromatic complete subgraph of order r.

We start describing the scheme of the proof of Younger's conjecture by the following lemma whose proof is left as Exercise 15.14.

Lemma 15.4.5 *[769] Let $c \geq 1$ be an integer such that $t_0(c-1)$ exists. Let D be a digraph with $\nu_0(D) < c$ and let T be a feedback vertex set of D of cardinality $\tau_0(D)$. Suppose U, W are disjoint subsets of T both of cardinality r, where $r \geq 2t_0(c-1)$. Then there is an r-path subdigraph of D from U to W, which contains no vertex in $T - (U \cup W)$.* □

Let $\mathcal{L} = P_1 \cup \ldots \cup P_k$ be a k-path subdigraph in a digraph D and let u_i (w_i) be the initial (terminal) vertex in P_i, $i \in [k]$. Recall that we say that $\mathcal{L}$ is a k-linkage from $(u_1, \ldots, u_k)$ to $(w_1, \ldots, w_k)$ and $\mathcal{L}$.

The following lemma was proved by the authors of [769] in joint work with Alon. Its proof uses Ramsey's theorem as well as Theorem 13.5.3 of Erdős and Szekeres.

Lemma 15.4.6 *[769] Let $c \geq 2$ be an integer such that $t_0(c-1)$ exists, and let $k \geq 1$ be an integer. Then there exists an integer $t \geq 0$ (depending on k) so that the following holds. If D is a digraph with $\nu_0(D) < c$ and $\tau_0(D) \geq t$, then there are distinct vertices $u_1, \ldots, u_k, w_1, \ldots, w_k$ of D and a pair of k-path subdigraphs $\mathcal{L}_1, \mathcal{L}_2$ of D so that*

(a) $\mathcal{L}_1$ is a k-linkage from $(u_1, \ldots, u_k)$ to $(w_1, \ldots, w_k)$,
(b) $\mathcal{L}_2$ is a k-linkage from $(w_1, \ldots, w_k)$ to either $(u_1, \ldots, u_k)$ or $(u_k, \ldots, u_1)$,
(c) every (directed) cycle of $\mathcal{L}_1 \cup \mathcal{L}_2$ meets $\{u_1, \ldots, u_k, w_1, \ldots, w_k\}$.

Proof: Let $l := (k-1)^2 + 1$, $r := \max\{2t_0(c-1), (k+1)l\}$, $q := (l! + 1)^2$, and $t := R_l(r, q) + l$, where $R_l(r, q)$ is as in Theorem 15.4.4. Then $r \geq l$ and $t \geq 2r$ as clearly $R_l(r, q) \geq 2r - 1$. We will show that this choice for t satisfies the lemma. Let D be a digraph satisfying $\nu_0(D) < c$ and $\tau_0(D) \geq t$. Choose a feedback vertex set T of D of cardinality $\tau_0(D)$ and an l-subset U of T. Let $Z := \{z_1, z_2, \ldots, z_{|Z|}\} := T - U$. Thus, $|Z| \geq R_l(r, q)$.

For each $X \subseteq Z$, with $X = \{z_{i_1}, \ldots, z_{i_{|X|}}\}$ where $i_1 < \ldots < i_{|X|}$; we put $\bar{X} := (z_{i_1}, \ldots, z_{i_{|X|}})$ and $\bar{X}(h) = z_{i_h}$ for $h = 1, \ldots, |X|$.

Let X be an l-subset of Z. If there is an l-path subdigraph $\mathcal{L}_1(X)$ in D from U to X containing no vertex in $Z - X$, then there is a permutation $(u_1, \ldots, u_l)$ of the vertices of U so that $\mathcal{L}_1(X)$ is an l-linkage from $(u_1, \ldots, u_l)$ to $\bar{X}$, and we put $p_1(X) := (u_1, \ldots, u_l)$; if no such path subdigraph exists, we put $p_1(X) := \emptyset$. Similarly, if there is an l-path subdigraph $\mathcal{L}_2(X)$ from X to U that is an l-linkage from $\bar{X}$ to $(w_1, \ldots, w_l)$ containing no vertex in $Z - X$, we put $p_2(X) := (w_1, \ldots, w_l)$; if no such linkage exists, we put $p_2(X) := \emptyset$. We assign to X the colour $(p_1(X), p_2(X))$. Clearly, there are q possible colours (q is defined in the beginning of this proof). By Theorem 15.4.4, there exist an r-subset S of Z and a colour (u, w) such that every l-subset X of S is of colour (u, w).

We claim that both u and w are non-empty. Indeed, suppose that $u = \emptyset$ and choose an r-set U' such that $U \subseteq U' \subseteq T - S$. By Lemma 15.4.5 there is an r-path subdigraph $\mathcal{L}'$ in D from U' to S containing no vertex in $T - (U' \cup S)$. The path subdigraph $\mathcal{L}'$ includes a path subdigraph from U to some $X \subseteq S$ having no vertex in $T - (U \cup X)$. Thus, $u = p_1(X) \neq \emptyset$. Analogously, one proves that $w \neq \emptyset$.

Let $u := (u_1, \ldots, u_l)$ and $w := (w_1, \ldots, w_l)$ and let $\mathcal{L}_1(X), \mathcal{L}_2(X)$ be the corresponding l-linkages. We have already established that for every l-subset X of S, $\mathcal{L}_1(X)$ is an l-linkage from u to $\bar{X}$ and $\mathcal{L}_2(X)$ is an l-linkage from $\bar{X}$ to w.

For $i = 1, 2, \ldots, l$ define j_i as follows: $w_{j_i} = u_i$. By the definition of l and Theorem 13.5.3 of Erdős and Szekeres, there are $1 \leq i_1 < i_2 < \ldots < i_k \leq l$ so that the sequence $j_{i_1}, j_{i_2}, \ldots, j_{i_k}$ either increases or decreases. Define

$(i'_1, \ldots, i'_k)$ to be $(j_{i_1}, \ldots, j_{i_k})$ in the first case and $(j_{i_k}, \ldots, j_{i_1})$ in the second. Hence, $i'_1 < \ldots < i'_k$.

Let $G := \{\bar{S}(l), \bar{S}(2l), \ldots, \bar{S}(kl)\}$. Choose an l-subset X of S so that $\bar{S}(hl) = \bar{X}(i_h)$ for $h \in [k]$. Since $\mathcal{L}_1(X)$ is an l-linkage from $(u_1, \ldots, u_l)$ to $\bar{X}$, it includes a path subdigraph $\mathcal{L}_1$ linking $(u_{i_1}, \ldots, u_{i_k})$ to $\bar{G}$. Moreover, the only vertices of T in $\mathcal{L}_1$ belong to $U \cup G$.

Analogously choose an l-subset Y of S so that $\bar{S}(hl) = \bar{Y}(i'_h)$ for $h \in [k]$. Since $\mathcal{L}_2(Y)$ is an l-linkage from $\bar{Y}$ to $(w_1, \ldots, w_l)$, it includes a path subdigraph $\mathcal{L}_2$ linking $\bar{G}$ to $(w_{i'_1}, \ldots, w_{i'_k})$. Observe that $(w_{i'_1}, \ldots, w_{i'_k})$ is either $(u_{i_1}, \ldots, u_{i_k})$ or $(u_{i_k}, \ldots, u_{i_1})$. Moreover, every (directed) cycle in $\mathcal{L}_1 \cup \mathcal{L}_2$ meets T (since T is a feedback vertex set), and the only vertices of T in $V(\mathcal{L}_1 \cup \mathcal{L}_2)$ are $u_{i_1}, \ldots, u_{i_k}$ and the elements of G; and so $\mathcal{L}_1, \mathcal{L}_2$ satisfy the lemma. □

A digraph D is **bivalent** if, for every $v \in V(D)$, $d^+(v) = d^-(v) \in \{1, 2\}$. The following lemma is the most technically involved basic result in [769].

Lemma 15.4.7 *For every integer $c \geq 1$ there exists $k \geq 0$ such that, for every bivalent digraph D, if there exists a pair of k-path subdigraphs $\mathcal{L}_1, \mathcal{L}_2$ in D so that each path of $\mathcal{L}_1$ meets each path of $\mathcal{L}_2$ and $\mathcal{L}_1 \cup \mathcal{L}_2$ has no (directed) cycles, then $\nu_0(D) \geq c$.* □

Using this lemma and Theorem 15.4.4, one can prove the following:

Lemma 15.4.8 *For every integer $c \geq 1$ there exists $k \geq 0$ so that the following holds. Let D be a digraph and let $u_1, \ldots, u_k, w_1, \ldots, w_k$ be distinct vertices of D. Let $\mathcal{L}_1, \mathcal{L}_2$ be path subdigraphs in D linking $(u_1, \ldots, u_k)$ to $(w_1, \ldots, w_k)$ and $(w_1, \ldots, w_k)$ to one of $(u_1, \ldots, u_k)$, $(u_k, \ldots, u_1)$, respectively. If every (directed) cycle of $\mathcal{L}_1 \cup \mathcal{L}_2$ meets $\{u_1, \ldots, u_k, w_1, \ldots, w_k\}$, then $\nu_0(D) \geq c$.* □

Theorem 15.4.9 (Reed, Robertson, Seymour and Thomas) *[769] For every integer $c \geq 1$ there exists a (minimum) integer $t_0(c)$ such that, for every digraph D with $\nu_0(D) < c$, we have $\tau_0(D) \leq t_0(c)$.*

Proof: We use induction on $c \geq 1$. For $c = 1$, this theorem is trivially true. Assume that $c \geq 2$ and $t_0(c-1)$ exists. Let k be as in Lemma 15.4.8, and let t be as in Lemma 15.4.6. We prove that there is no digraph D with $\nu_0(D) < c$ and $\tau_0(D) > t - 1$ (i.e., $t_0(c) \leq t - 1$). Suppose that D is such a digraph. By Lemma 15.4.6, there exist $u_1, \ldots, u_k, w_1, \ldots, w_k$ and $\mathcal{L}_1, \mathcal{L}_2$ as in Lemma 15.4.6. This means, by Lemma 15.4.8, that $\nu_0(D) \geq c$, a contradiction. □

15.5 Optimal Orderings and Seymour's Second Neighbourhood Conjecture

Recall that for a vertex x in a digraph D, $N^{+2}(x)$ is the set of vertices of distance two from x. Seymour posed the following conjecture (see [255] and

Problem 325, page 804 in volume 197/198 (1999) of *Discrete Mathematics*) called **Seymour's Second Neighbourhood Conjecture**.

Conjecture 15.5.1 *Every oriented graph $D = (V, A)$ has a vertex x such that*

$$|N^+(x)| \leq |N^{+2}(x)|. \tag{15.2}$$

Note that if we allow 2-cycles, then the conjecture is no longer true as can be seen by taking the complete digraph $\overleftrightarrow{K}_n$. Note also that if the oriented graph has a vertex of out-degree zero, then this vertex satisfies the conjecture. This observation implies that it is sufficient to consider the conjecture for strong oriented graphs. Kaneko and Locke [581] proved the conjecture for digraphs with minimum out-degree at most 6. Godbole, Cohn and Wright [411] showed that the conjecture holds for almost all digraphs. Chen, Shen and Yuster [202] proved that for every digraph, there is a vertex x such that $|N^+(x)| \leq r|N^{+2}(x)|$, where r is the solution of $2x^3 + x^2 = 1$ ($r > 0.657$).

Conjecture 15.5.1 in the case of tournaments was also stated by Dean and Latka [255]. This special case of the conjecture was proved by Fisher [317] using an analytic approach. Fisher's argument is non-trivial and involves the use of a probability distribution on the vertices along with Farkas' Lemma and several other tools. Moreover, Fisher's method does not explicitly identify a vertex which satisfies (15.2).

Below we give an elementary proof, due to Havet and Thomassé [508], of Conjecture 15.5.1 for the case of tournaments. Havet and Thomassé [508] used the following relaxation of the concept of an optimal ordering, which they called a **local median order**, but we will call a locally optimal ordering. An ordering $\mathcal{L} = v_1, v_2, \ldots, v_n$ of the vertices of a tournament $T = (V, A)$ is **locally optimal** if the following holds for all $1 \leq i \leq j \leq n$. (Here and below we use the notation $[v_i, v_j] = \{v_i, v_{i+1}, \ldots, v_j\}$ for all $1 \leq i \leq j \leq n$.)

$$|N^+(v_i) \cap [v_i, v_j]| \geq |N^-(v_i) \cap [v_i, v_j]| \text{ and} \tag{15.3}$$

$$|N^-(v_j) \cap [v_i, v_j]| \geq |N^+(v_j) \cap [v_i, v_j]|. \tag{15.4}$$

Note that if (15.3) does not hold, then the number of forward arcs in

$$\mathcal{L}' = v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{j-1}, v_i, v_j, \ldots, v_n$$

is larger than in $\mathcal{L}$. Similarly if (15.4) does not hold, then we can obtain a better ordering (with respect to the number of forward arcs) by moving v_j just after v_i. Thus a locally optimal ordering is precisely a local optimum, which cannot be improved by moving just one vertex in the ordering. Such an ordering can be found in polynomial time for any given digraph by using the 1-OPT procedure in Section 18.7.

The following is a direct consequence of the definition of a locally optimal ordering:

Lemma 15.5.2 *Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a locally optimal ordering of the vertices of a tournament T. Then for every $1 \leq i \leq j \leq n$ the ordering $\mathcal{L}_{ij} = v_i, v_{i+1}, \ldots, v_j$ is a locally optimal ordering of $T\langle [v_i, , v_j] \rangle$.* □

Lemma 15.5.2 provides us with a powerful inductive tool as we will see below. Let T be a tournament and let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a locally optimal ordering of $V(T)$. We define a partition $G_{\mathcal{L}}, B_{\mathcal{L}}$ of $N^-(v_n)$ as follows:

$$G_{\mathcal{L}} = \{v_j : v_j \rightarrow v_n \text{ and there exists } i < j \text{ such that } v_n \rightarrow v_i \rightarrow v_j\};$$

$$B_{\mathcal{L}} = N^-(v_n) - G_{\mathcal{L}}.$$

The vertices of $G_{\mathcal{L}}$ are called **good** and those in $B_{\mathcal{L}}$ **bad** vertices. Note that $|N^{+2}(v_n)| \geq |G_{\mathcal{L}}|$. The following result by Havet and Thomassé implies that Conjecture 15.5.1 holds for tournaments.

Theorem 15.5.3 *[508] Let T be a tournament and let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a locally optimal ordering of T. Then the vertex v_n has $|N^{+2}(v_n)| \geq |N^+(v_n)|$.*

Proof: Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a locally optimal ordering of T. We prove by induction on n that

$$|N^+(v_n)| \leq |G_{\mathcal{L}}|. \tag{15.5}$$

If $n = 1$, the claim is trivially true so suppose that $n > 1$. If $B_{\mathcal{L}} = \emptyset$, then we have $|G_{\mathcal{L}}| = |N^-(v_n)| \geq |N^+(v_n)|$, where the equality holds by the definition of the good vertices and the inequality holds by the definition of a locally optimal ordering. Hence we may assume that there is a bad vertex. Choose i as small as possible so that v_i is bad. Define the sets $G^l_{\mathcal{L}}, G^h_{\mathcal{L}}, N^l, N^h$ as follows:

$G^l_{\mathcal{L}} = G_{\mathcal{L}} \cap [v_1, v_i]$ and $G^h_{\mathcal{L}} = G_{\mathcal{L}} \cap [v_{i+1}, v_n]$,
$N^l = N^+(v_n) \cap [v_1, v_i]$ and $N^h = N^+(v_n) \cap [v_{i+1}, v_n]$.

Note that if a vertex is good with respect to the pair $(T\langle \{v_{i+1}, \ldots, v_n\} \rangle, \mathcal{L}^h)$, where $\mathcal{L}^h = v_{i+1}, \ldots, v_n$, then it is also good with respect to $(T, \mathcal{L})$. Hence, by the induction hypothesis (applied to $T\langle \{v_{i+1}, \ldots, v_n\} \rangle$ and the ordering $\mathcal{L}^h$), we have $|N^h| \leq |G^h_{\mathcal{L}}|$. The minimality of i implies that every vertex in $\{v_1, \ldots, v_{i-1}\}$ is either in $G^l_{\mathcal{L}}$ or N^l. Furthermore, since v_i is bad we have $N^l \subseteq N^+(v_i) \cap [v_1, v_{i-1}]$ and $N^-(v_i) \cap [v_1, v_{i-1}] \subseteq G^l_{\mathcal{L}}$. Now using (15.4) we obtain

$$|G^l_{\mathcal{L}}| \geq |N^-(v_i) \cap [v_1, v_{i-1}]| \geq |N^+(v_i) \cap [v_1, v_{i-1}]| \geq |N^l|.$$

Thus we have

$$|G_{\mathcal{L}}| = |G^l_{\mathcal{L}}| + |G^h_{\mathcal{L}}| \geq |N^l| + |N^h| = |N^+(v_n)|,$$

implying that (15.5) holds for all positive integers n. □

If a tournament has a vertex of out-degree zero, then this vertex satisfies (15.2) and the transitive tournament on n vertices shows that this vertex may be the only vertex satisfying (15.2). Using locally optimal orderings Havet and Thomassé [508] proved that unless there is a vertex of out-degree zero, a tournament has at least two vertices satisfying (15.2).

Havet and Thomassé showed by an example that their method (just as Fisher's method [317]) will not suffice to prove Conjecture 15.5.1 in full.

Sullivan stated the following three conjectures similar to Conjecture 15.5.1, where he used $|N^-(v)|$ instead of or together with $|N^+(v)|$.

Conjecture 15.5.4 *[837] Every oriented graph D has a vertex v such that $|N^{+2}(v)| \geq |N^-(v)|$.*

The inequality of Conjecture 15.5.1 can be written as $|N^{+2}(v)|+|N^+(v)| \geq 2|N^+(v)|$. This inspires the following:

Conjecture 15.5.5 *[837] Every oriented graph D has a vertex v such that $|N^{+2}(v)| + |N^+(v)| \geq 2|N^-(v)|$.*

The next conjecture is a weakening of the previous two.

Conjecture 15.5.6 *[837] Every oriented graph D has a vertex v such that $|N^{+2}(v)| + |N^+(v)| \geq 2 \cdot \min\{|N^+(v)|, |N^-(v)|\}$.*

15.6 Ádám's Conjecture

Ádám's conjecture [3, 4] seems one of the most challenging conjectures on cycles in digraphs.

Conjecture 15.6.1 (Ádám) *Every digraph has an arc whose reversal decreases the total number of cycles.*

Originally, Ádám formulated the conjecture for directed multigraphs. This extension was disproved independently by Grinberg, Jirásek and Thomassen (see [424, 566, 864]). Thomassen [864] used the following result of Penn and Witte [745], which is of independent interest and was established with the aid of knot theory on the torus. Note that this theorem generalizes Theorem 6.9.7.

Theorem 15.6.2 *The Cartesian product $\vec{C}_p \times \vec{C}_q$ has a cycle of length k if and only if there is a pair a, b of relatively prime natural numbers such that $ap + bq = k$.* □

The main idea of Thomassen is to apply the following corollary:

Corollary 15.6.3 *[864] Infinitely many digraphs of the type $\vec{C}_p \times \vec{C}_q$ have the property that the reversal of any arc increases the length of a longest cycle.*

Proof: By the above theorem, $\vec{C}_5 \times \vec{C}_{7+10k}$, $k \geq 0$, has no cycle of length $35+50k$ or $34+50k$ (Exercise 15.15). However, the reversal of any arc creates a $(34+50k)$-cycle. This is depicted in Figure 15.3 (due to Thomassen [864]) for $k = 0$ and a similar structure can be used to obtain a cycle of length $34+50k$ when $k \geq 1$. (Actually, Figure 15.3 shows a 35-cycle, too, and this cycle can be generalized for every $k \geq 0$.) □

Theorem 15.6.4 *[864] There is an infinite family of counterexamples to Ádám's conjecture in the case of directed multigraphs.*

Proof: Let $D(k, f)$ be the directed multigraph obtained from $\vec{C}_5 \times \vec{C}_{7+10k}$ by replacing each arc by f parallel arcs. Let t denote the maximum number of cycles through an arc of $\vec{C}_5 \times \vec{C}_{7+10k}$ and let s be the length of a longest cycle in $\vec{C}_5 \times \vec{C}_{7+10k}$. Then no arc of $D(k, f)$ is contained in more than tf^{s-1} cycles, but if we reverse an arc a of $\vec{C}_5 \times \vec{C}_{7+10k}$, then a is is contained in a cycle of length at least $s+1$ and hence a is contained in at least f^s cycles. Hence, if $f > t$, $D(k, f)$ is a counterexample to Ádám's conjecture. □

Grinberg's counterexamples were inspired by projective geometry. Studying circulant digraphs, Jirásek proved the following result.

Theorem 15.6.5 *[568] For any integer $t \geq 1$ and all sufficiently large p, the directed multigraph $C^{[p]}_{8t+4}(2t+1, 2, 4t+4)$ obtained from the circulant digraph $C_{8t+4}(2t+1, 2, 4t+4)$ by replacing each arc with p parallel ones is a counterexample to Ádám's conjecture.* □

Jirásek [568] also showed that $C^{[p]}_{12}(3, 2, 8)$ is a counterexample to Ádám's conjecture for each $p \geq 4$. All the examples by Grinberg, Jirásek and Thomassen have parallel arcs. At the same time, Ádám's conjecture holds for some families of digraphs. Actually, it holds when a digraph has a 2-cycle.

Proposition 15.6.6 *[567] If a digraph D contains a 2-cycle, then D has an arc whose reversal decreases the total number of cycles in D.*

Proof: Let uvu be a 2-cycle in D and, for every $a \in A(D)$, let c_a be the number of cycles in D containing a. Without loss of generality, we may assume that $c_{uv} \leq c_{vu}$. Then, the reversal of vu decreases the number of cycles in D by $c_{vu} - c_{uv} + 1 > 0$. □

Apart from this proposition, Jirásek proved several other assertions on families of digraphs that satisfy Ádám's conjecture. The most interesting is the following:

Theorem 15.6.7 *[567] If, after reversal of at most three arcs, a non-acyclic digraph D becomes acyclic, then D has an arc whose reversal decreases the total number of cycles in D.* □

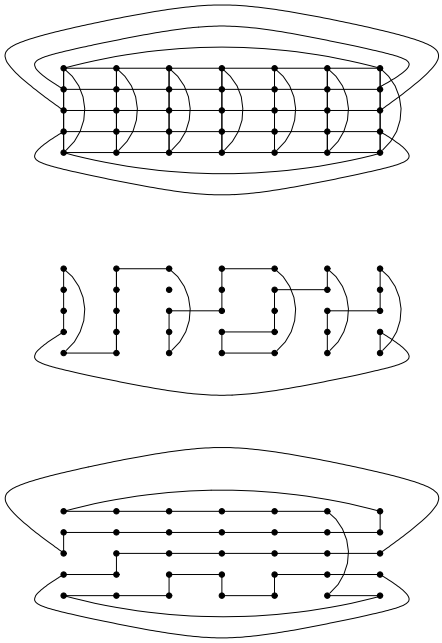


Figure 15.3 $\vec{C}_5 \times \vec{C}_7$ and (directed) cycles of lengths 34 and 35 when an arc is reversed. (All arcs represented by vertical or horizontal straight line segments are directed upwards or to the right.)[864]

To the best of our knowledge, Ádám's conjecture is still open for oriented graphs.

Problem 15.6.8 *[864] Verify Ádám's conjecture for oriented graphs and, in particular, for tournaments.*

15.7 Exercises

15.1. Prove Proposition 15.0.1.

15.2. Prove Lemma 15.3.6.

15.3. **The directed dual of a plane directed multigraph is planar.** Show that if D is a plane directed multigraph, then its directed dual D^* is also planar.

15.4. **Taking duals repeatedly.** Let D be a plane directed multigraph and let D^* be the directed dual of D. Show that the directed dual of D^* is isomorphic to the converse of D.

15.5. Let D be a plane directed multigraph and let D^* be the directed dual of D. Show that if $(S, \bar{S})$ is a directed cut in D^*, then the corresponding arcs in D form a directed cycle.

15.6. Let $D = (V, A)$ be the plane digraph in Figure 15.1(a). Find two arcs in A whose deletion leaves an acyclic directed multigraph. Then check that contracting the corresponding two arcs in D^*, the directed dual of D, results in a strongly connected digraph.

15.7. Show that the problem of finding a maximum size acyclic subdigraph of a directed multigraph $D = (V, A)$ is equivalent to that of finding an ordering $v_1, v_2, \ldots, v_n$ of V such that the number of arcs $v_i v_j$ with $i < j$ is maximum.

15.8. Prove Proposition 15.3.11.

15.9. Let D be an arbitrary directed multigraph. Prove that every minimum feedback arc set of D induces an acyclic subdigraph of D.

15.10. Show that the tournament T in Figure 15.4 has a minimum feedback arc set which induces a transitive subtournament of T.

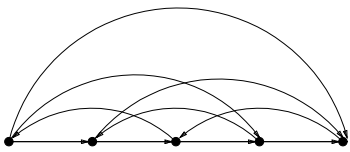


Figure 15.4 A tournament T on five vertices.

15.11. Show that if there exists a polynomial approximation algorithm with approximation guarantee $\rho(n)$ for the feedback arc set problem, then there also exists a polynomial approximation algorithm with approximation guarantee $\rho(n)$ for the feedback vertex set problem and vice versa.

15.12. Construct an infinite family of digraphs D such that $\nu_0(D) < \tau_0(D)$.

15.13. Prove that if the functions $t_0(k)$ and $t_1(k)$ introduced in Subsection 15.4.1 exist, then they are equal.

15.14. (+) Prove Lemma 15.4.5. Hint: use Menger's theorem.

15.15. Prove that $\vec{C}_5 \times \vec{C}_{7+10k}$, $k \geq 0$, has no cycle of length $35 + 50k$ or $34 + 50k$. Hint: apply Theorem 15.6.2.

16. Generalizations of Digraphs: Edge-Coloured Multigraphs

In this chapter, we study the class of edge-coloured multigraphs which is one of the most investigated generalizations of directed multigraphs. The fact that edge-coloured multigraphs generalize directed multigraphs follows, in particular, from Häggkvist's transformation described in Section 16.1. Apart from edge-coloured multigraphs, there are many other generalizations of directed multigraphs such as arc-coloured digraphs, hypertournaments and star hypergraphs.

The main results on arc-coloured digraphs and hypertournaments are provided in Chapter 11 of [91]. As suggested by the name, **arc-coloured digraphs** are digraphs in which all arcs are coloured; notions similar to the ones studied in this paper are investigated in Chapter 11 of [91]. A ***k*-hypertournament** on n vertices is a pair (V, A), where $V = [n]$ and each element of A is a permutation of a distinct k-subset of V (i.e., $|A| = \binom{n}{k}$). A **star hypergraph** is a triple (V, A, r), where (V, A) is a hypergraph and $r : A \to V$ is a function that identifies the initial vertex of each edge $e \in A$. Basic results on star hypergraphs are proved in [116].

We concentrate on edge-coloured multigraphs for a number of reasons: this generalization has been widely studied in graph theory, many concepts and results can be extended from digraphs to edge-coloured multigraphs (see Theorem 16.2.1 and many other results of this chapter) and there are several interesting applications of edge-coloured multigraphs (see, e.g., Section 17.6).

In Section 16.2 we study properly coloured trails (i.e., trails whose consecutive edges differ in colour) in edge-coloured undirected multigraphs. We prove Kotzig's characterization of edge-coloured multigraphs containing properly coloured (PC) Euler trails and Pevzner's theorem that shows how to generate all PC Euler trails of an edge-coloured multigraph from some initial one. Yeo's theorem on PC cycles in edge-coloured graphs, which in a sense characterizes edge-coloured graphs not having PC cycles, is proved in Section 16.3. Section 16.4 is devoted to the problems of finding a PC cycle and path with fixed end-vertices. It is shown that the shortest PC cycle (PC path with fixed end-vertices) can be found in polynomial time (if one exists). Section 16.6 is devoted to generalizations of strong connectivity to edge-coloured multigraphs.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_16,

We consider various interesting results on hamiltonian and longest PC paths and cycles in 2-edge-coloured bipartite multigraphs in Section 16.7. Many of these results can be easily obtained from the corresponding results on digraphs using some transformations also described in this subsection. The characterization of 2-edge-coloured complete graphs containing hamiltonian PC cycles, due to Bankfalvi and Bankfalvi, is given in Section 16.8. Among other things, we prove Saad's theorem characterizing longest PC cycles in 2-edge-coloured complete graphs. PC paths and cycles in c-edge-coloured complete graphs, $c \geq 3$, are studied in Section 16.9. In particular, we give a characterization of c-edge-coloured complete graphs containing PC Hamilton paths obtained by Feng, Giesen, Guo, Gutin, Jensen and Rafiey.

The chapter contains several conjectures and open problems.

16.1 Terminology, Notation and Initial Observations

In this section we consider **edge-coloured multigraphs**, i.e., undirected multigraphs such that each edge has a colour and no two parallel (i.e., joining the same pair of vertices) edges have the same colour. If the number of colours is restricted by an integer c, we speak about ***c*-edge-coloured multigraphs.** We usually use the integers $1, 2, \ldots, c$ to denote the colours in c-edge-coloured multigraphs. In case $c = 2$, we also use the names **red** and **blue** for colours 1 and 2, respectively. The **red subgraph** (**blue subgraph**, respectively) of a 2-edge-coloured multigraph G consists of the vertices of G and all red (blue, respectively) edges of G.

Let G be a c-edge-coloured multigraph ($c \geq 2$). A trail T in G is **properly coloured (PC)** if no two consecutive edges of T have the same colour. A **PC *m*-path-cycle subgraph** $\mathcal{F}_m$ of G is a union of m PC paths and a number of PC cycles in G, all vertex-disjoint. When $m = 0$, we will call $\mathcal{F}_0$ a **PC cycle subgraph** . If G is 2-edge-coloured, then we call a properly coloured trail in G **alternating**. To see that the alternating path and cycle structure of 2-edge-coloured multigraphs generalizes the path and cycle structure of directed multigraphs, we consider the following simple transformation attributed to Häggkvist in [683]; see Figure 16.1. Let D be a directed multigraph. Replace each arc xy of D by two (unoriented) edges xz_{xy} and $z_{xy}y$ by adding a new vertex z_{xy} and then colour the edge xz_{xy} red and the edge $z_{xy}y$ blue. Let G be the 2-edge-coloured graph obtained in this way. It is easy to see that each alternating cycle (path) in G corresponds to a directed cycle (path) in D and vice versa. Hence, in particular, we obtain the following proposition.

Proposition 16.1.1 *The following problems on paths and cycles in 2-edge-coloured graphs are $\mathcal{NP}$-complete:*

(a) The alternating Hamilton cycle problem.
(b) The problem to find an alternating cycle through a prescribed pair of vertices.

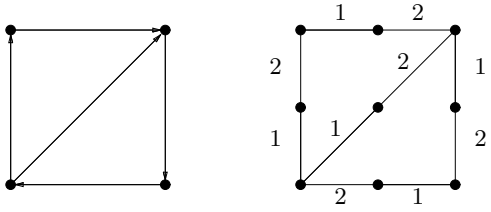


Figure 16.1 Häggkvist's transformation.

Proof: Exercise 16.1. □

Thus, we have established that the alternating path and cycle structure in 2-edge-coloured multigraphs generalizes the (directed) path and cycle structure of directed multigraphs. In fact, we will see, in this section, that the former is certainly more complicated than the latter. Still, several methods and results obtained for directed multigraphs can be adapted to edge-coloured multigraphs.

Petersen's famous paper [747] seems to be the first place where one can find applications of PC trails (cf. [707]). Besides a number of applications in graph theory and algorithms (cf. the papers [908, p. 58] by Woodall and [486] by Häggkvist), the concept of PC trails and its special cases, PC paths and cycles, appears in various other fields including genetics (cf. the papers [270, 271] by Dorninger, [272] by Dorninger and Timischl and [750] by Pevzner) and social sciences (cf. the paper [214] by Chow et al.).

Let G be a c-edge-coloured multigraph and let $v \in V(G)$. The $\boldsymbol{j}$th **neighbourhood** of v is $N_j(v) = \{u \in V(G) : uv \in E(G), \chi(uv) = j\}$ $(1 \leq j \leq c)$. The $\boldsymbol{j}$th **degree** of v, $d_j(v) = |N_j(v)|$ and the **colour set** of v is $\chi(v) = \{i : 1 \leq i \leq c, N_i(v) \neq \emptyset\}$. The **maximum** (**minimum**) **monochromatic degree** of $G = (V, E)$ is defined by

$$\Delta_{mon}(G) = \max\{d_j(v) : v \in V,\ 1 \leq j \leq c\}$$
$$(\delta_{mon}(G) = \min\{d_j(v) : v \in V,\ 1 \leq j \leq c\}).$$

The colour of an edge e in G will be denoted by $\chi(e)$. Let X and Y be two sets of the vertices of G. Then XY denotes the set of all edges having one end vertex in X and the other in Y and $\chi(XY)$ stands for the set of colours of edges in XY. In case all the edges in XY have the same colour, say i, we write $\chi(XY) = i$.

Edge-coloured multigraphs G and H are **colour-isomorphic** if there exists an isomorphism $f : V(G) \to V(H)$ such that $\chi(xy) = \chi(f(x)f(y))$ for every pair x, y of distinct vertices of G. Let $T = p_1p_2 \ldots p_l$ be a trail in G. Then, the trail $p_lp_{l-1} \ldots p_1$, called the **reverse of** $\boldsymbol{T}$, will be denoted by T^{rev}. Also, if $l \geq 2$, define

$$\chi_{end}(T) = \chi(p_{l-1}p_l),\ \chi_{start}(T) = \chi(p_1p_2).$$

Let G be a 2-edge-coloured multigraph of even order n; G is **alternating-pancyclic** if G has an alternating cycle of length $2k$ for every $k = 2, 3, 4, \ldots, n/2$; G is **vertex alternating-pancyclic** if, for every vertex $v \in V(G)$ and every integer $k \in \{2, 3, 4, \ldots, n/2\}$, G contains an alternating cycle through v of length $2k$.

16.2 Properly Coloured Euler Trails

In [624], Kotzig proved the following characterization of edge-coloured multigraphs which contain properly coloured Euler trails.

Theorem 16.2.1 (Kotzig) *[624] An edge-coloured multigraph G has a properly coloured Euler trail if and only if G is connected, each vertex of G is of even degree, and for every vertex x and every colour i, $d_i(x) \leq \sum_{j \neq i} d_j(x)$.*

Proof: Obviously, the conditions above are necessary.

Suppose G satisfies the conditions of Theorem 16.2.1. We will first show that, for every vertex x, the edges of G incident to x can be partitioned into disjoint pairs of distinct edges so that the colours of the edges in each pair are different. This guarantees that each time we enter x through an edge e we can leave it through the edge f forming one of the above pairs with e. (We will denote f by $\text{match}_x(e)$.)

In order to determine this partition, for each vertex x we define an auxiliary graph G_x so that the vertices of G_x are the edges incident to x. Two vertices are connected in G_x if their corresponding edges in G have different colours. It is easy to see that the above partition exists if and only if each G_x has a perfect matching. It remains to prove that each G_x indeed has a perfect matching.

Observe that each G_x is a complete multipartite graph with partite sets of some cardinalities $n_1, n_2, \ldots, n_t$ satisfying the following inequality:

$$n_i \leq \sum_{j \neq i} n_j \tag{16.1}$$

for every $i = 1, 2, \ldots, t$. Choose an edge b between two largest partite sets of G_x. Delete the vertices of b from G_x. Clearly, the partite sets of the obtained graph satisfy the inequality (16.1). This means we can proceed by choosing another edge as above. This process will clearly produce a perfect matching of G_x. (One could easily arrive at the same conclusion using Tutte's theorem on perfect matchings in multigraphs, see e.g. the book [170] by Bondy and Murty.)

Fix a perfect matching $\{(e, \text{match}_x(e)) : e \in V(G_x)\}$ in G_x for every x in G. We call a PC trail Q of G an ***M*-trail** if $\text{match}_x(e) \in E(Q)$ for every $x \in V(Q)$ and every $e \in E(Q)$ incident to x. Clearly, every M-trail is closed. In the obvious way (see the construction of R below), one can

build an M-trail. Let T be an M-trail of G with maximum number of edges. Assume that $E(T) \neq E(G)$. Since G is connected, $G - E(T)$ contains an edge e_1 incident to a vertex x_1 in T. We construct a trail R in $G - E(T)$ as follows: $x_1, e_1, x_2, e_2 = \text{match}_{x_2}(e_1), x_3, e_3 = \text{match}_{x_3}(e_2), x_4, \ldots, x_k, e_k = \text{match}_{x_k}(e_{k-1}), x_{k+1}$, where $e_i = x_i x_{i+1}$ for every $i = 1, 2, \ldots, k$, $x_{k+1} = x_1$ and $e_1 = \text{match}_{x_1}(e_k)$. Observe that T and R are edge-disjoint by the definition of M-trails.

Since x_1 is in T, we can write down T as $\ldots f, x_1, g, \ldots$. Assume, without loss of generality, that $\chi(f) = 1$, $\chi(g) = 2$ and $\chi(e_1) \neq 1$. If $\chi(e_k) \neq 2$, then replace the appearance of x_1 between f and g in T with the trail R obtaining, as a result, an M-trail of G with more edges than T, a contradiction. If $\chi(e_k) = 2$, then replace the appearance of x_1 between f and g in T with the trail R^{rev} obtaining, as a result, an M-trail (observe that $\chi(e_1) > 2$) of G with more edges than T, a contradiction.

Thus, $E(T) = E(G)$, i.e., T is eulerian. □

Benkouar, Manoussakis, Paschos and Saad [137] described an $O(n^2 \log n)$ algorithm for finding a properly coloured eulerian trail in an edge-coloured multigraph G on n vertices that satisfies the conditions of Theorem 16.2.1. Pevzner [750] suggested the following simple and practical algorithm to find a PC eulerian trail in G. Let $P = x_1 x_2 \ldots x_k$ be a PC trail. A colour χ' is **critical with respect to P** if it is the most frequent colour $\chi' \neq \chi(x_{k-1}x_k)$ of edges with one end at x_k and the other in $V(G) - V(P)$. Pevzner's algorithm for an edge-coloured multigraph G satisfying Theorem 16.2.1 proceeds as follows. Let x_1 be an arbitrary vertex in G. Put $P_1 = x_1$ and build up $P_k = x_1 x_2 \ldots x_k$ by adding an arbitrary edge $x_k x_{k+1}$ of colour $\chi(x_1 x_2)$, if this colour is critical with respect to P, or of any critical colour with respect to P, otherwise. We stop when no critical colour edge is available. Pevzner [750] proved that this simple algorithm always produces a PC eulerian trail if one exists (Exercise 16.3).

Using the above transformation by Häggkvist, one can readily obtain the following result (see a direct proof of it in Theorem 1.7.2):

Corollary 16.2.2 *A directed multigraph D is eulerian if and only if D is connected and $d^+(x) = d^-(x)$ for every vertex x in D.* □

Fleischner, Sabidussi and Wegner [322] and Pevzner [750] independently investigated what operations can be used to transform an alternating eulerian trail of a 2-edge-coloured multigraph to any other one. Interestingly enough, while the first paper had a pure theoretical motivation, in the second paper, the author showed some applications of alternating eulerian trails, in general, and those transformations, in particular, to an important $\mathcal{NP}$-hard problem in genetics. We discuss below only the characterization of the transformations in [750].

Let G be a 2-edge-coloured multigraph containing an alternating eulerian trail. In the rest of this subsection, for the sake of convenience, we consider alternating trails as ordered sets of edges. Let $T = T_1T_2T_3T_4T_5$ be an alternating trail (where T_i are fragments of T viewed as subsets of $E(G)$). The transformation $T \rightarrow T^* = T_1T_4T_3T_2T_5$ is called an **order exchange** if T^* is an alternating trail. Let $T = T_1T_2T_3$ be an alternating trail. The transformation $T \rightarrow T^* = T_1T_2^{rev}T_3$ is an **order reflection**, if T^* is an alternating trail. Let X and Y be a pair of alternating trails in G. The number of vertices in the largest common subtrail of X and Y is the **index** $\mathrm{ind}(X, Y)$ of X and Y.

Theorem 16.2.3 (Pevzner) *[750] Every pair of alternating eulerian trails X and Y in a 2-edge-coloured multigraph can be transformed into each other by means of a sequence of order transformations (exchanges and reflections).*

Proof: In the set of alternating eulerian trails $\mathcal{T}$, which can be obtained from X by means of a sequence of order transformations, choose an element, $X^* = x_1x_2 \dots x_q$, having the largest common subtrail with $Y = y_1y_2 \dots y_q$. (Clearly, $x_1 = x_q$ and $y_1 = y_q$.) Let us assume that $\mathrm{ind}(X^*, Y) = \ell < q$. Due to the fact that both X^* and Y are closed, without loss of generality, we may assume that $x_i = y_i$ for $1 \leq i \leq \ell$.

Let $e_1 = x_\ell x_{\ell+1}$ and $e_2 = y_\ell y_{\ell+1}$. Clearly, $\chi(e_1) = \chi(e_2)$. Since X^* is eulerian, X^* contains e_2. There are two possibilities depending on the direction in which we traverse the edge e_2 in X^* (going from x_1 to x_q).

Case 1: In X^* the edge e_2 is traversed from $y_{\ell+1}$ to y_ℓ. In this case,

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots y_{\ell+1}y_\ell \dots x_q.$$

Let $T_1 = x_1 \dots x_\ell$, $T_2 = x_\ell x_{\ell+1} \dots y_{\ell+1}y_\ell$ and $T_3 = y_\ell \dots x_q$. Since $\chi(e_1) = \chi(e_2)$, the transformation $X^* \rightarrow X^{**} = T_1T_2^{rev}T_3$ is an order reflection. But $X^{**} \in \mathcal{T}$ and $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*.

Case 2: In X^* the edge e_2 is traversed from y_ℓ to $y_{\ell+1}$. In this case,

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots (x_p = y_\ell)(x_{p+1} = y_{\ell+1}) \dots x_q.$$

Let $X_1 = x_1 \dots x_\ell$, $X_2 = x_\ell x_{\ell+1} \dots x_p$ and $X_3 = x_p x_{p+1} \dots x_q$.

Claim. *The trail X_3 contains a vertex x_j $(j > p)$ belonging to X_2.*
Proof of Claim: Let $i > \ell$ be the minimum number fulfilling the following condition: vertex y_i of the trail Y is in X_2. The existence of such an i follows from the fact that Y contains the edge $e_1 = y_{t-1}y_t$ for some $t > \ell$. Due to the minimality of i the edge $y_{i-1}y_i$ does not belong to X_2. Condition $i > l$ implies that this edge is not in X_1. Hence, this edge is in X_3 implying that X_2 and X_3 have a common vertex. The claim is proved.

Due to the claim, the trail X^* can now be rewritten as

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots (x_k = x_j) \dots (x_p = x_\ell)(x_{p+1} = y_{\ell+1}) \dots x_j \dots x_q.$$

Let $T_1 = x_1 \dots x_\ell$, $T_2 = x_\ell x_{\ell+1} \dots x_k$, $T_3 = x_k \dots x_p$, $T_4 = x_p \dots x_j$ and $T_5 = x_j \dots x_q$. Consider the edges $f_1 = x_{k-1}x_k$ and $f_2 = x_{j-1}x_j$. If $\chi(f_1) = \chi(f_2)$, then $\chi(f_2) \neq \chi(x_k x_{k+1})$ and $X^{**} = T_1T_4T_3T_2T_5$ is the alternating trail obtained from X^* by means of some order exchange. Clearly, $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*.

If $\chi(f_1) \neq \chi(f_2)$, then $X^{**} = T_1T_4T_2^{rev}T_3^{rev}T_5$ is an alternating trail. This trail is obtained from X^* by means of two order reflections:

$$\begin{aligned} & T_1T_2T_3T_4T_5 \rightarrow T_1T_2(T_3T_4)^{rev}T_5 \\ = {} & T_1T_2T_4^{rev}T_3^{rev}T_5 \rightarrow T_1(T_2T_4^{rev})^{rev}T_3^{rev}T_5 \\ = {} & T_1T_4T_2^{rev}T_3^{rev}T_5. \end{aligned}$$

Clearly, $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*. □

16.3 Properly Coloured Cycles

Using Häggkvist's transformation, we see that the problem to check whether a c-edge-coloured graph has a properly coloured cycle is more general (even for $c = 2$) than the simple problem to verify whether a digraph contains a directed cycle (see Proposition 2.1.1 and the remark afterwards). In this section we consider the following:

Problem 16.3.1 *Given a c-edge-coloured graph G, check whether G contains a properly coloured cycle.*

Grossman and Häggkvist [426] were the first to study this problem. They proved Theorem 16.3.2 below in the case $c = 2$. Yeo [914] showed Theorem 16.3.2 for every $c \geq 2$. Note that one can find a shortest PC cycle in polynomial time, see Section 16.4.

Let G be a c-edge-coloured graph and let x, y be arbitrary distinct vertices of G. We will use the following additional notation:

$$\chi_{end}(x, y) = \{\chi_{end}(P) : P \text{ is a PC } (x, y)\text{-path}\};$$
$$\chi_{start}(x, y) = \{\chi_{start}(P) : P \text{ is a PC } (x, y)\text{-path}\}.$$

Theorem 16.3.2 (Yeo) *[914] Let G be a c-edge-coloured graph, $c \geq 2$, with no PC cycle. Then, G has a vertex $z \in V(G)$ such that no connected component of $G - z$ is joined to z with edges of more than one colour.*

Proof: Let $G = (V, E)$ be an edge-coloured graph with no PC cycle. Let $p_1 \in V$ be arbitrary. Set $S = \{p_1\} \cup \{s \in V - \{p_1\} : |\chi_{end}(p_1, s)| = 1\}$. Now let $P = p_1p_2 \dots p_l$ ($l \geq 1$) be a PC path of maximum length such that $p_l \in S$, and set $T_k = \{t \in V - \{p_l\} : k \in \chi_{start}(p_l, t)\}$ for every colour $k \in [c]$. If $l = 1$, then let C^* be the set of all colours in G, and if $l \geq 2$ then let C^*

be the set of all colours in G except $\chi_{end}(P)$. We will prove this theorem in three steps.

(1) $V(P) \cap T_k = \emptyset$ for all $k \in C^*$.

If $l = 1$, then this statement is trivially true (since $p_\ell \notin T_k$), so assume that $l \geq 2$ and that the statement is false, which implies that there is a PC (p_l, p_i)-path $R = p_l r_1 r_2 \dots r_{m-1} r_m p_i$ $(m \geq 0)$ with $\chi_{start}(R) = k$, $i \in \{1, 2, \dots, l-1\}$ and $V(R) \cap V(P) = \{p_i, p_l\}$. Clearly $\chi(p_i p_{i+1}) = \chi_{end}(R)$, since otherwise we would obtain the PC cycle $p_i p_{i+1} \dots p_l r_1 r_2 \dots r_{m-1} r_m p_i$. This implies that $Q = p_1 p_2 \dots p_i r_m r_{m-1} \dots r_1 p_l$ is a PC (p_1, p_l)-path, with $\chi_{end}(Q) = \chi_{start}(R) = k \neq \chi_{end}(P)$. We have thus shown that $\{\chi_{end}(Q), \chi_{end}(P)\} \subseteq \chi_{end}(p_1, p_l)$, which implies that $|\chi_{end}(p_1, p_l)| \geq 2$. Therefore $p_l \notin S$, contradicting the definition of P.

(2) If $xy \in E$, $x \in T_k$, $y \notin T_k$ for some $k \in C^*$, then $y = p_l$ and $\chi(xy) = k$.

First we claim that there is a PC (p_l, x)-path R with $\chi_{end}(R) \neq \chi(xy)$ and $\chi_{start}(R) = k$.

By the definition of T_k, there is a PC (p_l, x)-path Q with $\chi_{start}(Q) = k$. If $\chi_{end}(Q) \neq \chi(xy)$, we set $R := Q$, so assume that $\chi_{end}(Q) = \chi(xy)$. By (1), PQ is a PC (p_1, x)-path, which is longer than P. This implies that $x \notin S$, so $|\chi_{end}(p_1, x)| \geq 2$. Thus there is a PC (p_1, x)-path L with $\chi_{end}(L) \neq \chi(xy)$. Let $w \in (V(L) \cap V(P \cup Q)) - \{x\}$ be chosen so that $V(L[w, x]) \cap V(P \cup Q) = \{w, x\}$.

Suppose that $w \in V(P) - \{p_l\}$. Then $QL^{rev}[x, w]$ is a PC (p_l, w)-path whose first edge has colour k. This implies that $w \in T_k$, which contradicts (1). Hence $w \in V(Q)$ and $\chi_{start}(Q[w, x]) = \chi_{start}(L[w, x])$, since otherwise $Q[w, x]L^{rev}[x, w]$ is a PC cycle. This implies that $R = Q[p_l, w]L[w, x]$ is a PC (p_l, x)-path with $\chi_{start}(R) = k$ and $\chi_{end}(R) \neq \chi(xy)$. Thus, the claim is proved.

Let R be as guaranteed by the claim. If $y \neq p_l$, then Ry is a PC (p_l, y)-path with $\chi_{start}(Ry) = k$, which contradicts the assumption that $y \notin T_k$. Thus $y = p_l$. If $\chi(xy) \neq k$, then we obtain the PC cycle Ry, which is also a contradiction. Thus $\chi(xy) = k$.

(3) No connected component of $G - p_l$ is joined to p_l with edges of more than one colour.

Assume that the statement is false, and let $p_l x$ and $p_l y$ be a pair of distinct edges in G such that x and y belong to the same connected component of $G - p_l$ and $\chi(p_l x) \neq \chi(p_l y)$. Assume without loss of generality that $\chi(p_l x) \in C^*$ (otherwise interchange x and y). In $G - p_l$ there is a (not necessarily PC) path $R = r_1 r_2 \dots r_m$ $(m \geq 2)$ between $x = r_1$ and $y = r_m$. If $y \in T_{\chi(p_l x)}$, then since $p_l \notin T_{\chi(p_l x)}$, (2) implies that $\chi(p_l y) = \chi(p_l x)$, which is a contradiction. Therefore $y \notin T_{\chi(p_l x)}$, which together with $x \in T_{\chi(p_l x)}$ implies that there exists an i $(1 \leq i \leq m-1)$ such that $r_i \in T_{\chi(p_l x)}$ and $r_{i+1} \notin T_{\chi(p_l x)}$. This, however, contradicts (2), since $r_i r_{i+1} \in E$ but $p_l \neq r_{i+1}$. □

One can see that Theorem 16.3.2 actually solves Problem 16.3.1. Indeed, if G has no vertex z such that all edges from z to any of the components of $G-z$ are of the same colour, then Theorem 16.3.2 implies that G contains a PC cycle. If G has such a vertex z, we may consider only $G-z$ or its components (if $G-z$ is disconnected), since no PC cycle can contain z. (See also Figure 16.2.) This leads to an obvious polynomial recursive algorithm (for a vertex $x \in G$, the components of $G-x$ can be found in $O(|V(G)|+|E(G)|)$ time).

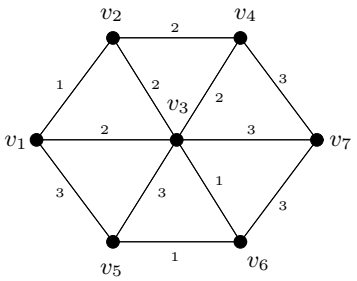


Figure 16.2 An edge-coloured graph with no PC cycle. To see this, it suffices to check that every vertex v_i has only edges of the same colour to $\{v_1, \ldots, v_{i-1}\}$.

Interesting corollaries of Theorem 16.3.2 are given as exercises (Exercises 16.7,16.8) in this chapter. Theorem 16.3.2 also implies:

Corollary 16.3.3 *[132, 623, 666] There does not exist a bridgeless graph that contains a unique perfect matching.*

Proof: Exercise 16.6. □

Let us consider the following function introduced by Gutin [460]: $d(n,c)$, the minimum number k such that every c-edge-coloured graph of order n and minimum monochromatic degree at least k has a properly coloured cycle. It was proved in [460] that $d(n,c)$ exists and that

$$d(n,c) \leq \frac{1}{\lfloor c/2 \rfloor}(\log_2 n - \frac{1}{3}\log_2 \log_2 n + \Theta(1)). \tag{16.2}$$

Abouelaoualim et al. [2] stated a conjecture which implies that $d(n,c) = 1$ for each $c \geq 2$. Using a recursive construction inspired by Theorem 16.3.2 of c-edge-coloured graphs with minimum monochromatic degree p and without PC cycles, we show that

$$d(n,c) \geq \frac{1}{c}(\log_c n - \log_c \log_c n) \tag{16.3}$$

and, thus, the conjecture does not hold.

Theorem 16.3.4 *[460] For each $d \geq 1$ there is an edge-coloured graph G with $\delta_{mon}(G) = d$ and with no PC cycle.*

Proof: Let $(p_1, p_2, \ldots, p_c)$ be a vector with non-negative integral coordinates p_i. For an arbitrary $(p_1, p_2, \ldots, p_c)$, $G(p_1, p_2, \ldots, p_c)$ is recursively defined as follows: take a new vertex x and graphs $H_1 = G(p_1 - 1, p_2, p_3, \ldots, p_{c-1}, p_c)$ if $p_1 > 0$, $H_2 = G(p_1, p_2 - 1, p_3, \ldots, p_{c-1}, p_c)$ if $p_2 > 0$, ..., $H_c = G(p_1, p_2, p_3, \ldots, p_{c-1}, p_c - 1)$ if $p_c > 0$ and add an edge of colour i between x and every vertex of H_i for each i such that $p_i > 0$. In particular, $G(0, 0, \ldots, 0) = K_1$.

It is easy to see, by induction on $p_1 + p_2 + \cdots + p_c$, that $G = G(p_1, p_2, \ldots, p_c)$ has no PC cycle and $\delta_{mon}(G) = \min\{p_i : \ i = 1, 2, \ldots, c\}$. □

In fact, for each $d \geq 1$ there are infinitely many edge-coloured graphs G with $\delta_{mon}(G) = d$ and with no PC cycle. Indeed, in the construction of $G(p_1, p_2, \ldots, p_c)$ above we may assume that $G(0, 0, \ldots, 0)$ is an edgeless graph of arbitrary order.

Proposition 16.3.5 *[460] Let $n(p_1, p_2, \ldots, p_c)$ be the number of vertices in graph $G(p_1, p_2, \ldots, p_c)$ and let $n_c(p) = n(p_1, \ldots, p_c)$ when $p_i = p$ for each $i \in [c]$. Then $n(p_1, \ldots, p_c) \leq s2^s$, where $s = p_1 + p_2 + \ldots + p_c$, provided $s > 0$ and $p \geq \frac{1}{c}(\log_c n_c(p) - \log_c \log_c n_c(p))$.*

Proof: We first prove $n(p_1, \ldots, p_c) \leq s2^s$ by induction on $s \geq 1$. The inequality clearly holds for $s = 1$. By induction hypothesis, for $s \geq 2$, we have

$$\begin{aligned} n(p_1, \ldots, p_c) &\leq 1 + \sum_{i=1}^{c} \{n(p_1, \ldots, p_{i-1}, p_i - 1, p_{i+1}, \ldots, p_c) : \ p_i > 0\} \\ &\leq 1 + c(s-1)c^{s-1} \leq sc^s. \end{aligned}$$

Thus, $n_c(p) \leq cp \cdot c^{cp}$. Observe that $n_c(p) > ac^a$ provided $a = \log_c n_c(p) - \log_c \log_c n_c(p)$ and, thus, $cp \geq \log_c n_c(p) - \log_c \log_c n_c(p)$. □

The inequality (16.3) follows from Proposition 16.3.5 and the fact that graphs $G(p, p, \ldots, p)$ have no PC cycles. The bounds (16.2) and (16.3) imply that $d(n, c) = \Theta(\log_2 n)$ for every fixed $c \geq 2$.

Conjecture 16.3.6 *[460] There is a function $s(c)$ dependent only on c such that $d(n, c) = s(c) \log_2 n(1 + o(1))$.*

In particular, it would be interesting to determine $s(2)$.

We conclude this section by mentioning a paper [323] by Fleischner and Szeider, where a characterization was obtained of edge-coloured graphs in which every edge belongs to a PC cycle.

16.4 Gadget Graphs and Shortest PC Cycles and (s,t)-Paths

In this section based on the paper [465] by Gutin and Kim, we consider a family of transformations of an edge-coloured multigraph G into an ordinary graph that allow us to check the existence PC cycles and PC (s,t)-paths in G and, if they exist, to find shortest ones among them. We raise a problem of finding the optimal transformation and consider a possible solution to the problem.

In Subsection 16.4.1, we consider gadget constructions introduced in [85, 465, 838]. The P-gadget graphs G^* and G^{**} of an edge-coloured graph G described in Subsection 16.4.2 allow one to transform several problems on properly coloured subgraphs of G into perfect matching problems in G^* or G^{**}.

16.4.1 P-Gadgets

Let G be an edge-coloured multigraph and let $G' = G - \{x \in V(G) : |\chi(x)| = 1\}$. For each $x \in V(G')$ let G_x be an arbitrary (non-edge-coloured) graph with the following four properties:

P1 $\{x_q : q \in \chi(x)\} \subseteq V(G_x)$;
P2 G_x has a perfect matching;
P3 For each $p \neq q \in \chi(x)$, if the graph $G_x - \{x_p, x_q\}$ is not empty, it has a perfect matching;
P4 For each set $L \subseteq \chi(x)$ with at least three elements; if the graph $G_x - \{x_l : l \in L\}$ is not empty, it has no perfect matching.

Each G_x with the properties P1-P4 is called a **P-gadget**. Let us consider the following three P-gadgets; the first two are known in the literature and the third one is new.

1. One P-gadget is due to Szeider [838]:

$$V(G_x) = \{x_i, x'_i : i \in \chi(x)\} \cup \{x''_a, x''_b\} \text{ and}$$

$$E(G_x) = \{x''_a x''_b, x'_i x''_a, x'_i x''_b : i \in \chi(x)\} \cup \{x_i x'_i : i \in \chi(x)\}.$$

   We will call this the **SP-gadget**.
2. Another gadget is due to Bang-Jensen and Gutin [85]:

$$V(G_x) = \{x_j : j \in \chi(x)\} \cup \{y_j : j \in \chi(x) \setminus \{m, M\}\},$$

   where $m = \min \chi(x)$, $M = \max \chi(x)$ and

$$E(G_x) = \{x_j y_k : j \in \chi(x),\ k \in \chi(x) \setminus \{m, M\}\} \cup \{x_j x_k : j \neq k \in \chi(x)\}.$$

   We will call this the **BJGP-gadget**.

3. The following gadget introduced by Gutin and Kim [465] is a sort of crossover of the above two and is called the **XP-gadget**:

$$V(G_x) = \{x_j : \ j \in \chi(x)\} \cup \{y_j : \ j \in \chi(x) \setminus \{m, M\}\},$$

where m and M are defined above, and

$$E(G_x) = \{x_m x_M\} \cup \{x_j y_j, \ x_m y_j, \ x_M y_j : \ j \in \chi(x) \setminus \{m, M\}\}.$$

It is not difficult to verify that the tree P-gadgets indeed satisfy P1-P4 (Exercise 16.9). Let $z = \chi(x)$. Observe that the SP-gadget has $2z+2$ vertices and $3z+1$ edges, the BJGP-gadget $2z-2$ vertices and $z(3z-5)/2$ edges, the XP-gadget $2z-2$ vertices and $3z-5$ edges. Thus, the XP-gadget has the minimum number of vertices and edges among the three P-gadgets. It is not difficult to verify that the XP-gadget has the minimum number of vertices and edges among all possible P-gadgets for $z = 2, 3, 4$ (Exercise 16.10). Perhaps, this is true for every z.

Conjecture 16.4.1 *[465] The XP-gadget has the minimum number of vertices and edges among all possible P-gadgets for every $z \geq 2$.*

We will see in the next subsection why minimizing the numbers of vertices and edges in P-gadgets is important for speeding up some algorithms on edge-coloured multigraphs.

16.4.2 P-Gadget Graphs

Let G be a c-edge-coloured multigraph and let G_x be a P-gadget for $x \in V(G')$. The graph G^* is defined as follows: $V(G^*) = \cup_{x \in V(G')} V(G_x)$ and $E(G^*) = E_1 \cup E_2$, where $E_1 = \cup_{x \in V(G')} E(G_x)$ and $E_2 = \{y_q z_q : \ y, z \in V(G'), \ yz \in E(G), \ \chi(yz) = q, \ 1 \leq q \leq c\}$. This construction is illustrated in Figure 16.3.

Let s, t be a pair of distinct vertices of G and let $H = G - \{s, t\}$. Let G^{**} be constructed from H^* by adding s and t and edges $E_3 = \{sx_i : \ sx \in E(G), \chi(sx) = i\} \cup \{tx_i : \ tx \in E(G), \chi(tx) = i\}$. This construction is illustrated in Figure 16.4.

We will denote the number of vertices and edges in graphs G, G^* and G^{**} by n, m, n^*, m^*, n^{**} and m^{**}, respectively.

The following result relates perfect matchings of G^* with PC cycle subgraphs of G. PC cycle subgraphs are important in several problems on edge-coloured graphs (for example, for the PC Hamilton cycle problem). Recall that $G' = G - \{x \in V(G) : \ |\chi(x)| = 1\}$.

Theorem 16.4.2 *[465] Let G be a connected edge-coloured multigraph such that G' is non-empty. Then G has a PC cycle subgraph with r edges if and only if G^* has a perfect matching with exactly r edges in E_2.*

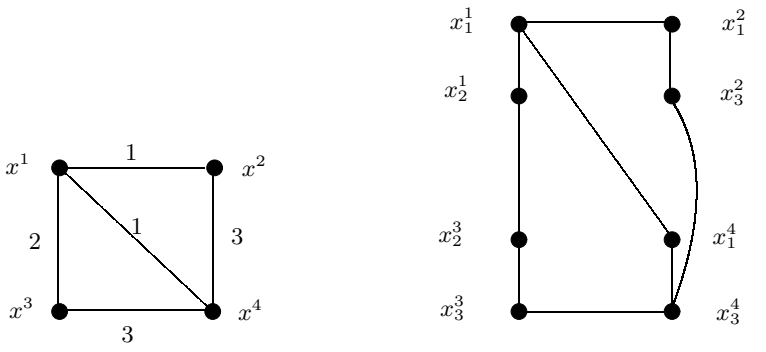


Figure 16.3 The left figure shows a 3-edge-coloured graph G. The right figure depicts G^* with BJGP-gadget (XP-gadget).

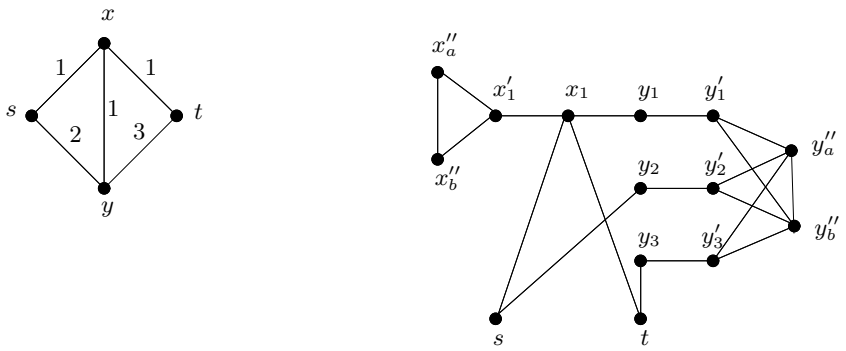


Figure 16.4 A 3-edge-coloured graph G and G^{**} with SP-gadget.

Proof: Let M be a perfect matching of G^* with exactly edges

$$x^1_{p_1}y^1_{q_1}, \ldots, x^r_{p_r}y^r_{q_r}$$

in E_2. For a vertex x of G', let Q_x be the set of edges in E_2 adjacent to G_x. By P2, each G_x has even number of vertices ($x \in V(G')$) and since M is a perfect matching in G^*, there is even number of edges in Q_x. By P4, Q_x has either no edges or two edges for each $x \in V(G')$. Let X be the set of all vertices $x \in V(G')$ such that $|Q_x| = 2$. Then, by the definition of G^*, $G\langle X\rangle$ contains a PC cycle factor. It remains to observe that $|X| = r$.

Now let F be a PC cycle subgraph of G with r edges. Observe that the edges of F correspond to a set Q of r independent edges of G^* and that either no edges or two edges of Q are adjacent to G_x for each $x \in V(G')$. Now delete the vertices adjacent with Q from each G_x and observe that

each remaining non-empty gadget has a perfect matching by P2 and P3. Combining the perfect matchings of the non-empty gadgets with Q, we get a perfect matching of G^* with exactly r edges from E_2. □

The first part of the next assertion generalizes a result from [85]. The second part is based on an approach which leads to a more efficient algorithm than in [1].

Corollary 16.4.3 *[465] One can check whether an edge-coloured multigraph G has a PC cycle and, if it does, find a maximum PC cycle subgraph of G in time $O(n^* \cdot (m^* + n^* \log n^*))$. Moreover one can find a shortest PC cycle in G in time $O(n \cdot n^* \cdot (m^* + n^* \log n^*))$.*

Proof: We may assume that G is connected and that G' is not empty. By Theorem 16.4.2, it is enough to find a perfect matching of G^* containing the maximum number of edges from E_2. Assign weight 0 (1, respectively) to edges of G^* in E_1 (E_2, respectively). Now we need to find a maximum weight perfect matching of G^* which can be done in time $O(n^* \cdot (m^* + n^* \log n^*))$ by a matching algorithm in [373].

To find a shortest PC cycle in G, choose a vertex $x \in V(G')$. We will find a shortest PC cycle in G traversing x. By Theorem 16.4.2, it is enough to find a perfect matching of G^* containing the minimum number of edges from E_2 while containing at least one edge from E_2 so that the corresponding PC cycle in G should be non-trivial. We define the weights on edges of G^* as follows. Assign M, where M is a sufficiently large number, to each edge in E_2 incident with G_x. For all other edges, assign weight 1 (0, respectively) to edges of G^* in E_1 (E_2, respectively). A maximum weight perfect matching of G^* contains exactly two edges of weight M by P4, and contains the minimum number of edges in E_2. Finding a maximum weight perfect matching of G^* can be done in time $O(n^* \cdot (m^* + n^* \log n^*))$ and we iterate the process for each $x \in V(G')$. □

The proof of the following result is analogous to the proof of Theorem 16.4.2 and is left as an exercise (Exercise 16.11).

Theorem 16.4.4 *[465] Let G be an edge-coloured multigraph and let s, t be a pair of distinct vertices of G. If G^{**} is non-empty, then G has a PC 1-path-cycle subgraph with r edges in which the path is between s and t if and only if G^{**} has a perfect matching with exactly r edges not in E_1.* □

The next assertion generalizes a result from [1].

Corollary 16.4.5 *[465] Let G be an edge-coloured multigraph. One can check whether there is a PC (s,t)-path in G in time $O(m^{**})$ and if G has one, a shortest PC (s,t)-path can be found in time $O(n^{**} \cdot (m^{**} + n^{**} \log n^{**}))$.*

Proof: Let L be a graph. Given a matching M in L, we call a path P in L **M-augmenting** if, for any pair of adjacent edges in P, exactly one of

them belongs to M and the first and last edges of P do not belong to M. Consider a perfect matching M of H^*, where $H = G - \{s,t\}$, which is a collection of perfect matchings of G_x for all $x \in V(G')$. The existence of a perfect matching in G_x is guaranteed by P2. Observe that G has a PC (s,t)-path if and only if there is an M-augmenting (s,t)-path P in G^{**}. Since an M-augmenting path P can be found in time $O(m^{**})$ (see [845]), we can find a PC (s,t)-path in G, if one exists, in time $O(m^{**})$.

To find a shortest PC (s,t)-path, we assign each edge in $\bigcup_{x \in V(G')} E(G_x)$ weight 0 and every other edge of G^{**} weight 1. Observe that a minimum weight perfect matching Q in the weighted G^{**} corresponds to a shortest PC (s,t)-path. Finding a minimum weight perfect matching can be done in time $O(n^{**} \cdot (m^{**} + n^{**} \log n^{**}))$. □

We finish this section by the following result from the paper [85] by Bang-Jensen and Gutin that will be used in various parts of this chapter.

Theorem 16.4.6 *[85] One can construct a maximum PC cycle subgraph and a maximum PC 1-path-cycle subgraph, respectively, in a c-edge-coloured multigraph G on n vertices in time $O((cn)^3)$.*

Proof: The assertion for a maximum PC cycle subgraph follows from Corollary 16.4.3 (we can use any of the three P-gadgets). We can easily transform the maximum PC 1-path-cycle subgraph problem to the maximum PC cycle subgraph problem as follows. Add an extra vertex x to G and join x to every vertex of G by two edges of colour $c+1$ and $c+2$, respectively (new colours). Clearly, a maximum PC cycle subgraph of the new multigraph corresponds to a maximum PC 1-path-cycle subgraph of G. □

16.5 Long PC Cycles and Paths

The following interesting result and conjecture were obtained by Abouelaoualim, Das, Fernandez de la Vega, Karpinski, Manoussakis, Martinhon and Saad [2].

Theorem 16.5.1 *[2] Let G be a c-edge-coloured multigraph G with n vertices and with $\delta_{mon}(G) \geq \lceil \frac{n+1}{2} \rceil$. If $c \geq 3$ or $c = 2$ and n is even, then G has a PC Hamilton cycle. If $c = 2$ and n is odd, then G has a PC cycle of length $n-1$.* □

Conjecture 16.5.2 *[2] Theorem 16.5.1 holds if we replace $\delta_{mon}(G) \geq \lceil \frac{n+1}{2} \rceil$ by $\delta_{mon}(G) \geq \lceil \frac{n}{2} \rceil$.*

We cannot replace $\delta_{mon}(G) \geq \lceil \frac{n+1}{2} \rceil$ by $\delta_{mon}(G) \geq \lceil \frac{n-1}{2} \rceil$ due to the following example. Let H_1 and H_2 be c-edge-coloured complete multigraphs (for each pair x, y of vertices and each $i \in [c]$ and $j \in \{1,2\}$, H_j has an edge

between x and y of colour i) of order $p+1$ that have precisely one vertex in common. Clearly, a longest PC cycle in $H_1 \cup H_2$ is of length $p+1$.

Since the longest PC path problem is $\mathcal{NP}$-hard, it makes sense to study lower bounds on the length of a longest PC path. The following result was proved by Abouelaoualim, Das, Fernandez de la Vega, Karpinski, Manoussakis, Martinhon and Saad [2].

Theorem 16.5.3 *Let G be a c-edge-coloured graph on n vertices and with $\delta_{mon}(G) = d \geq 1$. Then G has a PC path of length at least* $\min\{n-1, 2\lfloor \frac{c}{2} \rfloor d\}$. □

The authors of [2] raised the following two conjectures.

Conjecture 16.5.4 *Let G be a c-edge-coloured graph of order n and let $d = \delta_{mon}(G) \geq 1$. Then G has a PC path of length at least* $\min\{n-1, 2cd\}$.

They also conjectured the following analog of Theorem 16.5.3 for multigraphs:

Conjecture 16.5.5 *Let G be a c-edge-coloured multigraph of order n with $\delta_{mon}(G) = d \geq 1$. Then G has a PC path of length at least* $\min\{n-1, 2d\}$.

16.6 Connectivity of Edge-Coloured Multigraphs

Strong connectivity plays a central role in the study of digraphs. Hence, it is natural to try to obtain some extensions of strong connectivity to edge-coloured graphs. Such extensions have been introduced and studied in the literature. In fact, there are two useful extensions of strong connectivity: one of them generalizes the usual definition of strong connectivity that refers to paths between pairs of vertices and the other extends the definition of cyclic connectivity in digraphs (see Exercise 1.17), which is equivalent to strong connectivity (for digraphs). However, for edge-coloured graphs these two generalizations are not equivalent any more.

In this subsection we study the above-mentioned generalizations of strong connectivity. We restrict ourselves to 2-edge-coloured multigraphs since we will later use connectivity results only for 2-edge-coloured graphs. Also this will make our arguments easier to follow. However, the reader should bear in mind that some of the results below could be generalized to c-edge-coloured multigraphs, $c \geq 2$.

The following notion of colour-connectivity was introduced by Saad [791] (he used another name for this notion). Let G be a 2-edge-coloured multigraph. A pair of vertices x, y of G are **colour-connected** if there exist alternating (x,y)-paths P and Q such that $\chi_{start}(P) \neq \chi_{start}(Q)$ and $\chi_{end}(P) \neq \chi_{end}(Q)$. (Notice that P and Q are paths, not trails.) We define a

vertex x to be colour-connected to itself. We say that G is **colour-connected** if every pair of vertices of G is colour-connected.

Clearly, every alternating cycle is a colour-connected graph. This indicates that colour-connectivity may be useful for solving alternating cycle problems. We can use colour-connectivity more effectively if we know that this is an equivalence relation on the vertices of the graph under consideration. This leads us to the following definition: a 2-edge-coloured multigraph G is **convenient** if colour-connectivity is an equivalence relation on the vertices of G. Unfortunately, there are non-convenient multigraphs. Consider the graph H in Figure 16.5. It is easy to check that the vertices x and y are colour-connected to u, but x and y are not colour-connected in H.

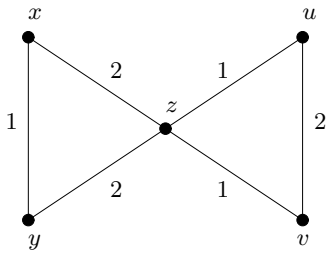


Figure 16.5 A non-convenient 2-edge-coloured graph.

The following result due to Bang-Jensen and Gutin provides another way of checking colour-connectivity. Its proof is left to the reader as Exercise 16.12.

Proposition 16.6.1 *[88] A pair of vertices, x_1, x_2, in a 2-edge-coloured multigraph G are colour-connected if and only if G has four (not necessarily distinct) alternating (x_1, x_2)-paths, P_1, P_2, Q_1, Q_2, such that $\chi_{start}(P_i) = \chi_{end}(Q_i) = i$, $i = 1, 2$.* □

The following proposition by Bang-Jensen and Gutin shows that we can check whether a pair of vertices of a 2-edge-coloured multigraph are colour-connected in polynomial time. To prove Proposition 16.6.2 we can use an easy modification of the special graph $F(G)$ introduced in the previous section: set $E_2 = \bigcup_{x \in W}(\{sx_i : sx \in E(G), \chi(sx) = i\} \cup \{tx_j : tx \in E(G), \chi(tx) = j\})$.

Proposition 16.6.2 *[88] Let $G = (V, E)$ be a connected 2-edge-coloured multigraph and let s and t be distinct vertices of G. For each choice of $i, j \in \{1, 2\}$ we can find an alternating (s, t)-path P with $\chi_{start}(P) = i$ and $\chi_{end}(P) = j$ in time $O(|E|)$ (if one exists).* □

Since colour-connectivity is not an equivalence relation on the vertices of every 2-edge-coloured multigraph, another notion of connectivity, cyclic

connectivity, introduced by Bang-Jensen and Gutin [85], is sometimes more useful. Let $\mathcal{P} = \{H_1, \ldots, H_p\}$ be a set of subgraphs of a multigraph G. The **intersection graph** $\Omega(\mathcal{P})$ of $\mathcal{P}$ has the vertex set $\mathcal{P}$ and the edge set $\{H_iH_j : V(H_i) \cap V(H_j) \neq \emptyset, 1 \leq i < j \leq p\}$. A pair, x, y, of vertices in a 2-edge-coloured multigraph H is **cyclic connected** if H has a collection of alternating cycles $\mathcal{P} = \{C_1, \ldots, C_p\}$ such that x and y belong to some cycles in $\mathcal{P}$ and $\Omega(\mathcal{P})$ is a connected graph.

We formulate the following trivial but useful observation as a proposition.

Proposition 16.6.3 *Cyclic connectivity is an equivalence relation on the vertices of a 2-edge-coloured multigraph.* □

This proposition allows us to consider cyclic connectivity components similar to strong connectivity components of digraphs.

The following theorem due to A. Yeo (private communication, 1998) shows that cyclic connectivity between a pair of vertices can be checked in polynomial time.

Theorem 16.6.4 *For a pair x, y of vertices in a 2-edge-coloured multigraph $H = (V, E)$, one can check whether x and y are cyclic connected in time $O(|E|(|V| + |E|))$.*

Proof: By Proposition 16.6.2, in time $O(|E|)$, one can check whether H has an alternating cycle through a fixed edge $e \in E$. This implies that, in time $O(|V||E|)$, one can verify whether H has an alternating cycle through a fixed vertex $v \in V$.

We now describe a polynomial algorithm to check whether x and y are cyclic connected. Our algorithm starts by initiating $X := \{x\}$. Then, we find an alternating cycle through x; let X' be the vertices except for x of such a cycle. If $y \in X'$, then we are done. Otherwise, delete the vertices of X from H, set $X := X'$ and $X' := \emptyset$. Then, for each edge e with one end-vertex in X and the other not in X find an alternating cycle through the edge (if one exists). Now append all the vertices, except for those in X, in the cycles we have found to X' and check whether $y \in X'$. If $y \notin X'$, then we continue as above. We proceed until either $y \in X'$ or there is no alternating cycle through any edge with one end-vertex in X and the other not in X. Clearly, if $y \in X'$ at some stage, then x and y are cyclic connected, otherwise they are not.

The total time required for the operation of deletion is $O(|V||E|)$. By the complexity bounds above and the fact that we may want to find an alternating cycle through an edge at most once, the complexity of the described algorithm is $O(|E|(|V| + |E|))$. □

The following theorem by Bang-Jensen and Gutin shows that cyclic connectivity implies colour-connectivity.

Theorem 16.6.5 *[88] If a pair, x, y, of vertices in a 2-edge-coloured multigraph G are cyclic connected, then x and y are colour-connected.*

Proof: If x and y belong to a common alternating cycle, then they are colour-connected. So, suppose that this is not the case.

Since x and y are cyclic connected, there is a collection $\mathcal{P} = \{C_1, \ldots, C_p\}$ of alternating cycles in G so that $x \in V(C_1)$, $y \in V(C_p)$, and, for every $i \in [p-1]$ and every $j \in [p]$, $|i-j| > 1$, $V(C_i) \cap V(C_{i+1}) \neq \emptyset$, $V(C_i) \cap V(C_j) = \emptyset$. ($\mathcal{P}$ corresponds to a (C_1, C_p)-path in $\Omega(\mathcal{R})$, where $\mathcal{R}$ is the set of all alternating cycles in G.) We traverse $\mathcal{P}$ as follows. We start at the red (blue, respectively) edge of C_1 incident to x and go along C_1 to the first vertex u that belongs to both C_1 and C_2. After meeting u, we go along C_2 such that the path that we are forming will stay alternating. We repeat the procedure above when we meet the first vertex that belongs to both C_2 and C_3 and so on. Clearly, we will eventually reach y. It follows that there is an (x, y)-path that starts from a red (blue, respectively) edge. By symmetry, we can construct an (x, y)-path that ends at a red (blue, respectively) edge. It follows from Proposition 16.6.1 that x and y are colour-connected. □

16.7 Alternating Cycles in 2-Edge-Coloured Bipartite Multigraphs

The aim of this subsection is to describe two simple approaches which allow one to obtain results for bipartite 2-edge-coloured multigraphs using results on directed graphs.

Let D be a bipartite digraph with partite sets V_1, V_2. Define a 2-edge-coloured bipartite multigraph $CM(D)$ in the following way: $CM(D)$ has the same partite sets as D; every arc (v_1, v_2) from V_1 to V_2 is replaced with red edge v_1v_2 and every arc (v_2, v_1) from V_2 to V_1 is replaced with blue edge v_1v_2. Moreover, $CM^{-1}(G) = H$ if $CM(H) = G$. This simple correspondence which we call the **BB-correspondence** leads us to a number of easy and some more complex results which are described in this and the next subsections. (One example is the fact that the alternating Hamilton cycle problem for bipartite 2-edge-coloured graphs is $\mathcal{NP}$-complete.) In many of our results on cycles we will exploit the following easily verifiable proposition (see Exercise 16.13).

Proposition 16.7.1 *The following three claims are equivalent for a bipartite digraph D:*

(a) D is strongly connected.
(b) $CM(D)$ is colour-connected.
(c) $CM(D)$ is cyclic connected. □

The following correspondence which we call the **BD-correspondence** is less universal but may allow one to exploit the wider area of results on

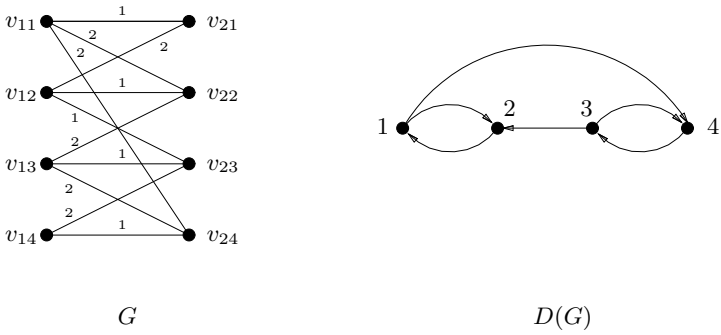


Figure 16.6 An illustration of BD-correspondence.

arbitrary digraphs. The idea of the BD-correspondence can be traced back to Häggkvist [486]. Let G be a 2-edge-coloured bipartite multigraph with partite sets V_1 and V_2 so that $|V_1| = |V_2| = m$ and let G' be the red subgraph of G. Suppose that G' has a perfect matching $v_{11}v_{21}, v_{12}v_{22}, \ldots, v_{1m}v_{2m}$, where $v_{ij} \in V_i$ $(i = 1, 2$ and $1 \leq j \leq m)$. Construct a digraph $D = D(G)$ as follows: $V(D) = [m]$ and, for $1 \leq i \neq j \leq m$, (i, j) is an arc of D if and only if $v_{1i}v_{2j} \in E(G) - E(G')$ (see Figure 16.6). It is easy to see that if D has a Hamilton cycle, then G has a Hamilton alternating cycle including all the edges of the perfect matching. Using the BD-correspondence and Corollary 6.4.3 Hilton [529] proved the following:

Theorem 16.7.2 *Let G be a 2-edge-coloured r-regular bipartite graph such that each of the partite sets of G has m vertices and let G'' be the blue subgraph of G. If $r \geq \frac{m}{2} + 1$ and G'' is s-regular such that $\frac{m}{2} \leq s \leq r - 1$, then G has an alternating Hamilton cycle.*

Proof: Exercise 16.14. □

Although the last theorem is the best possible (consider two disjoint copies of $K_{m/2,m/2}$ with perfect matchings in both copies in red and all other edges in blue), Hilton [529] believes that the bound on r could be lowered considerably if we assume that G is connected. It was noticed by Chetwynd and Hilton [213] that Theorem 16.7.2 follows easily from the following result by Häggkvist [486] (using the BB-correspondence).

Theorem 16.7.3 *Let G be a bipartite graph so that each of the partite sets contains m vertices. If $d(v) + d(w) \geq m + 1$ for every pair v, w of vertices from different partite sets, then every perfect matching of G lies in a Hamilton cycle of G.* □

The BB-correspondence is very useful when we consider 2-edge-coloured complete bipartite multigraphs. In this case we can use the rich theory of

semicomplete bipartite digraphs (discussed in Chapters 6 and 8). By the BB-correspondence, Proposition 16.7.1 and Theorem 6.6.4, we obtain the following:

Theorem 16.7.4 *A 2-edge-coloured complete bipartite multigraph contains an alternating Hamilton cycle if and only if it is colour-connected and has an alternating cycle factor. There is an algorithm for constructing an alternating Hamilton cycle in a colour-connected 2-edge-coloured complete bipartite multigraph on n vertices in time $O(n^{2.5})$ (if one exists).* □

Another condition for a 2-edge-coloured complete multigraph to contain an alternating Hamilton cycle was obtained by Chetwynd and Hilton [213]:

Theorem 16.7.5 *A 2-edge-coloured complete bipartite graph B with partite sets U and W ($|U| = |W| = n$) has an alternating Hamilton cycle if and only if B has an alternating cycle factor and, for every $k = 2, \ldots, n-1$ and every pair of k-sets X and Y such that $X \subset U$, $Y \subset W$, we have*

$$\min\{\textstyle\sum_{x \in X} d_i(x) + \sum_{y \in Y} d_{3-i}(y) : \ i = 1, 2\} > k^2.$$

□

We point out that the original proof of Theorem 16.7.5 is quite similar to that of Theorem 6.6.4. (Another proof of Theorem 16.7.5 is given by Bang-Jensen and Gutin [85]; see also Exercise 16.17.) To see that the set of inequalities of this theorem is necessary, observe that the number of edges between X and Y is precisely k^2. If B has a Hamilton cycle C, then C contains an edge e_1 from $U - X$ to Y as well as an edge e_2 from X to $W - Y$ such that $\chi(e_1) = \chi(e_2)$. Precisely one of these edges contributes to the sum in the corresponding inequality.

Using the corresponding result on longest cycles in semicomplete bipartite digraphs (Theorem 6.6.6), one can obtain the following:

Theorem 16.7.6 *The length of the longest alternating cycle in a colour-connected 2-edge-coloured complete bipartite multigraph G is equal to the number of vertices in maximum alternating cycle subgraph of G. There is an algorithm for finding a longest alternating cycle in a colour-connected 2-edge-coloured complete bipartite multigraph on n vertices in time $O(n^3)$.* □

Let B_r and B'_r be 2-edge-coloured complete bipartite graphs with the same partite sets $\{v_1, \ldots, v_{2r}\}$ and $\{w_1, \ldots, w_{2r}\}$. The edge set of the red (blue) subgraph of B_r (B'_r) consists of

$$\{v_i w_j : \ 1 \leq i, j \leq r\} \cup \{v_i w_j : \ r+1 \leq i, j \leq 2r\}.$$

The following result is a characterization of vertex-alternating-pancyclic 2-edge-coloured complete bipartite multigraphs that can be readily obtained

from the corresponding characterization for semicomplete bipartite digraphs in Theorem 8.6.1.

Theorem 16.7.7 *A 2-edge-coloured complete bipartite multigraph is vertex-alternating-pancyclic if and only if it has an alternating Hamilton cycle and is not colour-isomorphic to one of the graphs* B_r, B'_r $(r = 2, 3, \ldots)$. □

Since none of the graphs B_r, B'_r $(r = 2, 3, \ldots)$ is alternating-pancyclic, we obtain the following:

Corollary 16.7.8 *Let* G *be a 2-edge-coloured complete bipartite multigraph. Then* G *is is alternating-pancyclic if and only if it has an alternating Hamilton cycle and is not colour-isomorphic to one of the graphs* B_r, B'_r $(r = 2, 3, \ldots)$. □

This result was obtained by Das [250]. The equivalent (via the BB-correspondence) claim was proved by Beineke and Little [131] for bipartite tournaments. (Both results were published in the same year!)

To save space we will not give any other 'BB-translations' of results obtained for cycles and paths in semicomplete bipartite digraphs (see Chapters 6 and 8) into the alternating cycles and paths language.

16.8 Paths and Cycles in 2-Edge-Coloured Complete Multigraphs

Since the longest alternating path problem for 2-edge-coloured complete multigraphs is much simpler than the longest alternating cycle problem, we start our study from the former. Bang-Jensen and Gutin characterized 2-edge-coloured complete multigraphs which have an alternating Hamilton path (see Corollary 16.8.2).

Theorem 16.8.1 *[85] Let* G *be a 2-edge-coloured complete multigraph with* n *vertices. Then for any 1-path-cycle subgraph* $\mathcal{F}$ *of* G *there is an alternating path* P *of* G *satisfying* $V(P) = V(\mathcal{F})$ *(if* $\mathcal{F}$ *is a maximum alternating 1-path-cycle subgraph of* G*, then* P *is a longest alternating path in* G*); there exists an* $O(n^3)$ *algorithm for finding a longest alternating path in* G*.*

Proof: Obviously, $\mathcal{F}$ is a 1-path-cycle factor of a complete bipartite subgraph B of G. The factor $\mathcal{F}$ corresponds to a directed path together with a collection of directed cycles, all vertex disjoint, $\mathcal{F}'$ of $CM^{-1}(B)$. Therefore, by Theorem 6.6.1 restricted to semicomplete bipartite digraphs, there is a path P' in $CM^{-1}(B)$ such that $V(P') = V(\mathcal{F}')$. This path corresponds to an alternating path P of B so that $V(P') = V(P)$. Clearly, P is an alternating path in G and, moreover, $V(P) = V(\mathcal{F})$.

The complexity result easily follows from the construction above, and Theorems 6.6.1 and 16.4.6. □

Corollary 16.8.2 *[85] A 2-edge-coloured complete multigraph has an alternating Hamilton path if and only if it contains an alternating 1-path-cycle factor.* □

It is not difficult to prove Corollary 16.8.2 directly (see Exercise 16.20). Clearly, Corollary 16.8.2 implies immediately the first part of Theorem 16.8.1. Thus, the first part of Theorem 16.8.1 and Corollary 16.8.2 are in fact equivalent.

In 1968, solving a problem by Erdős, Bankfalvi and Bankfalvi [123] gave the following characterization of 2-edge-coloured complete graphs which have an alternating Hamilton cycle.

Theorem 16.8.3 (Bankfalvi and Bankfalvi) *[123] A 2-edge-coloured complete graph G of order $2n$ has an alternating Hamilton cycle if and only if it has an alternating cycle factor and, for every $k = 2, \ldots, n-1$ and every pair of disjoint k-subsets X and Y of $V(G)$, $\sum_{x \in X} d_1(x) + \sum_{y \in Y} d_2(y) > k^2$.* □

It is easy to see that the conditions of this theorem are necessary (Exercise 16.16). Saad [791] proved the following more general result, using the notion of colour-connectivity rather than degree conditions. We provide a proof of Theorem 16.8.4 at the end of this subsection after some discussion of implications and generalizations of Theorem 16.8.4.

Theorem 16.8.4 (Saad) *[791] The length of a longest alternating cycle in a colour-connected 2-edge-coloured complete multigraph G is equal to the number of vertices in a maximum alternating cycle subgraph of G.*

Corollary 16.8.5 *[791] A 2-edge-coloured complete multigraph G has an alternating Hamilton cycle if and only if G is colour-connected and contains an alternating cycle factor.* □

Corollary 16.8.5 and the fact that colour-connectivity can be checked in polynomial time (see Propositions 16.6.1 and 16.6.2) shows that the alternating hamiltonian cycle problem for 2-edge-coloured complete multigraphs is polynomial time solvable. However, one cannot deduce the analogous result for the longest alternating cycle problem (for 2-edge-coloured complete multigraphs) from Theorems 16.8.4 and 16.4.6 and Propositions 16.6.1 and 16.6.2, only. The reason is that we do not know how to obtain all maximal colour-connected subgraphs of an arbitrary 2-edge-coloured multigraph in polynomial time. Fortunately, for 2-edge-coloured complete multigraphs G, colour-connectivity is an equivalence relation on the set of vertices (this was first proved by Saad [791] and also follows from Proposition 16.6.3 and the following deeper theorem by Bang-Jensen and Gutin [88]):

Theorem 16.8.6 *[88] A 2-edge-coloured complete multigraph G is colour-connected if and only if G is cyclic connected.*

Proof: Exercise 16.18. □

Thus, we can use Propositions 16.6.1 and 16.6.2 to obtain (vertex-disjoint) colour-connected components of G. Hence, the longest alternating cycle problem for 2-edge-coloured complete multigraphs is also polynomial time solvable. In [88], Bang-Jensen and Gutin showed the following more general result. (Clearly, the case $f(x) = 1$ for every $x \in V(G)$ corresponds to the longest alternating cycle problem.)

Theorem 16.8.7 *[88] The following problem is polynomial time solvable. Given a function f from $V(G)$, the vertex set of a 2-edge-coloured complete multigraph G, to $\mathbb{Z}_0$, find a maximum size alternating closed trail H in G such that $d_{1,H}(x) = d_{2,H}(x) \leq f(x)$ for every $x \in V(G)$.* □

Das [250] and later Häggkvist and Manoussakis [488] observed that the alternating hamiltonian cycle problem for 2-edge-coloured complete bipartite multigraphs can be reduced to the same problem for 2-edge-coloured complete multigraphs using the following simple construction. Consider a 2-edge-coloured complete bipartite multigraph L with bipartition (X, Y). Add to L the edges $\{x'x'', y'y'' : x', x'' \in X, y', y'' \in Y\}$ and set $\chi(XX) = 1$, $\chi(YY) = 2$. Let K be the 2-edge-coloured complete multigraph obtained in this way. It is not difficult to verify that K has no alternating cycle containing any of the edges from $XX \cup YY$. Hence, K contains an alternating hamiltonian cycle if and only if L has one. Moreover, it is easy to check that K is colour-connected if and only if L is colour-connected. In the following, we will call the construction above the **DHM-construction**. The DHM-construction shows that (the non-algorithmic part of) Theorem 16.7.4 follows immediately from Corollary 16.8.5. This illustrates the fact that many problems on alternating cycles for 2-edge-coloured complete multigraphs are more general than those for 2-edge-coloured complete bipartite multigraphs.

Consider the following Hamiltonian 2-edge-coloured complete graphs which are not even-pancyclic (see the proof of this fact below). Let $r \geq 2$ be an integer. Each of the graphs $H(r)$, $H'(r)$, $H''(r)$ has a vertex set $A \cup B \cup C \cup D$ so that the sets A, B, C, D are pairwise disjoint and each of these sets contains r vertices. Moreover, the edge set of the red subgraph of $H(r)$ consists of $AA \cup CC \cup AC \cup AD \cup CB$. The edge set of the red (blue) subgraph of $H'(r)$ ($H''(r)$) consists of $AC \cup CB \cup BD \cup DA$. By the DHM-construction, the following result by Bang-Jensen and Gutin [85] is a generalization of Theorem 16.7.7 (the proof is left as Exercise 16.19).

Theorem 16.8.8 *Let G be a 2-edge-coloured complete multigraph. Then G is vertex-alternating-pancyclic if and only if G has an alternating Hamilton*

cycle and is not colour-isomorphic to the graphs $H(r)$, $H'(r)$, $H''(r)$ *for* $r = 2, 3, \ldots$. □

Since the graphs $H(r)$, $H'(r)$, $H''(r)$ are not alternating-pancyclic for $r = 2, 3, \ldots$, we obtain the following characterization first proved by Das [250].

Corollary 16.8.9 *A 2-edge-coloured complete multigraph* G *is alternating-pancyclic if and only if* G *has an alternating Hamilton cycle and is not colour-isomorphic to the graphs* $H(r)$, $H'(r)$, $H''(r)$ *for* $r = 2, 3, \ldots$. □

The rest of this subsection is devoted to the proof of Theorem 16.8.4 adapted from Bang-Jensen and Gutin [88]. In the statements and the proofs of the rest of this subsection, we use the following notation: G is a 2-edge-coloured complete multigraph with n vertices, $\mathcal{F}_p = C_1 \cup \ldots \cup C_p$ is an alternating cycle subgraph in G consisting of p cycles, $C_1, \ldots, C_p$; for each $i = 1, 2, \ldots, p$, $C_i = v_1^i v_2^i \ldots v_{2k(i)}^i v_1^i$ such that $\chi(v_1^i v_2^i) = 1$, $\chi(v_{2k(i)}^i v_1^i) = 2$, and $X_i = \{v_1^i, v_3^i, \ldots, v_{2k(i)-1}^i\}$, $Y_i = V(C_i) - X_i$. We write $C_j \rightarrow C_i$ to denote that

$$\chi(X_i X_i) = \chi(X_i V(C_j)),\ \chi(Y_i Y_i) = \chi(Y_i V(C_j)) \text{ and } \chi(X_i X_i) \neq \chi(Y_i Y_i).$$

We point out that the meaning of $C_j \rightarrow C_i$ is that, for any choice of vertices $x \in V(C_j)$ and $y \in V(C_i)$, there exist alternating (x, y)-paths P and P' such that the colours of the edges incident with x in P and P' are distinct, but for every such choice of paths P and P', the colours of the edges in P and P' incident with y are equal. Hence, if $C_j \rightarrow C_i$, then the multigraph induced by the vertices of these two cycles is not colour-connected. (See Figure 16.7, where $C_2 \rightarrow C_3$.)

Lemma 16.8.10 *Suppose* G *has an alternating cycle factor* $\mathcal{F}_2 = C_1 \cup C_2$. *Then,* G *has an alternating Hamilton cycle if and only if neither* $C_1 \rightarrow C_2$ *nor* $C_2 \rightarrow C_1$. *Given a pair* C_1 *and* C_2 *of cycles of* G, *so that neither* $C_1 \rightarrow C_2$ *nor* $C_2 \rightarrow C_1$, *an alternating Hamilton cycle of* G *can be found in time* $O(|V(C_1)||V(C_2)|)$.

Proof: It is easy to see that if either $C_1 \rightarrow C_2$ or $C_2 \rightarrow C_1$, then G is not colour-connected. Hence, G has no alternating Hamilton cycle. Assume that neither $C_1 \rightarrow C_2$ nor $C_2 \rightarrow C_1$, but G has no alternating Hamilton cycle. Consider the bipartite digraph T with partite sets $V_1 = X_1 \cup X_2$ and $V_2 = Y_1 \cup Y_2$ obtained from G in the following way: delete all edges between vertices both on C_1 or on C_2 except those edges that are on the cycles and delete all edges between vertices both in the same partite set. Now make the following orientations of the edges in the resulting bipartite multigraph. For $i = 1, 2$ and any pair $v_1 \in V_1$, $v_2 \in V_2$, if there is an edge e between v_1 and v_2, then delete the colour of the edge e and orient it as the arc (v_i, v_{3-i}) if and only if $\chi(e) = i$.

Obviously, T has a spanning cycle subgraph consisting of two directed cycles Z_1, Z_2 which are orientations of the cycles C_1, C_2, respectively. Similarly we see that every directed cycle in T corresponds to an alternating cycle in G. Thus, since G has no alternating Hamilton cycle, T is not hamiltonian. By Exercise 6.23, this means that T is not strong, i.e., all arcs between Z_1 and Z_2 have the same orientation. Without loss of generality we may assume that all these arcs are oriented from Z_1 to Z_2. Then, by the definition of T, we obtain that $\chi(X_1Y_2) = 1,\ \chi(Y_1X_2) = 2$.

Consider next the bipartite digraph T' with partite sets $V_1' = X_1 \cup Y_2$ and $V_2' = Y_1 \cup X_2$. The rest of the definition of T' coincides with that of T. T' also contains a spanning cycle subgraph consisting of orientations of C_1 and C_2. Since G has no alternating Hamilton cycle, T' is not hamiltonian. By Corollary 6.6.16, this means that T' is not strongly connected. This leads us to the conclusion that either $\chi(X_1X_2) = 1$ and $\chi(Y_1Y_2) = 2$ or $\chi(X_1X_2) = 2$ and $\chi(Y_1Y_2) = 1$. The first possibility together with the conclusion of the previous paragraph implies $\chi(X_1V(C_2)) = 1, \chi(Y_1V(C_2)) = 2$. The second gives $\chi(X_2V(C_1)) = 2, \chi(Y_2V(C_1)) = 1$. Without loss of generality we may assume that $\chi(X_1V(C_2)) = 1, \chi(Y_1V(C_2)) = 2$.

Suppose that, for some $i \neq j$, there exists an edge $v^1_{2i+1}v^1_{2j+1}$ of colour 2. Then G has the alternating Hamilton cycle

$$v^2_1 v^1_{2j} v^1_{2j-1} \dots v^1_{2i+1} v^1_{2j+1} \dots v^1_{2k(1)} v^1_1 \dots v^1_{2i} v^2_{2k(2)} \dots v^2_1.$$

Hence, $\chi(X_1X_1) = 1$. Analogously, $\chi(Y_1Y_1) = 2$. Now $C_2 \to C_1$ and we have obtained a contradiction.

The complexity bound follows from that of Corollary 6.6.16. □

An alternating cycle subgraph $\mathcal{R}$ of G is **irreducible** if there is no other alternating cycle subgraph $\mathcal{Q}$ in G so that $V(\mathcal{R}) = V(\mathcal{Q})$ and $\mathcal{Q}$ has fewer cycles than $\mathcal{R}$. (See Figure 16.7.)

Theorem 16.8.11 *Let G have an alternating cycle factor $\mathcal{F}$ consisting of $p \geq 2$ cycles. $\mathcal{F}$ is an irreducible alternating cycle factor of G if and only if we can label the cycles in $\mathcal{F}$ as $C_1, \dots, C_p$, such that, with the notation introduced above, for every $1 \leq i < j \leq p$, $\chi(X_jV(C_i)) = 1,\ \chi(Y_jV(C_i)) = 2, \chi(X_jX_j) = 1, \chi(Y_jY_j) = 2$. An irreducible alternating cycle factor of G (if any) can be found in time $O(n^{2.5})$.*

Proof: If the edges have the structure described above, then $C_i \to C_j$ for all $i < j$ and each of the cycles in $\mathcal{F}$ forms a colour-connected component and $\mathcal{F}$ is clearly irreducible.

To prove the other direction we let $\mathcal{F}$ be an irreducible alternating cycle factor of G and let $p \geq 2$ be the number of cycles in $\mathcal{F}$. By Lemma 16.8.10, no two cycles in $\mathcal{F}$ induce a colour-connected subgraph. Thus, for all $1 \leq i < j \leq p$, either $C_i \to C_j$ or $C_j \to C_i$. Therefore, the digraph with vertex set $\{C_1, \dots, C_p\}$ and arc set $\{(C_i, C_j) : C_i \to C_j; 1 \leq i \neq j \leq p\}$

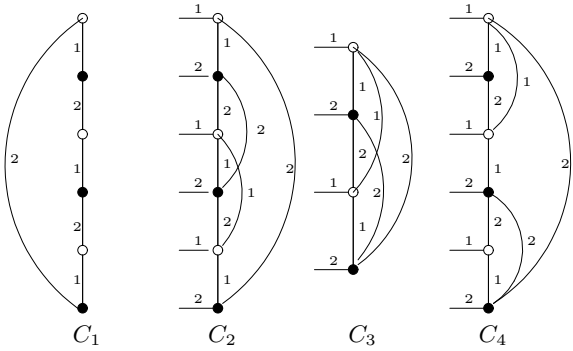


Figure 16.7 An irreducible PC cycle factor. The number $s \in \{1, 2\}$ on the edge emanating to the left from a vertex on C_i, $2 \leq i \leq 2$, indicates that the colour of all edges from that vertex to all the vertices of C_j with $j < i$ is s. The vertices are partitioned into two equal-sized sets indicated by black and white vertices. The number $r \in \{1, 2\}$ on an edge between two black (white vertices) on the same cycle indicates that all edges between black (white) vertices on that cycle have the same colour r.

is a tournament. So, if there exist cycles $C'_1, C'_2, \ldots, C'_k$ from $\mathcal{F}$ such that $C'_1 \rightarrow C'_2 \rightarrow \ldots \rightarrow C'_k \rightarrow C'_1$, then there also exists such a collection for $k = 3$ and the reader can easily find an alternating cycle covering precisely the vertices of those cycles, contradicting the irreducibility of $\mathcal{F}$. Hence we can assume that there is no such cycle. Thus there is a unique way to label the cycles in $\mathcal{F}$ as $C_1, C_2, \ldots, C_p$, so that $C_i \rightarrow C_j$ if and only if $i < j$. If there are three cycles C_i, C_j and C_k from $\mathcal{F}$ such that $C_i \rightarrow C_j, C_k$ and $C_j \rightarrow C_k$, but $\chi(X_k V(C_i)) \neq \chi(X_k V(C_j))$, then we can easily find an alternating cycle covering precisely the vertices of C_i, C_j and C_k, contradicting the irreducibility of $\mathcal{F}$. Hence we may assume that for all $1 \leq i < j \leq p$, $\chi(X_j V(C_i)) = 1$ and $\chi(Y_j V(C_i)) = 2$. The fact that $\chi(X_j X_j) = 1, \chi(Y_j Y_j) = 2$ follows from the proof of Lemma 16.8.10 and the minimality of $\mathcal{F}$.

Using the proof of Lemma 16.8.10, the proof above can be converted into an $O(n^2)$ algorithm for transforming any alternating cycle factor into an alternating hamiltonian cycle or an irreducible alternating cycle factor. Now the complexity bound of the lemma follows from a simple fact that one can find a spanning alternating cycle subgraph (if any) in a 2-edge-coloured multigraph L in time $O(|V(L)|^{2.5})$. Indeed, find maximum matchings in the red and blue subgraphs of L. Obviously, L has a spanning alternating cycle subgraph if and only if both subgraphs have perfect matchings. The complexity bound follows from that of the algorithm for finding a maximum matching in an arbitrary graph described in the book [306] by Even. □

We will make use of the following simple lemma.

Lemma 16.8.12 *Let $P = x_1x_2 \dots x_k$ be an alternating path and C an alternating cycle disjoint from P in G. Suppose $\chi(x_1V(C)) = i \neq \chi(x_1x_2)$ where $i = 1$ or $i = 2$ and that G contains an edge x_kz, where $z \in V(C)$ and $\chi(x_{k-1}x_k) \neq \chi(x_kz)$. If $\chi(x_kz) = i$, then G contains a cycle C' with $V(C') = V(P) \cup V(C)$. Otherwise G has a cycle C'' with $V(C'') = V(P) \cup V(C) - w$, where w is the neighbour of z on C for which $\chi(wz) = 3 - i$.*

Proof: Exercise 16.21. □

Proof of Theorem 16.8.4: Let $\mathcal{F} = C_1 \cup \ldots \cup C_p$ be an alternating cycle subgraph of G and let $\mathcal{F}' = C_1 \cup \ldots \cup C_{p-1}$. We will show by induction on p that G has an alternating cycle C^* having at least the same number of vertices as $\mathcal{F}$. If $p = 1$, we are done. So, we may suppose that $p \geq 2$. By Theorem 16.8.11, we may assume, using the (obvious) induction hypothesis, that, for all $1 \leq i < j \leq p$,

$$\chi(X_jV(C_i)) = 1,\ \chi(Y_jV(C_i)) = 2, \chi(X_jX_j) = 1, \chi(Y_jY_j) = 2. \qquad (16.4)$$

Since G is colour-connected there is an alternating (x, y)-path R of minimum length such that $x \in V(C_p)$, $\{y\} = V(R) \cap V(\mathcal{F}')$ and $\chi(xx') \neq \chi(xV(\mathcal{F}'))$, where x' is the successor of x in R. We prove that $(V(R) - \{x, y\}) \cap V(\mathcal{F}) = \emptyset$. Assume this is not so, that is, R contains at least two vertices from C_p. Consider a vertex z in $(V(R) \cap V(C_p)) - x$. Let z' be the successor of z in R. Clearly, $\chi(zz') = \chi(zV(\mathcal{F}'))$ since the (z, y)-part of R is shorter than R. On the other hand, by (16.4) x' is not in C_p and by the minimality of R, $\chi(x'V(\mathcal{F}')) = \chi(xx')$. Then, the alternating path Qv, where Q is the reverse of the (x', z)-part of R and v is a vertex in C_{p-1}, is shorter than R; a contradiction.

Now consider an alternating (x, y)-path R with the properties above including $(V(R)-\{x, y\})\cap V(\mathcal{F}) = \emptyset$. We may assume without loss of generality that $x = v_1^p$ and $\chi(xV(\mathcal{F}')) = \chi(v_2^pv_1^p)$. Choose t such that $y \in V(C_t)$. Apply Lemma 16.8.12 to the path

$$v^p_{2k(p)}v^p_{2k(p)-1} \dots v_2^pR',$$

where R' is the path R without y, and the cycle C_t. We get a new alternating cycle C', with $V(C') \subset V(R) \cup V(C_t) \cup V(C_p)$, covering at least as many vertices as C_t and C_p together, so by replacing C_t and C_p by C' in $\mathcal{F}$, we obtain a new alternating cycle subgraph with fewer cycles which covers at least as many vertices as $\mathcal{F}$ and the existence of C^* follows by induction. □

The proof above can be converted into an $O(n^3)$ algorithm for finding a longest cycle in G, provided we are given a maximum cycle subgraph as input.

16.9 PC Paths and Cycles in c-Edge-Coloured Complete Graphs, $c \geq 3$

Let K_n^c denote a c-edge-coloured complete graph with n vertices. The properly coloured (PC) Hamilton path problem for c-edge-coloured complete graphs is much more difficult in the case $c \geq 3$, than in the case $c = 2$ treated above. However, a hamiltonian analog of Theorem 16.8.1 still holds as was shown by Feng, Giesen, Guo, Gutin, Jensen and Rafiey.

Theorem 16.9.1 *[313] A K_n^c ($c \geq 3$) has a PC Hamilton path if and only if K_n^c contains a PC spanning 1-path-cycle subgraph.* □

Observe that to prove Theorem 16.9.1 it suffices to show that given a pair $C = v_1v_2 \dots v_nv_1$ ($n \geq 3$) and $P = u_1u_2 \dots u_m$ ($m \geq 1$) of disjoint PC cycle and PC path in K_n^c ($c \geq 3$) covering all vertices of K_n^c, one can find a PC Hamilton path. If $m = 1$, at least one of the paths $u_1v_1v_2 \dots v_nv_1$ and $u_1v_1v_n v_{n-1} \dots v_1$ is PC. Thus, we may assume that $m \geq 2$. Let $j \in \{1, 2, \dots, n\}$. If $\chi(u_1v_j) \neq \chi(u_1u_2)$, then at least one of the paths $u_mu_{m-1} \dots u_1v_jv_{j+1} \dots v_{j-1}$ and $u_mu_{m-1} \dots u_1v_jv_{j-1} \dots v_{j+1}$ is PC. Similarly there exists a PC Hamilton path if $\chi(u_mv_j) \neq \chi(u_{m-1}u_{m-2})$. So we may assume the following:

$$c(u_1v_j) = c(u_1u_2) \text{ and } c(u_mv_j) = c(u_{m-1}u_m) \text{ for each } j \in [n]. \qquad (16.5)$$

Thus, to show Theorem 16.9.1 it suffices to prove the following claim:

Theorem 16.9.2 *If (16.5) is satisfied, then there exists a PC Hamilton path H in G with u_1 as its first vertex, such that the initial edge of H is either u_1u_2 or one of the edges u_1v_j ($1 \leq j \leq n$), and such that if $m \geq 2$, then u_m is the last vertex of H and the last edge of H is either $u_{m-1}u_m$ or one of the edges v_ju_m ($1 \leq j \leq n$).* □

The proof of this theorem in the paper [313] is quite lengthy and, thus, is not presented here.

By Theorems 16.4.6 and 16.9.2, the PC Hamilton path problem for c-edge-coloured complete graphs is polynomial time solvable. The proof of Theorem 16.9.2 is constructive and can be turned into a polynomial time algorithm for transforming a PC 1-path-cycle factor into a PC Hamilton path.

Benkouar, Manoussakis, Paschos and Saad posed the following problem:

Problem 16.9.3 *[137] Determine the complexity of the PC Hamilton cycle problem for c-edge-coloured complete graphs when $c \geq 3$.*

Gutin and Kim believe that the complexity is polynomial.

Conjecture 16.9.4 *[465]For every $c \geq 3$ the PC Hamilton cycle problem for c-edge-coloured complete graphs is polynomial time solvable.*

Another interesting problem is to find a non-trivial characterization of c-edge-coloured ($c \geq 3$) complete graphs containing PC hamiltonian cycles. We consider results from [137] related to Problem 16.9.3. We give an example showing that the obvious analogue of Corollary 16.8.5 is not valid for $c \geq 3$. Later we present some conditions which guarantee the existence of a PC Hamilton cycle in a c-edge-coloured complete graph.

A **strictly alternating cycle** in K_n^c is a cycle of length pc (p is an integer) so that the sequence of colours $(12\ldots c)$ is repeated p times. Benkouar, Manoussakis, Paschos and Saad [137] proved the following:

Theorem 16.9.5 *[137] Let $c \geq 3$. The problem of determining the existence of a strictly alternating Hamilton cycle in K_n^c is $\mathcal{NP}$-complete.*

Proof: Exercise 16.22. □

The following result shows that if we relax the property of colours to be at strict places, but maintain the number of their appearances in a Hamilton cycle, then we still have an $\mathcal{NP}$-complete problem.

Theorem 16.9.6 *[137] Given positive integers p and $c \geq 3$, the problem of determining the existence of a PC Hamilton cycle C of K_{cp}^c so that each colour appears p times in C is $\mathcal{NP}$-complete.*

Proof: Exercise 16.23. □

The following example shows that the obvious analogue of Corollary 16.8.5 is not valid for $c \geq 3$. The graph G_6 is a 3-edge-coloured complete graph on vertices 1,2,3,4,5,6. All the edges of G_6 have colour 1 except for the following: the triangles 2342 and 2562 have colours 2 and 3, respectively, $\chi(36) = \chi(45) = 2$, $\chi(12) = 3$. It is easy to check that G_6 is colour-connected and has the alternating spanning cycle subgraph $1231 \cup 4564$, but G_6 contains no PC Hamilton cycle (Exercise 16.24). Note that alternating paths showing that G_6 is colour-connected may be chosen so that for each choice of vertices x and y the two paths P and P' described in the definition of colour-connectivity are internally disjoint. Hence it will not be enough to change this definition to require that P and P' are disjoint, a condition which is obviously necessary for the existence of a PC Hamilton cycle. For every even n, using the definition of G_6, one can easily construct a 3-edge-coloured complete graph on $n \geq 8$ vertices which is colour-connected and has a PC spanning cycle subgraph, but contains no PC Hamilton cycle (see Exercise 16.25).

We start our consideration of sufficient conditions for an edge-coloured complete graph to contain a PC Hamilton cycle with the following simple result by Manoussakis, Spyratos, Tuza and Voigt:

Proposition 16.9.7 *[684] If $c \geq \frac{1}{2}(n-1)(n-2)+2$, then every K_n^c has a PC Hamilton cycle.*

Proof: Exercise 16.26. □

To see that the bound of Proposition 16.9.7 is sharp consider the following K_n^c. Assign colour 1 to all edges incident to a fixed vertex $x \in V(K_n^c)$. Each of the remaining edges has a distinct colour not equal 1. Clearly, such K_n^c has no PC Hamilton cycle and $c = \frac{1}{2}(n-1)(n-2)+1$.

In [251] Daykin posed the following interesting problem. Find a positive constant d such that every K_n^c with $\Delta_{mon}(K_n^c) \leq dn$ has a PC Hamilton cycle. This problem was independently solved by Bollobás and Erdős [163] and Chen and Daykin [200]. In [163] (in [200], respectively), it was proved that if $69\Delta_{mon}(K_n^c) < n$ ($17\Delta_{mon}(K_n^c) \leq n$, respectively), then K_n^c has a PC Hamilton cycle. Shearer [814] improved the last result showing that if $7\Delta_{mon}(K_n^c) < n$, then K_n^c has a PC Hamilton cycle. So far, the best asymptotic estimate was obtained by Alon and Gutin [24].

Theorem 16.9.8 *[24] For every $\epsilon > 0$ there exists an $n_0 = n_0(\epsilon)$ so that for each $n > n_0$, every K_n^c satisfying*

$$\Delta_{mon}(K_n^c) \leq (1 - \frac{1}{\sqrt{2}} - \epsilon)n \quad (\ = (0.2928\ldots - \epsilon)n \ ) \tag{16.6}$$

contains a PC Hamilton cycle. □

However, Theorem 16.9.8 seems to be far from the best possible, at least, if the following conjecture by Bollobás and Erdős [163] is true.

Conjecture 16.9.9 *Every K_n^c with $\Delta_{mon}(K_n^c) \leq \lfloor n/2 \rfloor - 1$ has a PC Hamilton cycle.*

The rest of this subsection is devoted to a probabilistic[1] proof of Theorem 16.9.8. For simplicity we assume first that $n = 2m$ is even, and remark at the end of the subsection how to modify the argument for the case of odd n. Fix a positive ϵ, and let $K = K_n^c$ be an edge-coloured complete graph on $n = 2m$ vertices satisfying (16.6). We first prove the following lemma.

Lemma 16.9.10 *For all sufficiently large m, K contains a spanning edge-coloured complete bipartite graph $K_{m,m}^c$ satisfying*

$$\Delta_{mon}(K_{m,m}^c) \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m. \tag{16.7}$$

Proof: Let $u_i v_i$ $(1 \leq i \leq m)$ be an arbitrary perfect matching in K and choose a random partition of the set of vertices of K into two disjoint subsets A and B of cardinality m each by choosing, for each i, $1 \leq i \leq m$, randomly and independently, one element of the set $\{u_i, v_i\}$ to be a member of A and

[1] Probabilistic methods have proved to be very powerful for various problems (see e.g. the book [28] by Alon and Spencer).

the other to be a member of B. Fix a vertex w of K and a colour, say red, that appears in the edge-colouring of K. The number of neighbours a of w in A so that the edge wa is red can be written as a sum of m independent indicator random variables $x_1, \ldots, x_m$, where x_i is the number of red neighbours of w in A among u_i, v_i. Thus each x_i is either 1 with probability one (in case both edges wu_i, wv_i are red) or 0 with probability 1 (in case none of the edges wu_i, wv_i is red) or 1 with probability 1/2 (in case exactly one of these two edges is red). It follows that if the total number of red edges incident with w is r, then the probability that w is adjacent with more than $(r+s)/2$ vertices in A by red edges is equal to the probability that more than $(q+s)/2$ flips among q independent flips of a fair coin give 'heads', where q is the number of non-constant indicator random variables among the x_i's. This can be bounded by the well-known inequality of Chernoff (cf. e.g. [28, Theorem A.4, page 235]) by $e^{-2s^2/q} < e^{-2s^2/m}$. Since the same argument applies to the number of 'red' neighbours of w in B, and since there are less than $8m^3$ choices for a vertex w, a colour in the given colouring of K and a partite set (A or B), we conclude that the probability that there exists a vertex with more than

$$(1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m$$

neighbours of the same colour in either A or B is at most

$$8m^3 e^{-2\epsilon^2 m},$$

which is (much) smaller than 1 for all sufficiently large m. Therefore, there exists a choice for A and B so that the above does not occur, completing the proof. □

The next lemma can be proved by applying a large deviation result for martingales, i.e., Azuma's inequality [28].

Lemma 16.9.11 *[24] Let U be a subset of $M = [m-1]$ and suppose that for each $u \in U$ there is a subset $S_u \subset M$, where $|S_u| \leq r$ for all u. Let $f : U \mapsto M$ be a random one-to-one mapping of U into M, chosen uniformly among all one-to-one mappings of U into M, and define:*

$$B(f) = |\{u \in U : f(u) \in S_u\}|.$$

Then the expectation of $B(f)$ is given by

$$E = E(B(f)) = \sum_{u \in U} \frac{|S(u)|}{m-1} \quad (\leq \frac{|U|r}{m-1}),$$

and the probability that $B(f)$ is larger satisfies the following inequality. For every $\lambda > 0$

$$\text{Prob}[B(f) - E > 4\lambda\sqrt{m-1}] < e^{-\lambda^2}.$$

□

Corollary 16.9.12 *Let $K^c_{m,m}$ be an edge-coloured complete bipartite graph on the partite sets A and B, and suppose that (16.7) holds. Then, for all sufficiently large m, there exists a perfect matching a_ib_i, $1 \leq i \leq m$, in $K^c_{m,m}$ so that the following two conditions hold.*
(i) For every i the number $d^+(i)$ of edges a_ib_j between a_i and B whose colours differ from those of a_ib_i and of a_jb_j is at least $m/2+1$.
(ii) For every j the number $d^-(j)$ of edges a_ib_j between b_j and A whose colours differ from those of a_ib_i and of a_jb_j is at least $m/2+1$.

Proof: Let a_ib_i, $1 \leq i \leq m$, be a random perfect matching between A and B, chosen among all possible matchings with uniform probability. Put $r = \Delta_{mon}(K^c_{m,m})$ and notice that by (16.7)

$$r \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m.$$

Fix an i, say $i = m$, and let us estimate the probability that the condition (i) fails for i. Suppose the edge a_mb_m has already been chosen for our random matching, and the rest of the matching still has to be chosen randomly. There are at most r edges a_mb ($b \in B$) having the same colour as a_mb_m. Let U be the set of all the remaining elements B. Then $|U| \geq m - r$. For each $u \in U$, let S_u denote the set of all elements $a \in A - a_m$ so that the colour of the edge au is equal to that of the edge a_mu. The random matching restricted to U is simply a random one-to-one function f from U to $A - a_m$. Moreover, the edge a_mu will not be counted among the edges incident with a_m and having colours that differ from those of a_mb_m and of the edge matched to u if and only if the edge matched to u will lie in S_u. It follows that the random variable counting the number of such edges of the form a_mu behaves precisely like the random variable $B(f)$ in Lemma 16.9.11. By choosing say, $\lambda = \sqrt{\log(4m)}$ we conclude that the probability that $B(f)$ exceeds $|U|r/(m-1) + 4\lambda\sqrt{m-1}$ is smaller than $1/(4m)$. Therefore, with probability at least $1 - \frac{1}{4m}$

$$\begin{aligned} d^+(m) &\geq |U| - \frac{|U|r}{m-1} - 4\sqrt{m}\sqrt{\log(4m)} \\ &\geq \frac{(m-r)(m-r-1)}{m-1} - 4\sqrt{m}\sqrt{\log(4m)} \\ &> m/2+1, \end{aligned}$$

for all sufficiently large m (using the fact that $r \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m$).

Since there are m choices for the vertex a_i (and similarly m choices for the vertex b_j for which the computation is similar) we conclude that with probability at least a half $d^+(i) > m/2+1$, and $d^-(j) > m/2+1$ for all i and j. In particular there exists such a matching, completing the proof of the corollary. □

Returning to the proof of Theorem 16.9.8 with $n = 2m$, and given an edge-coloured K^c_n satisfying (16.6) apply Lemma 16.9.10 and Corollary 16.9.12 to

obtain a matching a_ib_i satisfying the two conditions in the corollary. Construct a digraph $D = (V, E)$ on the set of vertices $V = \{v_1, v_2, \ldots, v_m\}$ by letting v_iv_j be a directed edge (for $i \neq j$) if and only if the colour of a_ib_j in K_n^c differs from that of a_ib_i and that of a_jb_j. By Corollary 16.9.12 the in-degree and the out-degree of every vertex of D exceed $m/2$, implying, by Corollary 6.4.3, that D contains a directed Hamilton cycle $v_{\pi(1)}v_{\pi(2)} \ldots v_{\pi(m)}v_{\pi(1)}$, where $\pi = \pi(1), \pi(2), \ldots, \pi(m)$ is a permutation of $\{1, 2, \ldots, m\}$. The cycle $b_{\pi(1)}a_{\pi(1)}b_{\pi(2)}a_{\pi(2)} \ldots b_{\pi(m)}a_{\pi(m)}b_{\pi(1)}$ is clearly a PC Hamilton cycle in K_n^c, as needed.

In case $n = 2m+1$ is odd we fix a path $P = a_1c_1b_1$ of length 2, so that the edges a_1c_1 and c_1b_1 have distinct colours, choose a random perfect matching $a_2b_2, \ldots, a_mb_m$ in the rest of the graph and show that with high probability there is a PC Hamilton cycle containing the path P and the matching by applying Corollary 6.4.3 as before. Since the details are almost identical to the ones for the even case, we omit them. This completes the proof of the theorem. □

16.10 Exercises

16.1. Prove Proposition 16.1.1. Hint: use the BB-correspondence, Häggkvist's transformation, Proposition 10.1.2 and Theorem 10.2.1.

16.2. (−) Deduce from Theorem 16.2.1 that an undirected multigraph G has an eulerian trail if G is connected and each vertex of G is of even degree.

16.3. Prove the correctness of Pevzner's algorithm described after Theorem 16.2.1.

16.4. (−) Every eulerian digraph has a cycle (unless it is the trivial digraph with one vertex). Show that the corresponding claim is not valid for alternating trails and cycles in 2-edge-coloured graphs.

16.5. Let G be a connected 2-edge-coloured graph. Let $V(G) = X + Y$ such that $d_1(x) = d_2(x)$ for every $x \in X$, and $d_1(y) = d_2(y) - 1$ for every $y \in Y$. What is the minimum number of edge-disjoint alternating trails to cover $E(G)$?

16.6. Prove Corollary 16.3.3.

16.7. **Every bridgeless graph G has an M-alternating cycle for a given perfect matching M of G.** Let M be a perfect matching in a graph G. Using Theorem 16.3.2 prove that if no edge of M is a bridge of G, then G has a cycle whose edges are taken alternatively from M and $G - M$ (Grossman and Häggkvist [426]).

16.8. (+) Let G be a 2-edge-coloured eulerian graph so that all monochromatic degrees are odd. Using Theorem 16.3.2 demonstrate that G has an alternating cycle (Grossman and Häggkvist [426]).

16.9. (−) Show that the tree P-gadgets defined in Subsection 16.4.1 satisfy the properties P1-P4 also defined in Subsection 16.4.1.

16.10. Prove that the XP-gadget defined in Subsection 16.4.1 has the minimum number of vertices and edges among all possible P-gadgets for $c = 2, 3, 4$.

16.11. Prove Theorem 16.4.4.

16.12. Prove Proposition 16.6.1.

16.13. Prove Proposition 16.7.1. Hint: see Exercise 1.17.

16.14. Prove Theorem 16.7.2 using the BD-correspondence and Corollary 6.4.3.

16.15. Deduce Theorem 16.7.2 from Theorem 16.7.3.

16.16. Show that the conditions of Theorem 16.8.3 are necessary. Hint: it is similar to the remark after Theorem 16.7.5.

16.17. Derive Theorem 16.7.5 from Theorem 16.8.3. Hint: you may use the DHM-construction.

16.18. (+) Prove Theorem 16.8.6.

16.19. (+) Prove Theorem 16.8.8.

16.20. Give a direct proof of Corollary 16.8.2.

16.21. Prove Lemma 16.8.12.

16.22. Prove Theorem 16.9.5.

16.23. Prove Theorem 16.9.6.

16.24. Check that G_6 introduced after Theorem 16.9.6 is colour-connected and has the alternating spanning cycle subgraph $1231 \cup 4564$, but does not contain a PC Hamilton cycle.

16.25. Using the definition of G_6 given after Theorem 16.9.6, construct, for every even n, a 3-edge-coloured complete graph on $n \geq 8$ vertices which is colour-connected and has a PC spanning cycle subgraph, but contains no PC Hamilton cycle.

16.26. Prove Proposition 16.9.7. Hint: consider the complete biorientation of a maximum spanning subgraph G of K_n^c such that no pair of edges in G is of the same colour. Apply Exercise 6.12 to see that $\overleftrightarrow{G}$ is hamiltonian.

17. Applications of Digraphs and Edge-Coloured Graphs

In this chapter we study applications of directed and edge-coloured graphs to quantum mechanics, embedded computing, the traveling salesman problem, constraint satisfaction, genetics and other areas.

17.1 A Digraph Model in Quantum Mechanics

The class of mediated digraphs defined below was introduced by Jones, Linden and Massar [574] as a model in quantum mechanics. In this section, we define and study an extremal parameter of digraphs in this class, the nth mediation number. The parameter is of interest in the study of quantum non-locality, which we discuss in the last subsection.

A digraph D is **mediated** if for every pair x, y of vertices there is a vertex z such that both $x, y \in N^-[z]$ (possibly $z = x$ or y). Semicomplete digraphs and symmetric digraphs of diameter 2 are special families of mediated digraphs. Figure 17.1 is another example of a mediated digraph.

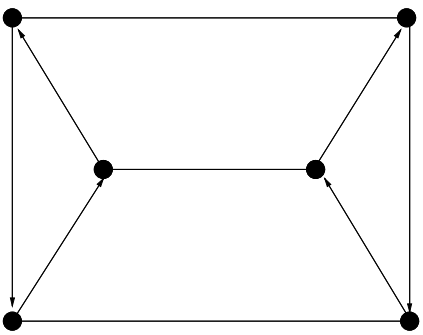


Figure 17.1 A mediated digraph H of order 6. Undirected edges correspond to directed 2-cycles.

The nth **mediation number** $\mu(n)$ is the minimum of $\Delta^-(D)$ over all mediated digraphs on n vertices. This parameter is of interest in quantum

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 643
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_17,

mechanics as explained in the last subsection. Figure 17.1 shows that $\mu(6) \leq 2$. One can see that by Proposition 17.1.1 we have $\mu(6) \geq 2$. Thus, in fact, $\mu(6) = 2$ and H in Figure 17.1 is an 'optimal' mediated digraph.

This section is based on the paper [464] by Gutin, Jones, Rafiey, Severini and Yeo. The rest of the section is organized as follows. In Subsection 17.1.1, we obtain a lower bound $f(n)$ for $\mu(n)$, which is proved to be sharp in the next two subsections. In the next subsection, we obtain a characterization of $\mu(n)$ as an extremal parameter of special families of sets. This allows us to use some results from design theory. Subsection 17.1.3 provides upper bounds for $\mu(n)$. We prove that $\mu(n) = f(n)(1 + o(1))$, which is the central result of the section and is of importance for quantum non-locality (see Subsection 17.1.5). In Subsection 17.1.4 we show that $\mu(n) > f(n)$ for an infinite number of values of n. It is conjectured in [464] that, in fact, there is a constant c such that $\mu(n) \leq f(n) + c$ for each $n \geq 1$. The authors of [464] also posed the problem of checking whether $\mu(n)$ is a monotonically increasing function. Subsection 17.1.5 provides a brief motivation for the study of mediated digraphs and the nth mediation number.

17.1.1 Lower Bound for $\mu(n)$

Let $f(n) = \lceil \frac{1}{2}(\sqrt{4n-3} - 1) \rceil$. The following proposition gives a lower bound for $\mu(n)$, which is the exact value of $\mu(n)$ for infinitely many values of n (see Corollaries 17.1.6 and 17.1.8).

Proposition 17.1.1 *[464] For each $n \geq 1$, we have $\mu(n) \geq f(n)$.*

Proof: Let D be a mediated digraph and let $d = \Delta^-(D)$. If D has just one vertex, the bound holds, so we may assume that $n \geq 2$. By the definition of a mediated digraph, each pair x, y of vertices of D belongs to the closed in-neighborhood of some vertex. Let $d_1^-, d_2^-, \ldots, d_n^-$ be the in-degrees of vertices $v_1, v_2, \ldots, v_n$ of D. Since a vertex v_i has $\binom{d_i^-}{2} + d_i^-$ pairs of vertices in its closed in-neighborhood and since D has overall $\binom{n}{2}$ pairs of vertices, we have

$$\sum_{i=1}^{n} \left(\binom{d_i^-}{2} + d_i^- \right) \geq \binom{n}{2}.$$

Therefore, we have $\sum_{i=1}^{n} ((d_i^-)^2 + d_i^-) \geq n(n-1)$. So, $n(d^2 + d) \geq n(n-1)$ and $d \geq \frac{1}{2}(\sqrt{4n-3} - 1)$ and the result follows by integrality of d. □

17.1.2 Families of Sets and $\mu(n)$

In this subsection we characterize $\mu(n)$ in terms of special families of sets. Symmetric families, 2-covering families and families having a system of distinct representatives are of significant interest in the theory and applications of combinatorics, see, e.g., [189, 579].

We consider families of subsets of a finite set X. Using block-design terminology, we call the elements of X **points** and the subsets of X **blocks**. Let $\mathcal{F} = \{X_1, X_2, \ldots, X_m\}$ be a family of blocks of X. An m-tuple $S = (x_1, x_2, \ldots, x_m)$ is a **system of distinct representatives** (**SDR**) if all points of S are distinct and $x_i \in X_i$ for each $i = 1, 2, \ldots, m$. The family $\mathcal{F}$ is **symmetric** if $m = |X|$. A family $\mathcal{F}$ is **2-covering** if, for each pair $j, k \in X$, there exists $i \in \{1, 2, \ldots, m\}$ such that $\{j, k\} \subseteq X_i$. Let mcard($\mathcal{F}$) be the maximum cardinality of a block in $\mathcal{F}$. We call $\mathcal{F}$ **mediated** if it is symmetric, 2-covering and has an SDR. Let $\mu^-(n)$ be the minimum mcard($\mathcal{F}$) over all mediated families on the set $[n]$.

We have the following:

Proposition 17.1.2 *[464] For each $n \geq 1$, $\mu(n) = \mu^-(n) - 1$.*

Proof: Let D be a mediated digraph on vertices $[n]$ with $\Delta^-(D) = \mu(n)$. By the definition of a mediated digraph, the family $\mathcal{N} = \{N^-[i] : \ i \in [n]\}$ is 2-covering. Clearly, $(1, 2, \ldots, n)$ is an SDR of $\mathcal{N}$. Thus, $\mathcal{N}$ is mediated and $\mu^-(n) \leq \mu(n) + 1$.

Let $\mathcal{F} = \{X_1, X_2, \ldots, X_n\}$ be a mediated family on $[n]$ with mcard($\mathcal{F}$) = $\mu^-(n)$. Since $\mathcal{F}$ has an SDR (since it is mediated), without loss of generality, we may assume that $i \in X_i$. Construct a digraph D with $V(D) = \{1, 2, \ldots, n\}$ and $N^-[i] = X_i$. Since $\mathcal{F}$ is 2-covering, D is mediated and $\mu(n) \leq \mu^-(n) - 1$. This inequality and $\mu^-(n) \leq \mu(n) + 1$ imply that $\mu(n) = \mu^-(n) - 1$. □

Let $n > k \geq 2$ and $\lambda \geq 1$ be integers. A family $\mathcal{F} = \{X_1, X_2, \ldots, X_b\}$ of blocks on X is called an (n, k, λ)-**design** if $|X| = n$, each block has k points and every pair of distinct points is contained in exactly λ blocks. Recall that an (n, k, λ)-design is symmetric if it has n blocks, i.e., $b = n$. A **projective plane of order** q is a symmetric $(q^2 + q + 1, q + 1, 1)$-design for some integer $q > 1$. For a family $\mathcal{F}$ of blocks and a point i, let $d(i)$ denote the number of blocks containing i. The following two theorems are well-known, see, e.g., [189, 579].

Theorem 17.1.3 *For each prime power q, there exists a projective plane of order q.* □

Theorem 17.1.4 *Let $\mathcal{S} = \{X_1, X_2, \ldots, X_n\}$ be a family of subsets of $\{1, 2, \ldots, n\}$ and let r be a natural number such that $|X_i| = d(i) = r$ for each $i = 1, 2, \ldots, n$. Then $\mathcal{S}$ has an SDR.* □

The last theorem can be used to prove the following:

Proposition 17.1.5 *Every symmetric (n, k, λ)-design is a mediated family of blocks.*

Proof: Let $\mathcal{F} = \{X_1, X_2, \ldots, X_b\}$ be an (n, k, λ)-design on X, $|X| = n$. It is well-known (see, e.g., [189]) that, for all such designs, there is a constant r

such that $r = d(i)$ for each point i. The parameters n, k, λ, b and r also satisfy the following two equalities: $bk(k-1) = \lambda n(n-1)$ and $r(k-1) = \lambda(b-1)$. Assume that $\mathcal{F}$ is symmetric. Using $b = n$ and the two equalities, we easily conclude that $r = k$. It now follows from Theorem 17.1.4 that $\mathcal{F}$ has an SDR. Since $\mathcal{F}$ is symmetric and 2-covering ($\lambda \geq 1$), $\mathcal{F}$ is mediated. □

Now we are ready to compute an infinite number of values of $\mu(n)$.

Corollary 17.1.6 *[464] For each prime power q, $\mu(q^2+q+1) = f(q^2+q+1) = q$.*

Proof: Let $n = q^2+q+1$. By Theorem 17.1.3 and Propositions 17.1.1 and 17.1.5, we have $f(n) \leq \mu(n) = \mu^-(n) - 1 \leq q$. However, one can trivially verify that $f(n) = q$. □

17.1.3 Upper Bounds for $\mu(n)$

We give the next theorem without a proof.

Theorem 17.1.7 *[464] Let $n = q^2+q+1+m(q+1)-t$, where q is a prime power, $1 \leq m \leq q+1$ and $0 \leq t \leq q$. Then $\mu(n) \leq q+m$.* □

Corollary 17.1.8 *[464] Let q be a prime power. If s is an integer such that $q^2+q+2 \leq s \leq q^2+2q+2$, then $\mu(s) = f(s) = q+1$.*

Proof: Let s be an integer such that $q^2+q+2 \leq s \leq q^2+2q+2$. By Theorem 17.1.7 for $m = 1$, $\mu(s) \leq q+1$. By Proposition 17.1.1, $q+1 \geq \mu(s) \geq f(s) \geq f(q^2+q+2)$. Thus, it suffices to show that $f(q^2+q+2) = q+1$, which is easily verifiable. □

The following number-theoretical result was proved by Baker, Harman and Pintz [57].

Theorem 17.1.9 *There is a real x_0 such that for all $x > x_0$ the interval $[x, x+x^\alpha]$, where $\alpha = 0.525$, contains prime numbers.* □

The last two assertions imply the following:

Theorem 17.1.10 *We have $\mu(n) = f(n)(1+o(1))$.*

Proof: Let n be sufficiently large. Let p and q be a pair of consecutive primes such that $p^2+p+1 \leq n < q^2+q+1$, and let $d = q^2+q-p^2-p$. By Theorem 17.1.7, $\mu(n) \leq p + \lceil d/(p+1) \rceil$. By Theorem 17.1.9, $q-p \leq p^\alpha$. Thus, $d = (q+p+1)(q-p) \leq 3p\dot{p}^\alpha = 3p^{1+\alpha}$. So, $\mu(n) \leq p+3p^\alpha+1 = p(1+o(1)) = f(p^2+p+1)(1+o(1)) \leq f(n)(1+o(1))$. □

The authors of [464] believe that the following holds for a small constant c. If this conjecture holds, they would like to know the smallest value of c.

Conjecture 17.1.11 *[464] There is a constant c such that $\mu(n) \leq f(n)+c$ for each n.*

17.1.4 When $\mu(n) > f(n)$

Corollaries 17.1.6 and 17.1.8 may prompt some to suspect that $\mu(n) = f(n)$ holds for each $n \geq 1$. However, this is not the case.

One of the best-known conjectures in combinatorics is that a projective plane does not exist if q is not a prime power. The celebrated Bruck-Ryser theorem (see, e.g., [189]) proves that if a projective plane of order q exists, where $q \equiv 1$ or 2 (mod 4), then q is the sum of two squares of integers. This gives infinitely many values of q for which there is no projective plane of order q (for example, every number $q = 2p$, where p is a prime congruent to 3 mod 4). The fact that there are infinitely many primes congruent to 3 mod 4 follows from the famous Dirichlet's theorem: every arithmetic progression with common difference relatively prime to the initial term contains infinitely many prime numbers (see, e.g., [715]). The above implies the following:

Theorem 17.1.12 *There are infinitely many positive integers q for which there is no projective plane of order q.* □

The proof of the following theorem can be found in [464] (it does not use digraph theory).

Theorem 17.1.13 *If there is no projective plane of order q, then $\mu(q^2+q+1) > f(q^2+q+1)$.* □

This theorem and Theorem 17.1.12 imply the following:

Corollary 17.1.14 *For an infinite number of values of n, $\mu(n) > f(n)$.* □

The following problem is of definite interest.

Problem 17.1.15 *[464] Is $\mu(n) \leq \mu(n+1)$ for each n?*

17.1.5 Mediated Digraphs in Quantum Mechanics

Non-locality is a fundamental, and curious, feature of quantum theory which confused Albert Einstein and continues to yield exciting results in physics (there are numerous popular explanations of non-locality, a more technical review is [901]). The study of non-locality is sometimes helped by considering classical analogies: it was in this endeavor that mediated digraphs were discovered (see [574] by Jones, Linden and Massar, where mediated digraphs are called Totally Paired Graphs). Consider two objects which are connected and then suddenly sent to such widely separated locations that they can no longer influence each other on relevant time scales. The results of local measurements on each member of a pair of classical objects, which have been connected and separated in this fashion, can be correlated, depending on their

relationship when they were together. Perhaps, when they were together, the objects exchanged some information.

The correlations between the results of certain sets of local measurements on some pairs of quantum objects, which have been connected and separated, cannot be explained by allowing only an exchange of information when they started together. This is an aspect of non-locality. Since it is the case that measurements on quantum objects can show classical correlations but the reverse is not true, there is a sense in which quantum objects have correlations beyond allowed classical ones.

A standard classical analogy for quantum non-locality is as follows. Classical separated objects are allowed to cheat and exchange information, faster than light, about the way they are to be measured. In this case the outcomes of a measurement on one object could indeed depend on the way the other object is measured. The correlations present in sets of quantum objects can now be classically approximated. One can ask how many bits of information have to be exchanged between classical objects in order to fool experimentalists into thinking they are measuring a quantum state. The classical objects are given an extra property; their characteristics can depend on the way other, distant, objects are measured. How much of this freedom need one allow in order to produce scenarios which can have the same measurement results as measurements of quantum objects?

In the scenario considered in [574] each object (a vertex) knows how it is to be measured and can send this information to other vertices (an arc from source vertex to target vertex). This information stays put on receipt and does not propagate around the graph. The measurement results of vertices, given that their properties might now depend on the way their neighbors are to be measured, are more general and now have a chance to reproduce quantum correlations. It was shown that it is necessary that the vertices be connected as a mediated digraph if they are to fool experimentalists into thinking they are measuring a quantum state.

Given that n classical objects linked as any mediated digraph can sometimes be at least as non-local as n quantum mechanical objects, it is interesting to find out how sparse these digraphs are. One would like to consider the least sparse members of the set of mediated digraphs – the least sparse digraphs that can still be at least as non-local as quantum states. In order to achieve this one must have a good measure of sparsity: we consider $\Delta^-(D)$. If an n-vertex digraph contains a vertex which depends on the settings of lots of other vertices, $\Delta^-(D)$ will be large: this defines a highly non-local pattern – one vertex is highly correlated with many others. If all vertices in a digraph are only connected to a few others, $\Delta^-(D)$ will be small: such digraphs seem to have a form of short-range non-locality. Proving that, for any n, there are mediated digraphs which have $\Delta^-(D)$ scaling with $\sqrt{n}$ (Theorem 17.1.10), shows that each object need only be connected to a fraction of the set of objects which *diminishes* as n increases (as $1/\sqrt{n}$). As n increases there exist

mediated digraphs in which each vertex becomes increasingly localized with respect to the whole – this must be telling us something about quantum non-locality.

17.2 Embedded Computing and Convex Sets in Acyclic Digraphs

In this section we will introduce convex sets in acyclic digraphs, give motivation for studying them and consider algorithmic and theoretical results on convex and connected convex sets.

A set X of vertices of an acyclic digraph D is **convex** if $X \neq \emptyset$ and there is no path between vertices of X which contains a vertex not in X. A set X is **connected** if $X \neq \emptyset$ and the underlying graph of $D\langle X\rangle$ is connected. A set is **connected convex** (a **cc-set**) if it is both connected and convex.

17.2.1 Embedded Computing Systems and Convex Sets

An embedded or application-specific computing system only ever executes a single application. Examples include automobile engine management systems, satellite and aerospace control systems and the signal processing parts of mobile cellular phones. Significant improvements in the price-performance ratio of such systems can be achieved if the instruction set of the application-specific processor is specifically tuned to the application.

Suppliers of embedded processor architectures are now delivering *extensible* versions of their general-purpose processors. Examples include the ARM OptimoDE, the MIPS Pro Series and the Tensilica Xtensa. Hardware development has achieved a new level of flexibility, but sophisticated design tools are required to exploit its potential. The goal of such tools is the identification of time-critical or commonly occurring patterns of computation that could be directly implemented in custom hardware, giving both faster execution and reduced program size, because a sequence of base machine instructions is being replaced by a single custom *extension* instruction. For example, a program solving simultaneous linear equations may find it useful to have a single instruction to perform matrix inversion on a set of values held in registers.

The approach proceeds by first locating the *basic blocks* of the program, regions of sequential computation with no control transfers into them. For each basic block we construct a **data dependency graph (DDG)** which contains vertices for each base (unextended) instruction in the block, along with a vertex for each initial input datum. Figure 17.2 shows an example of a DDG. There is an arc to the vertex for the instruction u from each vertex whose instruction computes an input operand of u. DDGs are acyclic because execution within a basic block is by definition sequential.

Extension instructions are combinations of base machine instructions and are represented by sets of vertices in the DDG. Consider sections A, B and

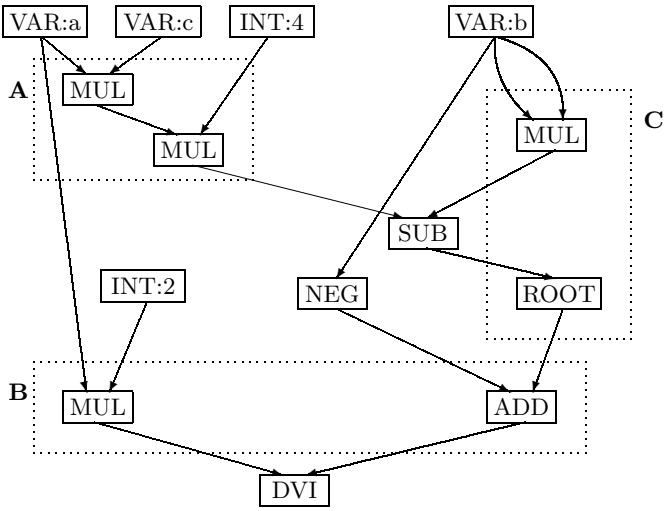


Figure 17.2 Data dependency graph for $\frac{-b+\sqrt{b^2-4ac}}{2a}$.

C in Figure 17.2 that represent candidate extension instructions. Observe that A is a cc-set, B is convex but not connected and C is neither convex nor connected. Note that every candidate extension instruction, viewed as a set of DDG vertices, must be convex since an extension instruction cannot perform computations that depend on instructions external to the extension instruction. This means that there can be no data flows out of and then back into the extension instruction: the set corresponding to an extension instruction must be convex.

While for one-processor computing only cc-sets are of interest, for multiprocessor computing all convex sets might be of interest. Ideally we would like to fully consider all possible candidate instructions and select the combination which results in the most efficient implementation. In practice this is unlikely to be feasible as, in the worst case, the number of candidates will be exponential in the number of original program instructions. However, it is useful to have a process which can find all the potential instructions, even if the set of instructions used for final consideration has to be restricted.

17.2.2 Bounds for the Number of Convex Sets

In this subsection, we consider graph-theoretical results on convex and connected convex sets which are of interest in designing and analyzing algorithms for generating convex sets. Such algorithms are topic of the next subsection.

We start by discussing lower and upper bounds on the **number cc(D) of cc-sets** in a connected acyclic digraph of order n.

The following two theorems by Gutin and Yeo give the bounds.

Theorem 17.2.1 *[481] For every connected acyclic digraph D of order n, $\mathrm{cc}(D) \geq n(n+1)/2$. If an acyclic digraph D of order n has a Hamilton path, then $\mathrm{cc}(D) = n(n+1)/2$.*

Theorem 17.2.2 *[481] Let $f(n) = 2^n + n + 1 - d_n$, where $d_n = 2 \cdot 2^{n/2}$ for every even n and $d_n = 3 \cdot 2^{(n-1)/2}$ for every odd n. For every connected acyclic digraph D of order n, $\mathrm{cc}(D) \leq f(n)$. Let $\vec{K}_{p,q}$ denote the digraph obtained from the complete bipartite graph $K_{p,q}$ by orienting every edge from the partite set of cardinality p to the partite set of cardinality q. We have $\mathrm{cc}(\vec{K}_{a,n-a}) = f(n)$ provided $|n - 2a| \leq 1$.* □

We will prove only Theorem 17.2.1 and leave the proof of Theorem 17.2.2 as an exercise. We start by showing the second part of Theorem 17.2.1. Let D_n be a connected acyclic digraph and let D_n have a Hamilton directed path $x_1x_2 \ldots x_n$. Observe that all cc-sets of D_n are of the form $\{x_i, x_{i+1}, \ldots, x_j\}$, where $i \leq j$. Thus, $\mathrm{cc}(D_n) = n(n+1)/2$. The following result implies the first part of Theorem 17.2.1.

Theorem 17.2.3 *Let H be a connected acyclic digraph of order n and let z be a vertex of H. Then $\mathrm{cc}(H) \geq n(n+1)/2$ and the number of cc-sets of H containing z, $\mathrm{cc}(H, z)$ is at least n.*

Proof: We will show the theorem by induction on n. It clearly holds for $n = 1$ so let $n > 1$. Let x be any vertex in H with $d^+_H(x) = 0$ and let $H' = H - x$. Let $R_1, R_2, \ldots, R_k$ be the connected components of H', where $k \geq 1$. Let $n_i = |V(R_i)|$ and let $H_i = H[V(R_i) \cup \{x\}]$.

First assume that $k \geq 2$. Let S_i be any cc-set in H_i which contains x. By the induction hypothesis, there are at least $n_i + 1$ such cc-sets. Note that $S_1 \cup S_2 \cup \ldots \cup S_k$ is a cc-set containing x. Since $(n_1+1)(n_2+1)\cdots(n_k+1) \geq n_1 + n_2 + \ldots + n_k + 1 = n$ we have shown that there are at least n cc-sets in H containing x.

Let $w \in V(H) \setminus \{x\}$ be arbitrary. Without loss of generality, we may assume that $w \in V(R_1)$. By the induction hypothesis, there are at least n_1 cc-sets containing w which do not contain x (all are in R_1). Note that $V(H_1) \cup S_2 \cup \ldots \cup S_k$ is a cc-set containing w and x. Since $(n_2+1)\cdots(n_k+1) \geq n_2 + \ldots + n_k + 1 = n - n_1$, we have shown that there are at least n cc-sets in H containing w.

We will now show that $\mathrm{cc}(H) \geq n(n+1)/2$. By the induction hypothesis, there are at least $n_i(n_i+1)/2$ cc-sets in R_i. Furthermore, we saw above that we have at least $(n_1+1)(n_2+1)\cdots(n_k+1)$ cc-sets in H containing x. Thus, we get the following:

$$\begin{aligned}\mathrm{cc}(H) &\geq \sum_{i=1}^{k} \frac{n_i(n_i+1)}{2} + \prod_{i=1}^{k}(n_i+1) \\ &\geq \sum_{i=1}^{k}(n_i^2+n_i)/2 \; + \sum_{1 \leq i < j \leq k} n_i n_j + \sum_{i=1}^{k} n_i + 1\end{aligned}$$

$$= \frac{1}{2}\left[\left(\sum_{i=1}^{k} n_i\right)^2 + 3\left(\sum_{i=1}^{k} n_i\right) + 2\right]$$

$$= [(n-1)^2 + 3(n-1) + 2]/2 = n(n+1)/2.$$

So now consider the case when $k = 1$.

Let $w \in V(H')$ be arbitrary. By the induction hypothesis, there are at least $n-1$ cc-sets containing w in H'. Since $V(H)$ is also a cc-set we have at least n cc-sets containing w. Let H^* be the converse of H (recall that H^* is obtained from H by reversing all arcs of H). Let $y \in V(H^*)$ be arbitrary with $d^+_{H^*}(y) = 0$ (i.e., $d^-_H(y) = 0$). By considering $H^* - y$ instead of H' we observe that there are also at least n cc-sets in H containing x (it does not matter whether $H^* - y$ is connected or not since we have already looked at the non-connected case).

Now we are able to show that $\mathrm{cc}(H) \geq n(n+1)/2$. By the induction hypothesis, there are at least $(n-1)n/2$ cc-sets in H' and they are all cc-sets in H as $d^+_H(x) = 0$. Since there are also at least n cc-sets containing x, we conclude that $\mathrm{cc}(H) \geq (n-1)n/2 + n = n(n+1)/2$. □

17.2.3 Algorithms for Generating Convex and Connected Convex Sets

Let D be a connected acyclic digraph of order n. Let $\boldsymbol{\mathcal{CO}(D)}$ ($\boldsymbol{\mathcal{CC}(D)}$) denote the family convex sets (cc-sets) of D and let $\mathbf{co(D)}$=$|\mathcal{CO}(D)|$. Gutin, Johnstone, Reddington, Scott, Soleimanfallah and Yeo [463] conjectured that the sum of the sizes of all convex sets (cc-sets) in D equals $\Theta(n \cdot \mathrm{co}(D))$ ($\Theta(n \cdot \mathrm{cc}(D))$). If the second part of the conjecture was true, then the cc-set generating algorithm $\mathcal{A}$ of complexity $O(n \cdot \mathrm{cc}(D))$ introduced in [463] (see also [63]) would be shown to be optimal with respect to its running time.

Balister, Gerke and Gutin [62] considered the following family of digraphs which refutes the conjecture. Let t be a positive integer, let $r = \lceil \sqrt{t} \rceil$, and let D_t be the digraph obtained from path $P = x_1 x_2 \dots x_{2t+3}$ by substituting vertex x_{t+1} and vertex x_{t+3} by r independent vertices (i.e., D_t is an extension of P). Let n be the order of D_t. Balister, Gerke and Gutin [62] proved that $\sum_{C \in \mathcal{CO}(D_t)} |C| = O(\sqrt{n} \cdot \mathrm{co}(D_t))$ and $\sum_{C \in \mathcal{CC}(D_t)} |C| = O(\sqrt{n} \cdot \mathrm{cc}(D_t))$.

As of the time of writing we do not know whether $\mathcal{A}$ is optimal or there is an asymptotically faster algorithm for cc-sets generation. It is not hard to modify $\mathcal{A}$ such that the new algorithm will generate all convex sets of an acyclic digraph D in time $O(n \cdot \mathrm{co}(D))$. However, a faster algorithm was designed by Balister et al. [63] and we present it here.

To obtain all convex sets of D (and $\emptyset$, which is not convex by definition), we call the following recursive procedure with the original digraph D and with $F = \emptyset$. This call yields an algorithm whose properties are studied below.

Recall that a vertex x is a **source** (**sink**) if it has no in-neighbors (out-neighbors). In general, the procedure $\mathcal{CS}$ takes as input an acyclic digraph $D = (V, A)$ and a set $F \subseteq V$ and outputs all convex sets of D which contain F. The procedure $\mathcal{CS}$ outputs V and then considers all sources and sinks of the graph that are not in F. For each such source or sink s, we call $\mathcal{CS}(D - s, F)$ and then add s to F. Thus, for each sink or source $s \in V \setminus F$ we consider all sets that contain s and all sets that do not contain s.

$\mathcal{CS}(D = (V, A), F)$
1. **output** V; set $X := V \setminus F$
2. **for all** $s \in X$ with $|N^+(s)| = 0$ or $|N^-(s)| = 0$ **do** {
3. **for all** vertices v find $N^+_{D-s}(v)$ and $N^-_{D-s}(v)$
4. call $\mathcal{CS}(D - s, F)$; set $F := F \cup \{s\}$
5. **for all** vertices v find $N^+_D(v)$ and $N^-_D(v)$ }

Correctness of the procedure. Proposition 17.2.5 and Theorem 17.2.6 imply that the procedure $\mathcal{CS}$ is correct. We first show that all sets generated in line 1 are, in fact, convex sets. To this end, we use the following lemma whose proof is left as an exercise.

Lemma 17.2.4 *Let D be an acyclic graph, let X be a convex set of D and let $s \in X$ be a source or sink of $D\langle X\rangle$. Then $X \setminus \{s\}$ is a convex set of D.* □

Now we can prove the following proposition.

Proposition 17.2.5 *Let $D = (V, A)$ be an acyclic digraph and let $F \subseteq V$. Then every set output by $\mathcal{CS}(D, F)$ is convex.*

Proof: We prove the result by induction on the number of vertices of the output set. The entire vertex set V is convex and is output by the procedure. Now assume all sets of size $n - i \geq 2$ that are output by the procedure are convex. We will show that all sets of size $n - i - 1$ that are output are also convex. When a set C is output the procedure $\mathcal{CS}(D\langle C\rangle, F')$ was called for some set $F' \subseteq V$. The only way $\mathcal{CS}(D\langle C\rangle, F')$ can be invoked is that there exist a set $C' \subset V$ and a source or sink c of $D\langle C'\rangle$ with $C = C' \setminus \{c\}$. Moreover C' will be output by the procedure and, thus, by our assumption is convex. The result now follows from Lemma 17.2.4. □

Theorem 17.2.6 *Let $D = (V, A)$ be an acyclic digraph and let $F \subseteq V$. Then every convex set of D containing F is output exactly once by $\mathcal{CS}(D, F)$.*

Proof: Let C be a convex set of D containing F. We first claim that there exist vertices $c_1, c_2, \ldots, c_t \in V$ with $V = \{c_1, c_2, \ldots, c_t\} \cup C$ and c_i is a source or sink of $D\langle C \cup \{c_i, c_{i+1}, \ldots, c_t\}\rangle$ for all $i \in [t]$. To prove the claim we will show that for every convex set H with $C \subset H \subseteq V$, there exists a source or sink $s \in H \setminus C$ of the digraph $D\langle H\rangle$. This will prove our claim as by Lemma 17.2.4 $H \setminus \{s\}$ is a convex set of D and we can repeatedly apply the claim.

If there exists no arc from a vertex of C to a vertex of $D\langle H \setminus C\rangle$, then any source of $H \setminus C$ is a source of $D\langle H\rangle$. Note that $D\langle H \setminus C\rangle$ is an acyclic digraph and, thus, has at least one source (and sink). Thus we may assume that there is an arc from a vertex u of C to a vertex v of $H \setminus C$. Consider a longest path $v = v_1v_2 \dots v_r$ in $D\langle H \setminus C\rangle$ leaving v. Observe that v_r is a sink of $D\langle H \setminus C\rangle$ and, moreover, there is no arc from v_r to any vertex of C since otherwise there would be a directed path from $u \in C$ to a vertex in C containing vertices in $H \setminus C$ which is impossible as C is convex. Hence v_r is a sink of $D\langle H\rangle$ and the claim is shown.

Next note that a sink or source remains a sink or source when vertices are deleted. Thus when $\mathcal{CS}(D, F)$ is executed and a source or sink s is considered, then we distinguish the cases when $s = c_i$ for some $i \in [t]$ or when this is not the case. If $s = c_i$ and we currently consider the digraph D' and the fixed set F', then we follow the execution path calling $\mathcal{CS}(D' - s, F')$. Otherwise we follow the execution path that adds s to the fixed set. When the last c_i is deleted, we call $\mathcal{CS}(D\langle C\rangle, F'')$ for some F'' and the set C is output. It remains to show that there is a unique execution path yielding C. To see this, note that when we consider a source or sink s, then either it is deleted or moved to the fixed set F. Thus every vertex is considered at most once and then deleted or fixed. Therefore each time we consider a source or sink there is a unique decision that finally yields C. □

Running time of $\mathcal{CS}$. We will use the following data structure for a set $Y = \{y_1, y_2, \dots, y_{|Y|}\} \subseteq \{1, 2, \dots, n\}$ that supports unit time element insertion and deletion, unit time checking whether Y is empty, and allows us to iterate over the elements of Y in $O(|Y|)$ time. We maintain arrays of integers SUCC and PRED indexed from 0 to $|Y|$ where $\text{SUCC}_k = k$ and $\text{PRED}_k = k$ if and only if $k \notin Y$. If $Y = \emptyset$, then $\text{PRED}_0 = \text{SUCC}_0 = 0$. If $Y \neq \emptyset$, then PRED_i (SUCC_i) hold y_{i-1} (y_{i+1}), where $i - 1$ and $i + 1$ are taken modulo $|Y|$, and we can iterate over the elements of V by following the chain of links from SUCC_0. Notice that SUCC_0 holds y_1 and PRED_0 holds $y_{|Y|}$.

By analogy with conventional doubly-linked list insertion and deletion, we have

insert(k)	*delete*(k)
$\text{SUCC}_k \leftarrow 0$	$\text{SUCC}_{\text{PRED}_k} \leftarrow \text{SUCC}_k$
$\text{PRED}_k \leftarrow \text{PRED}_0$	$\text{PRED}_{\text{SUCC}_k} \leftarrow \text{PRED}_k$
$\text{SUCC}_{\text{PRED}_0} \leftarrow k$	$\text{PRED}_k \leftarrow k$
$\text{PRED}_0 \leftarrow k$	$\text{SUCC}_k \leftarrow k$

We can use this data structure for sets V, X, $N_D^+(v)$, $N_D^-(v)$, $v \in V$, and F for the input acyclic digraph $D = (V, A)$ of order n. We can initialize the data structures for all these sets in time $O(n^2)$ using, say, the adjacency matrix of D. Observe that we output the vertex set of D as one convex set. Thus, it suffices to show that the running time of $\mathcal{CS}(D, F)$ without the

recursive calls is $O(|V|)$. This will yield the running time $O(\sum_{C \in \mathcal{CO}(D)} |C|)$ of $\mathcal{CS}$ by Theorem 17.2.6.

Using our data structure, we can determine *all* sources and sinks in $O(|V|)$ time. For the recursive calls of $\mathcal{CS}$ we delete one vertex and have to update the number of in-, respectively, out-neighbours of all neighbours of the deleted vertex s by iterating over V. The vertex s has at most $|V|-1$ neighbours and we can charge the cost of the updating information to the call of $\mathcal{CS}(D-s, F)$. Moreover we store the neighbours of s so that we can reintroduce them after the call of $\mathcal{CS}(D-s, F)$. Moving the sinks and sources to F needs constant time for each source or sink and thus we obtain $O(|V|)$ time in total.

In summary we initially need $O(n^2)$ time, and then each call of $\mathcal{CS}(D, F)$ is charged with $O(|V|)$ before it is called and then additionally with $O(|V|)$ time during its execution. Since we output a convex set of size $O(|V|)$, the total running time is $O(n^2) + O(\sum_{C \in \mathcal{CO}(D)} |C|)$. Since $\sum_{C \in \mathcal{CO}(D)} |C| = \Omega(n^2)$ by Theorem 17.2.1, the running time of $\mathcal{CS}$ is $O(\sum_{C \in \mathcal{CO}(D)} |C|)$.

17.3 When Greedy-Like Algorithms Fail

The Asymmetric Travelling Salesman Problem (ATSP) is the problem of finding a minimum weight Hamilton cycle in a weighted complete digraph. An instance of ATSP is depicted in Figure 17.3.

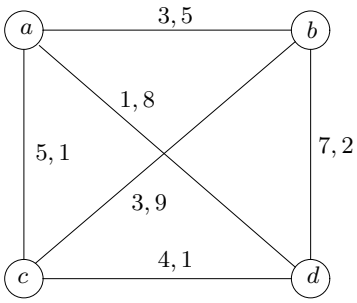


Figure 17.3 An ATSP instance on four vertices. The weights α, β of an edge xy (x is lexicographically smaller than y) mean the following: α is the weight of the arc (x, y) and β is the weight of the arc (y, x).

Greedy-like algorithms are among the simplest algorithms in combinatorial optimization. The greedy paradigm is often used in combinatorial optimization theory and practice. This phenomenon can possibly be explained by the fact that it is widely assumed that while greedy-like algorithms rarely output optimal solutions, they often provide some kind of 'approximation', i.e., they provide solutions that are significantly better than the worst ones.

This assumption seem to be justified by numerous results on 'good' behavior of greedy-like algorithms, see, e.g., [53]. However, several experimental and theoretical results question this assumption. For example, the experimental results for the ATSP presented in [571] led Johnson et al. to the conclusion that the greedy algorithm 'might be said to self-destruct' and that it should not be used even as 'a general-purpose starting tour generator'. The nearest neighbour algorithm produced mainly poor-quality Hamilton cycles as well.

In fact, as we see in this section some greedy-like algorithms may output unique worst possible solutions for many problems. There are many other combinatorial optimization heuristics that do not have this negative property; we discuss such ATSP algorithms in Section 17.4.

The main *practical message* of this section is that greedy-like algorithms should be used with great care, since for many optimization problems they may produce very poor quality solutions at least for some 'hard' families of instances. Whenever possible, more robust alternatives to simple greedy-like approaches should be considered.

In Subsection 17.3.1, we give necessary and sufficient conditions for the greedy algorithm for an optimization problem defined on a uniform independence system to produce the unique worst possible solution (we say that the algorithm **fails**). These conditions imply a threshold result for the greedy algorithm for ATSP with restricted weights to fail. A greedy-like approach, max-regret algorithms, is studied in Subsection 17.3.2. We see that max-regret algorithms also fail for ATSP despite some computational experiments that indicated that max-regret algorithms often outperform the greedy algorithm.

17.3.1 Greedy Algorithm

The theorem in [483] on the greedy algorithm for ATSP confirms the above-mentioned conclusion of Johnson et al. [571]: for every $n \geq 2$ there exist instances of ATSP with n vertices for which the greedy algorithm produces the unique worst Hamilton cycle. In this subsection based on [99] by Bang-Jensen, Gutin and Yeo we prove a stronger result: For every $n \geq 2$ there exists an instance of ATSP with weights restricted to the set $\{1, 2, \ldots, \lceil \frac{n+1}{2} \rceil\}$ for which the greedy algorithm may find the unique worst possible Hamilton cycle. Before showing this result we state a complete characterization of uniform independence systems for which the well-known greedy algorithm may produce the unique worst possible base.

The main result stated in this subsection, Theorem 17.3.2, is applicable to many combinatorial optimization problems. However, we restrict ourselves only to ATSP.

Definition 17.3.1 *Let E be a finite set and let $\mathcal{F}$ be a collection of subsets of E. The pair $(E, \mathcal{F})$ is an* **independence system** *if (I1) and (I2) below hold.*

(I1) $\emptyset \in \mathcal{F}$.
(I2) If $Y \in \mathcal{F}$ *and* $X \subseteq Y$, *then* $X \in \mathcal{F}$.

Let $(E, \mathcal{F})$ be an independence system. A set $X \subseteq E$ such that $X \in \mathcal{F}$ is called **independent**. All other sets are **dependent**. A **base** of M is a maximal independent set. An independence system is **uniform** if all its bases are of the same cardinality.

Many combinatorial optimization problems can be formulated as follows. We are given a uniform independence system $(E, \mathcal{F})$, a set $W \subseteq \mathbb{R}_0$ and a weight function w that assigns a weight $w(e) \in W$ to every element of E. The weight $w(S)$ of $S \in \mathcal{F}$ is defined as the sum of the weights of the elements of S. It is required to find a base $B \in \mathcal{F}$ of minimum weight. We will consider only such problems and call them the $(E, \mathcal{F}, W)$-*minimization problems*.

For an independent set $X \in \mathcal{F}$, let $\text{ext}(X) = \{x : \ X \cup \{x\} \in \mathcal{F}\} - X$. Note that by (I2) $\text{ext}(X) \neq \emptyset$ for every independent set X which is not a base.

The following simple algorithm $\mathcal{GA}$ is known as **the greedy algorithm for independence systems**:

Input: An independence system $(E, \mathcal{F})$ and a weight function $w : E \to \mathbb{R}_0$.
Output: a base of $(E, \mathcal{F})$.

1. Set $X := \emptyset$;
2. While $\text{ext}(X) \neq \emptyset$, choose an element $x \in \text{ext}(X)$ such that $w(x) = \min\{w(y) : y \in \text{ext}(X)\}$ and append it to X;
3. Return X.

It is well-known that for matroids $(E, \mathcal{F})$, the algorithm $\mathcal{GA}$ always solves the $(E, \mathcal{F}, \mathbb{R}_0)$-minimization problem to optimality (see Chapter 18). Unfortunately, this is far from being the case for general independence systems as we will see below.

We assume that the greedy algorithm may choose any element among equally weighted elements in $\text{ext}(X)$. Thus, when we say that $\mathcal{GA}$ **may construct** a base B, we mean that B is built provided the appropriate choices between elements of the same weight are made. An **ordered partitioning** of an ordered set $Z = \{z_1, z_2, \ldots, z_k\}$ is a collection of subsets $A_1, A_2, \ldots, A_q$ of Z satisfying that if $z_r \in A_i$ and $z_s \in A_j$ where $1 \leq i < j \leq q$, then $r < s$. Some of the sets A_i may be empty and $\cup_{i=1}^{q} A_i = Z$.

In the following theorem Bang-Jensen, Gutin and Yeo [99] characterized all independence systems $(E, \mathcal{F})$ for which there is a finite range assignment of weights to the elements of E such that $\mathcal{GA}$ for finding a base of minimum weight may construct the unique worst possible solution.

Theorem 17.3.2 *[99] Let* $(E, \mathcal{F})$ *be a uniform independence system and let* $r \geq 2$ *be a natural number. There exists a weight assignment* $w : \ E \to [r]$ *such that the greedy algorithm may produce the unique worst possible base if*

and only if $\mathcal{F}$ contains some base B with the property that for some ordering $x_1, \ldots, x_k$ of the elements of B and some ordered partitioning $A_1, A_2, \ldots, A_r$ of $x_1, \ldots, x_k$ the following holds for every base $B' \neq B$ of $\mathcal{F}$:

$$\sum_{j=0}^{r-1} |\mathrm{ext}(A_{0,j}) \cap B'| < \sum_{j=1}^{r} j \cdot |A_j|, \tag{17.1}$$

where $A_{0,j} = A_0 \cup A_1 \cup \ldots \cup A_j$ and $A_0 = \emptyset$. □

Now we are ready to consider $\mathcal{GA}$ for ATSP. Recall that ATSP is the problem of finding a minimum weight Hamilton cycle in a weighted complete digraph. Notice that ATSP can be viewed as a uniform independence system in which $E = A(\overleftrightarrow{K}_n)$ and a set of arcs is an independent set if and only if it can be completed to a Hamilton cycle. Thus, every Hamilton cycle of $\overleftrightarrow{K}_n$ is a base.

Theorem 17.3.3 *[99] Consider ATSP with weights from the set $W = [r]$. Let $n \geq 3$. For every $r \geq \lceil \frac{n+1}{2} \rceil$ there exists a weight function $w : A(\overleftrightarrow{K}_n) \to [r]$ such that the greedy algorithm may produce the unique worst possible base.*

Proof: Consider a Hamilton cycle $v_1 v_2 \ldots v_n v_1$, and let $x_i = v_i v_{i+1}$ for each $i \in [n]$ and $x_n = v_n v_1$. Fix the base $B = \{x_1, x_2, \ldots, x_n\}$ and take the ordered partition to be $A_1, A_2, \ldots, A_{\lfloor \frac{n+1}{2} \rfloor}$, where $A_i = \{x_{2i-1}, x_{2i}\}$, $i = 1, 2, \ldots, \lfloor \frac{n-1}{2} \rfloor$ and $A_{\lfloor \frac{n+1}{2} \rfloor} = \{x_{n-1}, x_n\}$ if n is even and $A_{\lfloor \frac{n+1}{2} \rfloor} = \{x_n\}$ if n is odd. Let also $A_j = \emptyset$ for each $\lfloor \frac{n+1}{2} \rfloor < j \leq r$. We will prove (17.1). By the way we have defined the ordered partitioning, it suffices to show (17.1) for $r = \lfloor \frac{n+1}{2} \rfloor$ and we assume that $r = \lfloor \frac{n+1}{2} \rfloor$.

We assume that n is even since the case of odd n can be treated similarly. Note that $r = n/2$. Let B' be a base different from B. By the choice of the ordering of B, it follows that

$$|\mathrm{ext}(A_{0,j}) \cap B| = n - 2j \text{ for } j = 0, 1, \ldots, n/2.$$

Thus,

$$\sum_{j=1}^{n/2} j \cdot |A_j| = \sum_{j=0}^{(n-2)/2} |\mathrm{ext}(A_{0,j}) \cap B|. \tag{17.2}$$

Therefore, to prove (17.1), it suffices to show that

$$|\mathrm{ext}(A_{0,j}) \cap B'| \leq |\mathrm{ext}(A_{0,j}) \cap B| \tag{17.3}$$

for each $j = 0, 1, \ldots, (n-2)/2$ and one of these inequalities is, in fact, strict (i.e., $\leq$ can be replaced by $<$). Suppose that all inequalities in (17.3) are equalities. For $j = (n-2)/2$ this means that $\{x_{n-1}, x_n\} \subset B'$. Continuing

our analysis for $j = (n-4)/2, \ldots, 1$ of arcs in B', we conclude that $B' = B$, a contradiction. □

In fact, the choice of the lower bound for r in Theorem 17.3.3 is optimal. The reader is asked to prove this in Exercise 17.1.

17.3.2 Max-Regret Algorithms

Balas and Saltzman [58] were the first to introduce a Max-Regret algorithm as an alternative to the greedy algorithm; their Max-Regret algorithm is for the 3-index assignment problem. Ghosh, Goldengorin, Gutin and Jäger [401] introduced a Max-Regret algorithm for ATSP and studied it in a series of computational experiments. The experiments indicated that the Max-Regret algorithm outperformed several other algorithms including the greedy algorithm. The authors of [401] asked what is the domination number of the Max-Regret algorithm (see Section 17.4 for a definition of domination number). Surprisingly, Gutin, Goldengorin and Huang [461] proved that both the Max-Regret algorithm and its modification are of domination number 1, i.e., they are no better than the greedy algorithm in the worst case. We give their proof in this subsection.

We start by describing the two Max-Regret algorithms. Let $V = [n]$ be the vertex set of a weighted complete digraph $\overleftrightarrow{K}_n$. The weight of an arc (i,j) is denoted by w_{ij}. Let Q be a collection of disjoint paths in $\overleftrightarrow{K}_n$. An arc ij is a **feasible addition** to Q if $Q + ij$ is either a collection of disjoint paths or a Hamilton cycle in K_n^*.

ATSP-Max-Regret-FC proceeds as follows[1]. Set $W = T = \emptyset$. While $V \neq W$ do the following: For each $i \in V \setminus W$, compute two lightest arcs (i,j) and (i,k) that are feasible additions to T, and compute the difference $\Lambda_i = |w_{ij} - w_{ik}|$. For $i \in V - W$ with maximum Λ_i choose the lightest arc (i,j), which is a feasible addition to T, and add ij to T and i to W. Return T.

ATSP-Max-Regret proceeds as follows. Set $W^+ = W^- = T = \emptyset$. While $V \neq W^+$ do the following: For each $i \in V \setminus W^+$, compute two lightest arcs (i,j) and (i,k) that are feasible additions to T, and compute the difference $\Lambda_i^+ = |w_{ij} - w_{ik}|$; for each $i \in V \setminus W^-$, compute two lightest arcs (j,i) and (k,i) that are feasible additions to T, and compute the difference $\Lambda_i^- = |w_{ji} - w_{ki}|$. Compute $i' \in V \setminus W^+$ with maximum $\Lambda_{i'}^+$ and $i'' \in V \setminus W^-$ with maximum $\Lambda_{i''}^-$. If $\Lambda_{i'}^+ \geq \Lambda_{i''}^-$ choose the lightest arc $i'j'$, which is a feasible addition to T and add (i',j') to T, i' to W^+ and j' to W^-. Otherwise, choose the lightest arc (j'',i''), which is a feasible addition to T, and add $j''i''$ to T, i'' to W^- and j'' to W^+. Return T.

[1] FC is an abbreviation for First Coordinate, which makes sense for the multi-dimensional assignment problem for which ATSP-Max-Regret-FC was initially introduced.

Remark 17.3.4 In ATSP-Max-Regret-FC, if $|V \setminus W| = 1$, we set $\Lambda_i = 0$. A similar remark applies to ATSP-Max-Regret.

Theorem 17.3.5 *For every $n \geq 2$, there exists a weight function $w : A(\overleftrightarrow{K}_n) \to \mathbb{Z}$ such that both ATSP-Max-Regret-FC and ATSP-Max-Regret produce the unique heaviest possible Hamilton cycle.*

Proof: Since the proofs for both algorithms use the same family of instances and are similar, we restrict ourselves only to ATSP-Max-Regret-FC.

Consider an instance of ATSP on the complete digraph with vertex set $\{1, 2, \ldots, n\}$, $n \geq 2$, and arc set A. Let the weights be as follows: $w_{ik} = \min\{0, i - k\}$ for each $1 \leq i \neq k \leq n$, $i \neq n$, and $w_{nk} = -k$ for each $1 \leq k \leq n-1$. We will slightly modify the weights: $w'_{ij} = w_{ij}$ unless $j = i+1$ modulo n. We set $w'_{i,i+1} = -1 - \frac{1}{n+1}$ for $1 \leq i \leq n-1$ and $w'_{n,1} = -1 - \frac{1}{n+1}$. ATSP-Max-Regret-FC will use the weight function w'.

ATSP-Max-Regret-FC constructs the Hamilton cycle $T_{MR} = 123 \ldots n1$ by first choosing the arc $(n-1, n)$, then the arc $(n-2, n-1)$, etc. The last two arcs are $(1, 2)$ and $(n, 1)$ (they must be included in the Hamilton cycle). Indeed, initially $\Lambda_{n-1} = \frac{n+2}{n+1} > \Lambda_i$ for each $i \neq n-1$. Once $(n-1, n)$ is added to T_{MR}, $\Lambda_{n-2} = \frac{n+2}{n+1}$ becomes maximal, etc.

Let T', T'' be a pair of Hamilton cycles. Since $\sum_{ij \in A} |w_{ij} - w'_{ij}| < 1$, $w(T') < w(T'')$ implies $w'(T') < w'(T'')$. Thus, to prove that $w'(T) < w'(T_{MR})$ for each Hamilton cycle $T \neq T_{MR}$, it suffices to show that $w(T) < w(T_{MR})$.

Observe that $w(T_{MR}) = -n$. Let $T = i_1 i_2 \ldots i_n i_1$ be an arbitrary Hamilton cycle, where $i_1 = 1$. Suppose that $i_s = n$. Observe that the weight of the path $P = i_1 i_2 \ldots i_s$ equals $\sum_{k=1}^{s-1} \min\{0, i_k - i_{k+1}\}$. Thus, $w(P) \leq 1 - n$ and $w(P) = 1 - n$ if and only if $i_1 < i_2 < \ldots < i_s$. Since $i_s = n$, the weight of the arc (i_s, i_{s+1}) equals $-i_{s+1}$. Thus, $w(T) \leq 1 - n - i_{s+1}$ and $w(T) \geq w(T_{MR})$ if and only if $i_{s+1} = 1$ and $i_1 < i_2 < \ldots < i_s$. We conclude that $w(T) \geq w(T_{MR})$ if and only if $T = T_{MR}$. □

17.4 Domination Analysis of ATSP Heuristics

Recall that in the Asymmetric Traveling Salesman Problem (ATSP) we are required to find a minimum weight Hamilton cycle in a weighted complete digraph. A **heuristic** for an optimization problem $\mathcal{R}$ is an algorithm which given an instance R of $\mathcal{R}$ finds some solution s to R for which there is generally no guarantee on the quality of s compared to an optimal solution s^* to R. So for the ATSP a heuristic is any algorithm which returns some Hamilton cycle of the input complete digraph $\overleftrightarrow{K}_n$.

Research on combinatorial optimization (CO) heuristics has produced a large variety of heuristics especially for well-known CO problems and, thus,

it is important to develop ways of selecting the best ones among them. In most of the literature, heuristics are compared by means of computational experiments and, while experimental analysis is of definite importance, it cannot cover all possible families of instances of the CO problem at hand and, in particular, it normally does not cover the hardest instances. Worst-case analysis is often performed by approximation analysis [53], where upper or lower bounds for the worst-case performance ratio are of interest. Introduced by Glover and Punnen [409], domination analysis provides an alternative and a complement to approximation analysis. In domination analysis, we are interested in the domination number or domination ratio of the heuristic solution. We define these parameters below.

Pros and cons of domination analysis are discussed in [480] and, in our view, it is advantageous to have bounds for both performance ratio and domination ratio of a heuristic whenever it is possible. Roughly speaking this would enable us to see a 2D picture rather than a 1D picture.

Let $\mathcal{P}$ be a minimization CO problem, let $\mathcal{I}$ be an instance of $\mathcal{P}$, let $S(\mathcal{I})$ denote the set of feasible solutions of $\mathcal{I}$ and let H be a heuristic for $\mathcal{P}$. The size of $\mathcal{I}$ is denoted by $|\mathcal{I}|$ and the solution obtained by H for $\mathcal{I}$ is denoted by $H(\mathcal{I})$. When considering the weight of a solution y we write $w(y)$.

The *domination number* of a heuristic H is

$$\mathrm{domn}(H,n) = \min_{\mathcal{I}\in\mathcal{P}:\ |\mathcal{I}|=n} \mathrm{domn}(H,\mathcal{I}),$$

where $\mathrm{domn}(H,\mathcal{I}) = |\{y \in S(\mathcal{I}) :\ w(H(\mathcal{I})) \le w(y)\}|$. In other words, the domination number $\mathrm{domn}(H,n)$ is the maximum integer such that the solution $H(\mathcal{I})$ obtained by H for *any* instance $\mathcal{I}$ of $\mathcal{P}$ of size n is not worse than at least $\mathrm{domn}(H,n)$ feasible solutions of $\mathcal{I}$ (including $H(\mathcal{I})$). The *domination ratio* of H is

$$\mathrm{domr}(H,n) = \min_{\mathcal{I}\in\mathcal{P}:\ |\mathcal{I}|=n} \frac{\mathrm{domn}(H,\mathcal{I})}{|S(\mathcal{I})|}.$$

Clearly, the domination ratio is well defined for every heuristic and, for the same optimization problem, a heuristic with higher domination ratio may be considered a better choice than a heuristic with lower domination ratio. Ben-Arieh et al. [135] compared two heuristics for the Generalized ATSP (given a weighted complete k-partite digraph, find a minimum weight cycle containing exactly one vertex from each partite set). The heuristics performed equally well in computational experiments, but it was proved that one of them has a significantly larger domination number. For the Symmetric TSP, Punnen, Margot and Kabadi [757] showed that after a polynomial number of iterations the domination number of the best improvement 2-Opt that uses small neighbourhoods significantly exceeds that of the best improvement local search based on neighbourhoods of much larger cardinality. Punnen, Margot and Kabadi [757] and other papers have led Gutin and Yeo [480] to the conclusion that the cardinality of the neighbourhood used by a local search is

not the right measure of the effectiveness of the local search. Domination ratio, along with some other parameters such as the diameter of the neighbourhood digraph (see Gutin, Yeo and Zverovitch [484]), provides a much better measure.

There are many papers on domination analysis of various CO problems, see, e.g., [25] by Alon, Gutin and Krivelevich, [139] by Berend, Skiena and Twitto, [461] by Gutin, Goldengorin and Huang, [462] by Gutin, Jensen and Yeo, [617] by Koller and Noble and [757] by Punnen, Margot and Kabadi. For surveys on the topic, see [484] and [480].

In Section 17.3, we proved that several ATSP heuristics including the greedy algorithm are of domination number 1. Clearly, every exact ATSP algorithm has domination number $(n-1)!$. Thus, the domination number of an algorithm close to $(n-1)!$ may be taken as an indication that the algorithm is of high quality.

In Subsection 17.4.1, we will see that there are ATSP heuristics of domination number at least $(n-2)!$ for each $n \geq 2,\ n \neq 6$. This follows from the fact that if a heuristic $\mathcal{H}$ always produces a Hamilton cycle with weight not worse than the average weight of a Hamilton cycle, then $\mathcal{H}$ is of domination number at least $(n-2)!$ for each $n \geq 2,\ n \neq 6$. In Subsection 17.4.2, we will consider upper bounds on the domination number of polynomial-time ATSP heuristics.

17.4.1 ATSP Heuristics with Factorial Domination Numbers

Let $(\overleftrightarrow{K}_n, w)$ denote a complete digraph on n vertices whose arcs are weighted according to a weight function w. The total cost of all Hamilton cycles in $(\overleftrightarrow{K}_n, w)$ is denoted by $\sigma(n,w)$. Denote the sum of the costs of all arcs in $(\overleftrightarrow{K}_n, w)$ by $w(\overleftrightarrow{K}_n)$. The **average cost of a Hamilton cycle** in $(\overleftrightarrow{K}_n, w)$ is denoted by $\tau(n,w)$. As every arc of $\overleftrightarrow{K}_n$ is contained in $(n-2)!$ Hamilton cycles, $\tau(n,w) = \sigma(n,w)/(n-1)! = (n-2)!w(\overleftrightarrow{K}_n)/(n-1)!$, hence, $\tau(n,w) = w(\overleftrightarrow{K}_n)/(n-1)$. This formula can also be shown using linearity of expectation (see [28]). An **automorphism** of a digraph D is a bijection $\phi : V(D) \to V(D)$ such that $xy \in A(D)$ if and only if $\phi(x)\phi(y) \in A(D)$.

The following result was first obtained by Sarvanov [794] when n is odd and Gutin and Yeo [479] when n is even.

Theorem 17.4.1 *Let H be a Hamilton cycle in $\overleftrightarrow{K}_n$ such that $w(H) \leq \tau(n,w)$. If $n \neq 6$, then there are at least $(n-2)!$ Hamilton cycles in $\overleftrightarrow{K}_n$ whose cost is at least $w(H)$.*

Proof: The result is trivial for $n = 2, 3$. If $n = 4$, the result follows from the obvious fact that the most expensive Hamilton cycle T in $\overleftrightarrow{K}_n$ has cost $w(T) \geq w(H)$.

Assume that $n \geq 5$ and $n \neq 6$. Let $D_1 = \{C_1, C_2, \ldots, C_{n-1}\}$ be a decomposition of the arcs of $\overleftrightarrow{K}_n$ into Hamilton cycles (such a decomposition exists by Theorem 13.4.3). Given a Hamilton cycle T in $\overleftrightarrow{K}_n$, clearly there is an automorphism of $\overleftrightarrow{K}_n$ that maps C_1 into T. Therefore, if we consider D_1 together with the decompositions $(D_1, D_2, \ldots, D_{(n-1)!})$ of $\overleftrightarrow{K}_n$ obtained from D_1 using all automorphisms of $\overleftrightarrow{K}_n$ which map the vertex 1 into itself, we will have every Hamilton cycle of $\overleftrightarrow{K}_n$ in one of the D_i's. Moreover, every Hamilton cycle is in exactly $n-1$ of the decompositions $D_1, D_2, \ldots, D_{(n-1)!}$ (by mapping a Hamilton cycle C_i into a Hamilton cycle C_j $(i, j \in \{1, 2, \ldots, n-1\})$ we fix the automorphism).

Choose the most expensive Hamilton cycle in each of D_i and form a set $\mathcal{E}$ from all distinct Hamilton cycles obtained in this manner. Clearly, $|\mathcal{E}| \geq (n-2)!$. Since $\sum_{i=1}^{n-1} w(C_i) = w(\overleftrightarrow{K}_n)$, every Hamilton cycle T of $\mathcal{E}$ has weight $w(T) \geq \tau(n, w)$. Therefore, $w(H) \leq w(T)$ for every $T \in \mathcal{E}$. □

Vertex insertion algorithms for the ATSP work as follows. First, we find some ordering $v_1, v_2, \ldots, v_n$ of vertices of $(\overleftrightarrow{K}_n, w)$. Then, we perform $n-1$ steps. On the first step we form the cycle $v_1 v_2 v_1$. On step k, $2 \leq k \leq n-1$, given the k-cycle $v_{\pi(1)} v_{\pi(2)} \ldots v_{\pi(k)} v_{\pi(1)}$ from the previous step, we find j_0, which minimizes the expression

$$w(v_{\pi(j)} v_{k+1}) + w(v_{k+1} v_{\pi(j+1)}) - w(v_{\pi(j)} v_{\pi(j+1)}),$$

$1 \leq j \leq k$, and insert v_{k+1} between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ forming a $(k+1)$-cycle.

The fastest such algorithm is the random insertion algorithm, in which the initial vertex ordering is random (see the paper [407] by Glover, Gutin, Yeo and Zverovich for computational experiments with this and other heuristics for ATSP).

The following theorem was proved by Lifshitz for the Symmetric TSP (see [790]) and, independently, by Punnen and Kabadi [756] for ATSP.

Theorem 17.4.2 *Let H_n be a Hamilton cycle constructed by a vertex insertion algorithm $\mathcal{A}$ for ATSP on $(\overleftrightarrow{K}_n, w)$. Then $w(H_n) \leq \tau(n, w)$.*

Proof: We prove this result by induction on n. The theorem is trivially true for $n = 2$. Let $H_{n-1} = v_{\pi(1)} v_{\pi(2)} \ldots v_{\pi(n-1)} v_{\pi(1)}$ be the cycle constructed in Step $n-2$ of the algorithm and assume that in Step $n-1$, it was decided to insert v_n between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ in order to obtain H_n. Then, we have

$$\begin{aligned} w(H_n) &= w(H_{n-1}) + w(v_{\pi(j_0)} v_n) + w(v_n v_{\pi(j_0+1)}) - w(v_{\pi(j_0)} v_{\pi(j_0+1)}) \\ &\leq w(H_{n-1}) + \frac{\sum_{i=1}^{n-1} [w(v_{\pi(i)} v_n) + w(v_n v_{\pi(i+1)}) - c(v_{\pi(i)} v_{\pi(i+1)})]}{n-1} \\ &= w(H_{n-1}) + \frac{w(V - v_n, v_n) + w(v_n, V - v_n) - w(H_{n-1})}{n-1} \end{aligned}$$

$$\leq \frac{(n-2)\tau(n-1,w) + w(V-v_n,v_n) + w(v_n,V-v_n)}{n-1}$$
$$= \frac{w(\overleftrightarrow{K}_n - v_n) + w(V-v_n,v_n) + w(v_n,V-v_n)}{n-1}$$
$$= \frac{w(\overleftrightarrow{K}_n)}{n-1} = \tau(n,w),$$

where $\tau(n-1,w)$ is the average cost of a Hamilton cycle in $\overleftrightarrow{K}_n - v_n$. □

Theorems 17.4.1 and 17.4.2 imply the following result:

Theorem 17.4.3 *For every vertex insertion algorithm $\mathcal{A}$ we have* $\mathrm{domn}(\mathcal{A},n) \geq (n-2)!$. □

Results similar to Theorem 17.4.3 have been obtained for other ATSP heuristics (such as k-Opt, $k \geq 3$), see, e.g., [479, 757].

17.4.2 Upper Bounds on Domination Numbers

It is realistic to assume that any ATSP algorithm spends at least one unit of time on every arc of the weighted complete digraph $\overleftrightarrow{K}_n$ that it considers. We use this assumption in the rest of this subsection. Let P be an (x,y)-path in $\overleftrightarrow{K}_n$. The **path-contraction** of P is an operation of replacing $\overleftrightarrow{K}_n$ by $\overleftrightarrow{K}_{n-p}$, where p is the length of P, such that all vertices of P are replaced by a single new vertex z, the weight of all arcs uv with no vertex from P are unchanged and the weight of every arc of the form uz (zv) equals the weight of the arc ux (yv). The following theorem was obtained by Gutin, Koller and Yeo [469].

Theorem 17.4.4 *Let $\mathcal{A}$ be an ATSP heuristic that runs in time at most $t(n)$. Then the domination number of $\mathcal{A}$ does not exceed* $\max_{1\leq n'\leq n}(t(n)/n')^{n'}$.

Proof: Let $D = (\overleftrightarrow{K}_n, w)$ be an instance of ATSP and let H be the Hamilton cycle that $\mathcal{A}$ returns, when its input is D. Let $DOM(H)$ denote all Hamilton cycles in D which are not lighter than H including H itself. We assume that D is a worst instance for $\mathcal{A}$, namely, $\mathrm{domn}(\mathcal{A},n) = |DOM(H)|$. Since $\mathcal{A}$ is arbitrary, to prove this theorem, it suffices to show that $|DOM(H)| \leq \max_{1\leq n'\leq n}(t(n)/n')^{n'}$.

Let E denote the set of arcs in D, which $\mathcal{A}$ actually examines; observe that $|E| \leq t(n)$ by the assumption above. Let $A(H)$ be the set of arcs in H. Let F be the set of arcs in H that are not examined by $\mathcal{A}$, and let G denote the set of arcs in $D - A(H)$ that are not examined by $\mathcal{A}$.

We first prove that every arc in F must belong to each Hamilton cycle of $DOM(H)$. Assume that there is a Hamilton cycle $H' \in DOM(H)$ that avoids an arc $a \in F$. If we assign to a a very large weight, H' becomes lighter than H, a contradiction.

Similarly, we prove that no arc in G can belong to a Hamilton cycle in $DOM(H)$. Assume that $a \in G$ and a is in a Hamilton cycle $H' \in DOM(H)$. By making a very light (possibly negative), we can ensure that $w(H') < w(H)$, a contradiction.

Now let D' be the digraph obtained by path-contracting the paths comprising F and deleting the arcs in G, and let n' be the number of vertices in D'. Note that every Hamilton cycle in $DOM(H)$ corresponds to a Hamilton cycle in D' and, thus, the number of Hamilton cycles in D' is an upper bound on $|DOM(H)|$. In a Hamilton cycle of D', there are at most $d^+(i)$ possibilities for the successor of a vertex i, where $d^+(i)$ is the out-degree of i in D'. Hence we obtain that

$$|DOM(H)| \leq \prod_{i=1}^{n'} d^+(i) \leq \left(\frac{1}{n'} \sum_{i=1}^{n'} d^+(i) \right)^{n'} \leq \left(\frac{t(n)}{n'} \right)^{n'},$$

where we applied the arithmetic-geometric mean inequality. □

Corollary 17.4.5 *[469] Let $\mathcal{A}$ be an ATSP heuristic that runs in time at most $t(n)$. Then the domination number of $\mathcal{A}$ does not exceed*

$$\max\{e^{t(n)/e}, (t(n)/n)^n\},$$

where e is the basis of natural logarithms.

Proof: Let $U(n) = \max_{1 \leq n' \leq n} (t(n)/n')^{n'}$. By differentiating the function $f(n') = (t(n)/n')^{n'}$ with respect to n' we can readily obtain that $f(n')$ increases for $1 \leq n' \leq t(n)/e$, and decreases for $t(n)/e \leq n' \leq n$. Thus, if $n \leq t(n)/e$, then $f(n')$ increases for every value of $n' < n$ and $U(n) = f(n) = (t(n)/n)^n$. On the other hand, if $n \geq t(n)/e$, then the maximum of $f(n')$ is for $n' = t(n)/e$ and, hence, $U(n) = e^{t(n)/e}$. □

The next assertion follows directly from the proof of Corollary 17.4.5.

Corollary 17.4.6 *[469] Let $\mathcal{A}$ be an ATSP heuristic that runs in time at most $t(n)$. For $t(n) \geq en$, the domination number of $\mathcal{A}$ does not exceed $(t(n)/n)^n$.* □

Note that the restriction $t(n) \geq en$ is important since otherwise the bound of Corollary 17.4.6 can be invalid. Indeed, if $t(n)$ is a constant, then for n large enough the upper bound becomes smaller than 1, which is not correct since the domination number is always at least 1.

It is proved by Gutin [458] that there are $O(n)$-time ATSP algorithms of domination number $2^{\Theta(n)}$. It follows from the last corollary that this result cannot be improved.

The following result gives a better bound than Theorem 17.4.4 when we do not restrict the running time of a polynomial-time ATSP heuristic.

Theorem 17.4.7 *[469] Unless P=NP, there is no polynomial time ATSP algorithm of domination number at least* $(n-1)! - \lfloor n - n^{\alpha} \rfloor!$ *for any constant* $\alpha < 1$.

Proof: Assume that there is a polynomial time algorithm $\mathcal{H}$ with domination number at least $(n-1)! - \lfloor n - n^{\alpha} \rfloor!$ for some constant $\alpha < 1$. Choose an integer $s > 1$ such that $\frac{1}{s} < \alpha$.

Consider a weighted complete digraph $(\overleftrightarrow{K}_n, w)$. We may assume that all weights are non-negative as otherwise we may add a large number to each weight. Choose any pair of distinct vertices u and v in $\overleftrightarrow{K}_n$. Consider another complete digraph D on $n^s - n$ vertices, in which all weights are 0 and which is vertex disjoint from $\overleftrightarrow{K}_n$. Add all possible arcs between $\overleftrightarrow{K}_n$ and D such that the weights of all arcs coming into u and going out of v are 0 and the weights of all other arcs are M, where M is larger than n times the maximum weight in $(\overleftrightarrow{K}_n, w)$. Let the resulting weighted complete digraph be denoted by $\overleftrightarrow{K}_{n^s}$ and note that we have now obtained an instance $(\overleftrightarrow{K}_{n^s}, w')$ of ATSP.

Apply $\mathcal{H}$ to $(\overleftrightarrow{K}_{n^s}, w')$ (observe that $\mathcal{H}$ is polynomial in n for $(\overleftrightarrow{K}_{n^s}, w')$). Notice that there are exactly $(n^s - n)!$ Hamilton cycles in $(\overleftrightarrow{K}_{n^s}, w')$ of weight L, where L is the weight of a lightest Hamilton (u, v)-path in $\overleftrightarrow{K}_n$. Each of the $(n^s - n)!$ Hamilton cycles is obviously optimal. Observe that the domination number of $\mathcal{H}$ on $\overleftrightarrow{K}_{n^s}$ is at least $(n^s - 1)! - \lfloor n^s - (n^s)^{\alpha} \rfloor!$. However, for sufficiently large n, we have

$$(n^s - 1)! - \lfloor n^s - (n^s)^{\alpha} \rfloor! \geq (n^s - 1)! - (n^s - n)! + 1$$

as $n^{s\alpha} \geq n + 1$ for n large enough. Thus, a Hamilton cycle produced by $\mathcal{H}$ is always among the optimal solutions (for n large enough). This means that we can obtain a lightest Hamilton (u, v)-path in $\overleftrightarrow{K}_n$ in polynomial time, which is impossible since the lightest Hamilton (u, v)-path problem is NP-hard. We have arrived at a contradiction. □

17.5 Solving the 2-Satisfiability Problem

In this section we deal with a problem that is not a problem on digraphs, but it has applications to several problems on graphs, in particular when we want to decide whether a given undirected graph has an orientation with certain properties or whether there is a homomorphism from a digraph to a digraph, see, e.g., Chapter 11 and the paper [309] by Feder. We will show how to solve this problem efficiently using the algorithm for strong components of digraphs from Chapter 2.

A **boolean variable** x is a variable that can assume only two values 0 and 1. The **sum** of boolean variables $x_1 + x_2 + \ldots + x_k$ is defined to be 1 if

at least one of the x_i's is 1 and 0 otherwise. The **negation** $\overline{x}$ of a boolean variable x is the variable that assumes the value $1 - x$. Hence $\overline{\overline{x}} = x$. Let X be a set of boolean variables. For every $x \in X$ there are two **literals**, over x, namely, x itself and $\overline{x}$. A **clause** C over a set of boolean variables X is a sum of literals over the variables from X. The **size** of a clause is the number of literals it contains. For example, if u, v, w are boolean variables with values $u = 0, v = 0$ and $w = 1$, then $C = (u + \overline{v} + \overline{w})$ is a clause of size 3, its value is 1 and the literals in C are u, $\overline{v}$ and $\overline{w}$. An assignment of values to the set of variables X of a boolean expression is called a **truth assignment**. If the variables are $x_1, \ldots, x_k$, then we denote a truth assignment by $t = (t_1, \ldots, t_k)$. Here it is understood that x_i will be assigned the value t_i for $i = 1, \ldots, k$.

The **2-satisfiability problem**, also called **2-SAT**, is the following problem. Let $X = \{x_1, \ldots, x_k\}$ be a set of boolean variables and let $C_1, \ldots, C_r$ be a collection of clauses, all of size 2, for which every literal is over X. Decide if there exists a truth assignment $t = (t_1, \ldots, t_k)$ to the variables in X such that the value of every clause will be 1. This is equivalent to asking whether or not the boolean expression $\mathcal{F} = C_1 * \ldots * C_p$ can take the value 1. Depending on whether this is possible or not, we say that $\mathcal{F}$ is **satisfiable** or **unsatisfiable**. Here '$*$' stands for **boolean multiplication**, that is, $1 * 1 = 1$, $1 * 0 = 0 * 1 = 0 * 0 = 0$. For a given truth assignment $t = (t_1, \ldots, t_k)$ and literal q we denote by $q(t)$ the value of q when we use the truth assignment t (i.e., if $q = \overline{x_3}$ and $t_3 = 1$, then $q(t) = 1 - 1 = 0$).

To illustrate the definitions, let $X = \{x_1, x_2, x_3\}$ and let $C_1 = (\overline{x_1} + \overline{x_3})$, $C_2 = (x_2 + \overline{x_3})$, $C_3 = (\overline{x_1} + x_3)$ and $C_4 = (x_2 + x_3)$. Then it is not difficult to check that $\mathcal{F} = C_1 * C_2 * C_3 * C_4$ is satisfiable and that taking $x_1 = 0, x_2 = 1, x_3 = 1$ we obtain $\mathcal{F} = 1$.

If we allow more than 2 literals per clause, then we obtain the more general problem **Satisfiability** (also called **SAT**) which is $\mathcal{NP}$-complete, even if all clauses have size 3, in which case it is also called **3-SAT** (see e.g. page 359 in the book [742] by Papadimitriou and Steiglitz). (In his proof of the existence of an $\mathcal{NP}$-complete problem, Cook used the satisfiability problem and showed how every other problem in $\mathcal{NP}$ can be reduced to this problem.) Below we will show how to reduce 2-SAT to the problem of finding the strong components in a certain digraph. We shall also show how to find a satisfying truth assignment if one exists. This step is important in applications, such as those in Chapter 11.

Let $C_1, \ldots, C_r$ be clauses of size 2 such that the literals are taken among the variables $x_1, \ldots, x_k$ and their negations and let $\mathcal{F} = C_1 * \ldots * C_r$ be an instance of 2-SAT. Construct a digraph $D_{\mathcal{F}}$ as follows. Let $V(D_{\mathcal{F}}) = \{x_1, \ldots, x_k, \overline{x_1}, \ldots, \overline{x_k}\}$ (i.e., $D_{\mathcal{F}}$ has two vertices for each variable, one for the variable and one for its negation). For every choice of $p, q \in V(D_{\mathcal{F}})$ such that some C_i has the form $C_i = (p + q)$, $A(D_{\mathcal{F}})$ contains an arc from $\overline{p}$ to q and an arc from $\overline{q}$ to p (recall that $\overline{\overline{x}} = x$). See Figure 17.4 for examples

of 2-SAT expressions and the corresponding digraphs. The first expression is satisfiable, the second is not.

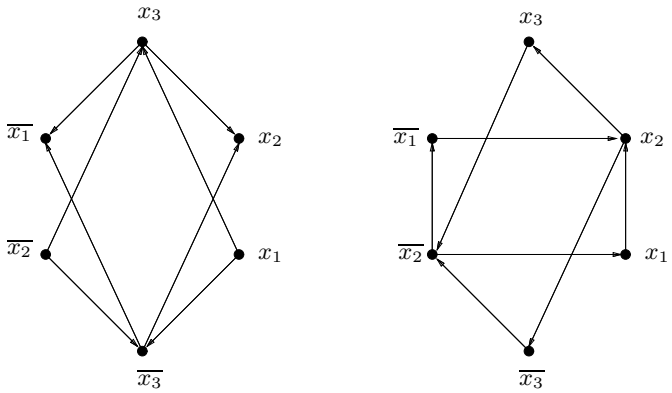


Figure 17.4 The digraph $D_{\mathcal{F}}$ is shown for two instances of 2-SAT. In (a) $\mathcal{F} = (\overline{x_1} + \overline{x_3}) * (x_2 + \overline{x_3}) * (\overline{x_1} + x_3) * (x_2 + x_3)$ and in (b) $\mathcal{F} = (x_1 + x_2) * (\overline{x_1} + x_2) * (\overline{x_2} + x_3) * (\overline{x_2} + \overline{x_3})$.

Lemma 17.5.1 *If $D_{\mathcal{F}}$ has a (p,q)-path, then it also has a $(\overline{q},\overline{p})$-path. In particular, if p, q belong to the same strong component in $D_{\mathcal{F}}$, then $\overline{p}, \overline{q}$ belong to the same strong component in $D_{\mathcal{F}}$.*

Proof: This follows easily by induction on the length of a shortest (p,q)-path, using the fact that $(x,y) \in A(D_{\mathcal{F}})$ if and only if $(\overline{y},\overline{x}) \in A(D_{\mathcal{F}})$. □

Lemma 17.5.2 *If $D_{\mathcal{F}}$ contains a path from p to q, then, for every satisfying truth assignment t, $p(t) = 1$ implies $q(t) = 1$.*

Proof: Observe that $\mathcal{F}$ contains a clause of the form $(\overline{a} + b)$ and every clause takes the value 1 under any satisfying truth assignment. Thus, by the fact that t is a satisfying truth assignment and by the definition of $D_{\mathcal{F}}$, we have that for every arc $(a,b) \in A(D_{\mathcal{F}})$, $a(t) = 1$ implies $b(t) = 1$. Now the claim follows easily by induction on the length of the shortest (p,q)-path in $D_{\mathcal{F}}$. □

The following is an easy corollary of Lemmas 17.5.1 and 17.5.2.

Corollary 17.5.3 *If t is a satisfying truth assignment, then for every strong component D' of $D_{\mathcal{F}}$ and every choice of distinct vertices $p, q \in V(D')$ we have $p(t) = q(t)$. Furthermore we also have $\overline{p}(t) = \overline{q}(t)$.* □

Lemma 17.5.4 *$\mathcal{F}$ is satisfiable if and only if for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$.*

Proof: Suppose t is a satisfying truth assignment for $\mathcal{F}$ and that there is some variable x_i such that x_i and $\overline{x_i}$ are in the same strong component in $D_{\mathcal{F}}$. Without loss of generality $x_i(t) = 1$ and now it follows from Lemma 17.5.2 and the fact that $D_{\mathcal{F}}$ contains a path from x_i to $\overline{x_i}$ that we also have $\overline{x_i}(t) = 1$ which is impossible. Hence if $\mathcal{F}$ is satisfiable, then for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$.

Now suppose that for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$. We will show that $\mathcal{F}$ is satisfiable by constructing a satisfying truth assignment for $\mathcal{F}$.

Let $D_1, \ldots, D_s$ denote some acyclic ordering of the strong components of $D_{\mathcal{F}}$ (i.e., there is no arc from D_j to D_i if $i < j$). Recall that by Proposition 2.1.3, such an ordering exists. We claim that the following algorithm will determine a satisfying truth assignment for $\mathcal{F}$: first mark all vertices 'unassigned' (meaning truth value still pending). Then going backwards starting from D_s and ending with D_1 we proceed as follows. If there is any vertex $v \in V(D_i)$ such that $\overline{v}$ has already been assigned a value, then assign all vertices in D_i the value 0 and otherwise assign all vertices in D_i the value 1. Observe that this means that, for every variable x_i, we will always assign the value 1 to whichever of $x_i, \overline{x_i}$ belongs to the strong component with the highest index. To see this, let p denote whichever of $x_i, \overline{x_i}$ belongs to the strong component of highest index j. Let $i < j$ be chosen such that $\overline{p} \in D_i$. Suppose we assign the value 0 to p. This means that at the time we considered p, there was some $q \in D_j$ such that $\overline{q} \in D_f$ for some $f > j$. But then $\overline{p} \in D_f$, by Lemma 17.5.1, contradicting the fact that $i < f$.

Clearly all vertices in $V(\mathcal{F})$ will be assigned a value, and by our previous argument this is consistent with a truth assignment for the variables of $\mathcal{F}$. Hence it suffices to prove that each clause has value 1 under the assignment. Suppose some clause $C_f = (p + q)$ attains the value 0 under our assignment. By our observation above, the reason we did not assign the value 1 to p was that at the time we considered p we had already given the value 1 to $\overline{p}$ and $\overline{p}$ belonged to a component D_j with a higher index than the component D_i containing p. Thus the existence of the arc $(\overline{p}, q) \in A(D_{\mathcal{F}})$ implies that $q \in D_h$ for some $h \geq j$. Applying the analogous argument to q we conclude that $\overline{q}$ is in some component D_g with $g > h$. But then, using the existence of the arc $(\overline{q}, p)$, we get that $i \geq g > h \geq j > i$, a contradiction. This shows that we have indeed found a correct truth assignment for $\mathcal{F}$ and hence the proof is complete. □

In Chapter 2 we will see that for any digraph D one can find the strong components of D and an acyclic ordering of these in $O(n+m)$ time. Since the number of arcs in $D_{\mathcal{F}}$ is twice the number of clauses in $D_{\mathcal{F}}$ and the number of vertices in $D_{\mathcal{F}}$ is twice the number of variables in $D_{\mathcal{F}}$, it is not difficult to see that the algorithm outlined above can be performed in time $O(k + r)$ and hence we have the following result.

Theorem 17.5.5 *The problem 2-SAT is solvable in linear time with respect to the number of clauses.* □

The approach we adopted is outlined briefly in Exercise 15.6 of the book [742] by Papadimitriou and Steiglitz, see also the paper [307] by Even, Itai and Shamir.

It is interesting to note that if, instead of asking whether $\mathcal{F}$ is satisfiable, we ask whether there exists some truth assignment such that at least ℓ clauses will get the value 1, then this problem, which is called **MAX-2-SAT**, is $\mathcal{NP}$-complete as shown by Garey, Johnson and Stockmeyer [394] (here ℓ is part of the input for the problem).

17.6 Alternating Hamilton Cycles in Genetics

In [270, 271] Dorninger considers Bennett's model (see Bennett's book [138] and the papers [519, 520] by Heslop-Harrison and Bennett) of chromosome arrangement in a cell of a eukaryotic organism. In [271], the case of even number, n, of chromosomes is studied. We consider here only this case as it is more interesting. Every individual chromosome consists of a long arm and a short arm, which are linked at the so-called centromere. At a certain stage of cell division, which is of interest to biologists, the arms of n chromosomes form an n-angle star whose internal points are the centromeres (see Figure 17.5) and external points created by the arms of 'adjacent' chromosomes. To

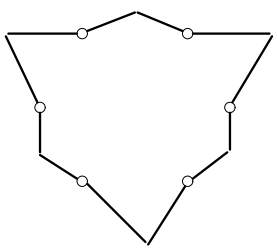


Figure 17.5 Chromosome arrangement.

find the order of the centromeres, Bennett [138] suggested that the external points are formed by the most similar size arms. Bennett and Dorninger (see [271]) generalized the notion of similarity to so-called k-similarity and Dorninger [271] analyzed the consistency of this generalized notion. Let us consider the following graph-theoretic model of this biological system. Let s_i and l_i denote the short and long arm of chromosome number i. Let the chromosomes be labelled 1,2,...,n in such a way that s_i is longer than s_j if

$i < j$, and let π be a permutation of $1, 2, \ldots, n$ such that $l_{\pi(i)}$ is longer than $l_{\pi(j)}$ if $i < j$.

We call two short arms s_i and s_j (long arms $l_{\pi(i)}$ and $l_{\pi(j)}$), $i \neq j$, ***k*-similar** if $|i - j| \leq k$. In this way, for $k = 1$, we obtain the original Bennett's notion of 'most similar size'. Let $G(n, k, \pi)$ be a 2-edge-coloured multigraph with vertex set $\{1, 2, \ldots, n\}$. The blue (red) subgraph $G_1(n, k, \pi)$ ($G_2(n, k, \pi)$) of $G(n, k, \pi)$ consists of edges pq ($p \neq q$) such that s_p and s_q (l_p and l_q) are k-similar. (See Figure 17.6.)

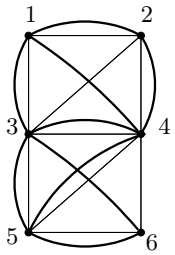


Figure 17.6 The 2-edge-coloured graph $G(6, 2, \tau)$, where $\tau(1) = 2$, $\tau(2) = 1$, $\tau(3) = 4$, $\tau(4) = 3$, $\tau(5) = 5$, $\tau(6) = 6$. The blue edges are shown by ordinary lines. The red edges are indicated by fat lines.

According to one of Bennett's assumptions, $G(n, k, \pi)$ has an alternating Hamilton cycle. Dorninger [271] analyzed when $G(n, k, \pi)$ has an alternating Hamilton cycle for every permutation π. Clearly, for $k = 1$, the 2-edge-coloured multigraph is a collection of $t \geq 1$ alternating cycles and, when $t \geq 2$, Bennett's assumption does not hold. Dorninger [271] proved that $G(n, 2, \pi)$ has an alternating Hamilton cycle for every π provided $n \leq 12$. He [271] also showed that for every $n \geq 14$ there exists a permutation π such that $G(n, 2, \pi)$ has no alternating Hamilton cycle. Yeo (private communication, April 1999) proved that the alternating Hamilton cycle problem for the graphs $G(n, 2, \pi)$ is $\mathcal{NP}$-hard. Interestingly enough, $G(n, 3, \pi)$ contains an alternating Hamilton cycle for every permutation π. Thus, the notion of 3-similarity seems to be most consistent with Bennett's assumptions.

In the rest of this section we will prove the following two results:

Theorem 17.6.1 *[271] For every even positive integer $n \leq 12$ and every permutation π of 1,2,...,n, the 2-edge-coloured multigraph $G(n, 2, \pi)$ has an alternating Hamilton cycle.*

Theorem 17.6.2 (Dorninger) *[271] For every even positive integer n and every permutation π of 1,2,...,n, the 2-edge-coloured multigraph $G(n, 3, \pi)$ has an alternating Hamilton cycle.*

17.6.1 Proof of Theorem 17.6.1

In this subsection, which contains certain proofs suggested by Yeo (private communication, April 1999), we consider multigraphs $G = G(n, 2, \pi)$. We recall that $V(G) = V(G_1) = V(G_2) = \{1, 2, \ldots, n\}$, $E(G_1) = \{ij : \ |i - j| \leq 2\}$ and $E(G_2) = \{\pi(i)\pi(j) : \ |i - j| \leq 2\}$ (see Figure 17.6). Clearly, every alternating cycle factor $\mathcal{F}$ of G is the union of a perfect matching F_1 of G_1 and a perfect matching F_2 of G_2. We write $\mathcal{F} = C(F_1, F_2)$.

Suppose that $e = ij$ and $f = pq$ are in F_1 and e and f belong to two distinct cycles X and Y of $\mathcal{F}$. Suppose also that $i < j$, $p < q$ and edges ip and jq are in G_1. If we delete e and f in F_1 and add edges ip and jq, we obtain a new perfect matching F_1' of G_1. Observe that $C(F_1', F_2)$ has one less cycle than $C(F_1, F_2)$ since the vertices of X and Y form a new alternating cycle Z. We call F_1' the **(e, f)-switch** of F_1; the operation to obtain F_1' from F_1 is a **switch** (or, the **(e, f)-switch**).

Let $S = \{\{2t - 1, 2t\} : \ t = 1, 2, \ldots, n/2\}$ and $L = \{\pi(2t - 1)\pi(2t) : t = 1, 2, \ldots, n/2\}$. Clearly, S and L are perfect matchings in G_1 and G_2, respectively.

Lemma 17.6.3 *Let $C(S, L)$ contain m cycles. There is a sequence of switches of edges in S, such that the resulting perfect matching F of G_1 has the property that $C(F, L)$ has at most $\lfloor (m + 1)/2 \rfloor$ cycles. Furthermore, given any cycle C_h in $C(S, L)$ we may choose F, such that all cycles in $C(F, L)$, except possibly C_h, have length at least 4.*

Proof: Let $C(S, L)$ consist of cycles $C_1, C_2, \ldots, C_m$. Let $e_i = \{2r_i - 1, 2r_i\}$ be an edge of C_i, such that r_i is minimum. Assume that the cycles $C_1, C_2, \ldots, C_m$ are labelled such that $1 = r_1 < r_2 < \ldots < r_m$. Define q_i to be the maximum number such that $\{2r_i - 1, 2r_i\}, \{2r_i + 1, 2r_i + 2\}, \ldots, \{2q_i - 1, 2q_i\}$ belong to C_i, for every $i = 1, 2, \ldots, m$. Observe that $1 = r_1 \leq q_1 < r_2 \leq q_2 < \ldots < r_m \leq q_m = n$.

Fix $h \in \{1, 2, \ldots, m\}$. We will now prove that by doing switches every cycle, except possibly C_h, can be merged with another cycle. We perform the switches recursively in the following way. While there is a cycle, C_i with $i < h$, which has not been merged to another cycle do the following: choose i to be the minimum such index and perform the $(\{2q_i - 1, 2q_i\}, \{2q_i + 1, 2q_i + 2\})$-switch. While there is some cycle, C_i with $i > h$, which has not been merged to another cycle do the following: choose i to be the maximum such index and perform the $(\{2r_i - 3, 2r_i - 2\}, \{2r_i - 1, 2r_i\})$-switch. Note that all the above switches use distinct edges.

Since every cycle, except possibly C_h, is merged to another cycle, we must have performed at least $\lfloor m/2 \rfloor$ merges. Therefore there are at most $m - \lfloor m/2 \rfloor = \lfloor (m + 1)/2 \rfloor$ cycles left, which proves the first part of the theorem. The second part follows immediately from the above construction. □

Theorem 17.6.1 follows from the next lemma.

Lemma 17.6.4 *If $C(S, L)$ has at most six cycles, then G has an alternating Hamilton cycle.*

Proof: By the previous lemma, the alternating cycle factor $C(F, L)$ has at most three cycles. Furthermore we may assume that all cycles in $C(F, L)$ have length at least 4, except possibly the cycle containing the vertex $\pi(1)$. If $C(F, L)$ consists of a unique cycle, then we are done. Assume that $C(F, L)$ has three or two cycles. Label them D_1, D_2, D_3 (or D_1, D_2) similarly to that in the proof of Lemma 17.6.3. Let $f_i = \pi(2r_i - 1)\pi(2r_i)$ be an edge of D_i, such that r_i is minimum. Assume that the cycles D_1, D_2, D_3 are labelled such that $1 = r_1 < r_2 < r_3$. Let $f'_i = \pi(2r_i - 3)\pi(2r_i - 2)$ for $i \geq 2$.

Note that all cycles except possibly D_1 have length at least 4. If $C(F, L)$ has two cycles (D_1 and D_2), then construct the (f'_2, f_2)-switch M of L. Clearly, $C(F, M)$ consists of a unique cycle. Assume that $C(F, L)$ has three cycles, D_1, D_2, D_3. Perform the (f'_2, f_2)-switch. If $f'_3 \neq f_2$, then perform the (f'_3, f_3)-switch, which gives the desired cycle. If $f'_3 = f_2$, then let $g = \pi(2j - 1)\pi(2j)$ be the edge of minimum $j > r_3$ which does not lie in D_3, and let $g' = \pi(2j - 3)\pi(2j - 2)$. Now perform the (g', g)-switch, which gives the desired cycle. □

17.6.2 Proof of Theorem 17.6.2

In this subsection, we follow [271]. We consider multigraphs $G = G(n, 3, \pi)$. We recall that $V(G) = V(G_1) = V(G_2) = \{1, 2, \ldots, n\}$, $E(G_1) = \{ij : |i - j| \leq 3\}$ and $E(G_2) = \{\pi(i)\pi(j) : |i - j| \leq 3\}$. We use the same notation as in the previous subsection, in particular, the notation $C_1, C_2, \ldots, C_m$ and $e_1, e_2, \ldots, e_m$ remain valid. Let G^k be the subgraph of G induced by the vertices of the cycles $C_1, C_2, \ldots, C_k$. Let $L^k = L \cap E(G_2^k)$.

We show that, for every $k \geq 1$, there is a perfect matching F^k of G_1^k such that $C(F^k, L^k)$ consists of a single cycle. Clearly, the assertion implies Theorem 17.6.2. Trivially, the assertion is true for $k = 1$. So let us assume that the assertion holds for every $i \leq k - 1$. Let $e_k = \{s + 1, s + 2\}$, where s is an appropriate even integer. Consider the following three cases.

Case 1: The edge $e = \{s, s - j\}$, where $j = 1$ or 2, is in F^{k-1}. Then, the desired F^k is the (e_k, e)-switch of $F^{k-1} + e_k$. Indeed, $C(F^k, L^k)$ consists of a single cycle.

Case 2: The edges $e' = \{s, s - 3\}$ and $e'' = \{s - 1, s - 2\}$ are in F^{k-1}. Let M_1 (M_2) be a perfect matching of G_1^{k-1} obtained from F^{k-1} by replacing edges e', e'' with $\{s, s - 1\}, \{s - 3, s - 2\}$ ($\{s, s - 2\}, \{s - 3, s - 1\}$). Clearly, for some $i \in \{1, 2\}$, $C(M_i, L^{k-1})$ consists of a single cycle H. Since either $\{s, s-1\}$ or $\{s, s-2\}$ is in H, we can apply the transformation of Case 1 to the appropriate matching M_i.

Case 3: The edges $e' = \{s, s - 3\}$ and $e'' = \{s - 1, s - 4\}$ are in F^{k-1}. Then $e = \{s - 2, s - 5\}$ must be in F^{k-1}. Let H be the single cycle of $C(F^{k-1}, L^{k-1})$. Consider the following two subcases.

Subcase 3.1: The vertices of e and e' are in the cyclic order $s-5, s-2, s-3, s$ in H. Replacing e and e' with $\{s-5, s-3\}$ and $\{s, s-2\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-2\} \in M$, we can apply the transformation of Case 1 to M.

Subcase 3.2: The vertices of e and e' are in the cyclic order $s-2, s-5, s-3, s$ in H. If e'' belongs to $H[s-5, s-3]$, then by replacing e, e', e'' with three edges, one of which is $\{s, s-1\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-1\} \in M$, we can apply the transformation of Case 1 to M. If e'' belongs to $H[s, s-2]$, then by replacing e, e', e'' with three edges, one of which is $\{s, s-2\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-2\} \in M$, we can apply the transformation of Case 1 to M. □

17.7 Gaussian Elimination

In many applications, such as modeling a problem by a system of differential equations and then solving this system by numerical methods (cf. the book [280] by Duff, Erisman and Reid), the final step of the solution of the problem under consideration consists of solving a system of linear equations: $Ax = b$, where $A = [a_{ij}]$ is an $n \times n$ matrix of coefficients, b is a given vector of dimension n and x is a vector of unknowns. In a considerable number of applications the matrix A is sparse, i.e., most entries of A are zero. The system $Ax = b$ is often solved by the Gaussian elimination method. To use this method, the only requirement is that all diagonal elements a_{ii} of matrix A can be made by non-zero row and column permutations.

In many cases in practice, a sparse matrix A has some special structure, which allows one to solve the system much faster than just using Gaussian elimination directly. One of the most important such structures is block-triangular structure. Let $n_1, n_2, \ldots, n_k$ be natural numbers such that $1 \leq n_1 < n_2 < \ldots < n_k = n$ and let $n_0 = 0$. We call the submatrices $A^{(p)} = [a_{i_p, j_p}]$, with $n_{p-1} + 1 \leq i_p, j_p \leq n_p$, the **main $(n_1, \ldots, n_p)$-blocks** (or just main blocks). We say that A has **$(n_1, \ldots, n_p)$-block-triangular structure** (or just block-triangular structure) if all entries of A below the main blocks are zero. (More precisely, one should call this structure upper block-triangular [280], but since we do not consider lower block-triangular structure here, we will omit the word 'upper'.) The matrix

$$\begin{bmatrix} 3 & 2 & 4 & 1 \\ 5 & 6 & 0 & 0 \\ 3 & 0 & 7 & 9 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

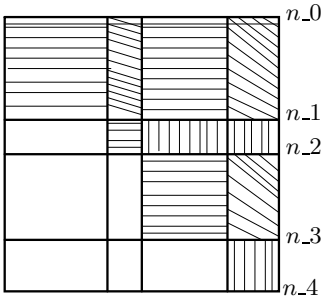


Figure 17.7 An (n_1, n_2, n_3, n_4)-block-triangular structure. White space consists of entries equal zero.

has $(3, 4)$-block-triangular structure. See also Figure 17.7.

If A has block-triangular structure, we solve first the system $A^{(p)}x^{(p)} = b^{(p)}$, where $x^{(p)}$ $(b^{(p)})$ is the vector consisting of n_p last coordinates of x (b). The values of coordinates of $x^{(p)}$, which we found, equal the values of the corresponding unknowns in the system $Ax = b$ since in the last n_p rows of A all coefficients except for some in the last n_p columns are zero. Taking into consideration that the values of coordinates of $x^{(p)}$ are already found, we can compute the values of coordinates of $x^{(p-1)}$ using the block $A^{(p-1)}$. Similarly, using all blocks of A (in the decreasing order of their indices) we can compute all coordinates in x.

However, quite often the block-triangular structure of A is **hidden**, i.e., A has no block-triangular structure, but A can be transformed into a matrix with block-triangular structure after certain permutations π and τ of its rows and columns, respectively. Here we are interested in using the Gaussian elimination method and thus we assume that all diagonal entries of A are non-zero (when it is possible, one can find permutations of rows and columns of A, which bring non-zero diagonal to A using perfect matchings in bipartite graphs, see [280]). Therefore, we do not wish to change the diagonal entries of A. This can be achieved by using only simultaneous permutations of rows and columns of A, i.e., $\pi = \tau$.

To reveal hidden block-triangular structure of A, the following approach can be used. Let us replace all non-zero entries of A by 1. We obtain matrix $B = [b_{ij}]$, which can be viewed as the adjacency matrix of some directed pseudograph D with vertex set $\{v_1, \ldots, v_n\}$, i.e., $b_{ij} = 1$ if and only if $v_i \rightarrow v_j$ in D. (Clearly, D has no parallel arcs, but due to the assumption on the diagonal elements it has a loop at every vertex.) Suppose that D is not strong, $D_1, \ldots, D_p$ is the acyclic ordering of the strong components of D (i.e., there is no arc from D_j to D_i if $j > i$) and the vertices of D are ordered $v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}$ such that

$$V(D_i) = \{v_{\pi(n_i+1)}, v_{\pi(n_i+2)}, \ldots, v_{\pi(n_{i+1})}\}.$$

It is easy to see that B has $(n_1, \ldots, n_p)$-block-triangular structure. This implies that A has block-triangular structure. The above observation suggests the following procedure to reveal hidden block-triangular structure of A.

1. Replace every non-zero entry of A by 1 to obtain a $(0,1)$-matrix B.
2. Construct a directed pseudograph D with vertex set $\{v_1, \ldots, v_n\}$ such that B is the adjacency matrix of D.
3. Find the strong components of D. If D is strong, then B (and thus A) does not have hidden block-triangular structure[2]. If D is not strong, let $D_1, \ldots, D_p$ be the strong components of D (in acyclic order). Find a permutation π on $\{1, \ldots, n\}$ such that

$$V(D_i) = \{v_{\pi(n_i+1)}, v_{\pi(n_i+2)}, \ldots, v_{\pi(n_{i+1})}\}.$$

This permutation reveals hidden block-triangular structure of B (and thus A). Use π to permute rows and columns of A and coordinates of x and b.

To perform Step 3 one may use Tarjan's algorithm in Section 5.2.

We will illustrate the procedure above by the following example. Suppose we wish to solve the system

$$\begin{aligned} x_1 \qquad\quad + 3x_3 + 8x_4 &= 2, \\ x_2 \qquad\quad + 5x_4 &= 1, \\ 2x_1 + 2x_2 + 4x_3 + 9x_4 &= 6, \\ 3x_2 \qquad\quad + 2x_4 &= 3. \end{aligned}$$

We first construct the matrix B and the directed pseudograph D. We have $V(D) = \{v_1, v_2, v_2, v_4\}$ and

$$A(D) = \{v_1v_3, v_1v_4, v_2v_4, v_3v_1, v_3v_2, v_3v_4, v_4v_2\} \cup \{v_iv_i : \ i = 1, 2, 3, 4\}.$$

The digraph D has strong components $D^{(1)}$ and $D^{(2)}$, which are subdigraphs of D induced by $\{v_1, v_3\}$ and $\{v_2, v_4\}$, respectively. These components suggest the following permutation π, $\pi(i) = i$ for $i = 1, 4$, $\pi(2) = 3$ and $\pi(3) = 2$, of rows and columns of A as well as elements of x and b, the right-hand side. As a result, we obtain the following:

$$\begin{aligned} x'_1 + 3x'_2 \qquad\quad + 8x'_4 &= 2, \\ 2x'_1 + 4x'_2 + 2x'_3 \quad 9x'_4 &= 6, \\ x'_3 + 5x'_4 &= 1, \\ 3x'_3 + 2x'_4 &= 3, \end{aligned}$$

where $x'_i = x_i$ for $i = 1, 4$, $x'_2 = x_3$ and $x'_3 = x_2$.

[2] Provided we do not change the set of entries of the diagonal of A.

Solving the last two equations separately, we obtain $x_3' = 1,\ x_4' = 0$. Now solving the first two equations, we see that $x_1' = 2,\ x_2' = 0$. Hence, $x_1 = 2,\ x_2 = 1,\ x_3 = x_4 = 0$.

A discussion on practical experience with revealing and exploiting block-triangular structures is given in [280].

17.8 Markov Chains

Markov chains are a special type of stochastic processes, which have numerous applications in genetics, economics, sport science, etc. We will see in this section that the corresponding digraph cycle structure is of great importance to Markov chains.

Let $S_1, S_2, \ldots, S_n$ be all possible states of some system. The system is initially in a state S_i with probability $p_i^{(0)}$, $i = 1, 2, \ldots, n$. At every step the system moves from the state S_i, which it is currently in, to a state S_j with probability p_{ij} depending only on i and j. Clearly, for all i, j, we have $0 \leq p_{ij} \leq 1$ and $\sum_{j=1}^{n} p_{ij} = 1$ for every $i = 1, 2, \ldots, n$. The stochastic process, which we have under these conditions, is called a **Markov chain** (for more details on Markov chains, see e.g. Feller [312] and Kemeny and Snell [589])[3]. Let $\pi^{(0)} = (p_1^{(0)}, \ldots, p_n^{(0)})$, let $p_i^{(k)}$ be the probability of the system to be in state S_i after the kth step, and let $\pi^{(k)} = (p_1^{(k)}, \ldots, p_n^{(k)})$. It is well-known that the vector $\pi^{(k)}$ can be found as follows: $\pi^{(k)} = \pi^{(0)} P^k$, where $P = [p_{ij}]$. However, this equality is difficult to use directly if we wish to know the probability distribution $\pi^{(k)}$ after a large number of steps. In fact, $\pi = \lim_{n \to \infty} \pi^{(0)} P^k$ is often of interest (if it exists).

To investigate when this limit exists and to see what happens when this limit does not exist, it is very useful to study directed pseudographs D associated with Markov chains. The vertex set of D is $\{v_1, \ldots, v_n\}$ and the arc set is $\{v_i v_j : \ p_{ij} > 0, 1 \leq i, j \leq n\}$; D has no parallel arcs but may have loops. It is not difficult to see that for $n \to \infty$ with probability tending to 1 the system will be in one of the stages corresponding to the vertices in the terminal strong components of D (once the system is in such a 'vertex' it cannot escape the corresponding terminal strong component). This shows that it suffices to study only strong directed pseudographs D corresponding to Markov chains. When D is strong, the following parameter of D is of interest. The **period** $p(D)$ of D is the greatest common divisor of the cycle lengths of D. If $p(D) = 1$, then it is well-known that the limit above does exist and, moreover, does not depend on the initial distribution $\pi^{(0)}$. If

[3] Some readers may find it useful to consider $S_1, \ldots, S_n$ as water containers, $p_i^{(0)}$ as the fraction of water in S_i initially, and p_{ij} as the fraction of water in S_i to be moved to S_j in one step. We are interested in how the water will be distributed after a large number of steps.

$p(D) \geq 2$, then the situation is absolutely different since D has a quite special structure. Actually, if $p(D)$ is even, then by Theorem 2.2.1 we obtain that D is bipartite. However, the following stronger assertion, which generalizes Theorem 2.2.1, holds[4]:

Theorem 17.8.1 *If a strong digraph $D = (V, A)$ has period $p \geq 2$, then V can be partitioned into sets $V_1, V_2, \ldots, V_p$ such that every arc with tail in V_i has its head in V_{i+1} for every $i = 1, 2, \ldots, p$, where $V_{p+1} = V_1$.*

Proof: Let $D = (V, A)$ have period $p \geq 2$. Every closed walk W of D, being an eulerian digraph, is the union of cycles (see Theorem 13.4.1); hence the length of W equals 0 modulo p. Let x, y be a pair of distinct vertices of D and let P, Q be a pair of distinct (x, y)-paths in D. We claim that the lengths of P and Q are equal modulo p. Indeed, let R be a (y, x)-path in D. Both P and Q form closed walks with R; hence our last claim follows from the remark above.

Since D is strong, it can be constructed from a cycle using ear composition (see Section 5.3). We start from a cycle C and in every iteration add to the current digraph H a path whose vertices apart from the end-vertices do not belong to H or a cycle with only one vertex in common with H. Initially, all sets $V_1, V_2, \ldots, V_p$ are empty. We choose an arbitrary vertex x in C and consider every vertex y in C; we put y in V_i if the length of $C[x, y]$ equals i modulo p. In the first iteration of ear composition, we add a path or cycle R to C. Let z be the initial vertex of R if R is a path or the only vertex of R in common with C if R is a cycle, and let $z \in V_k$. We consider every vertex y in R and put y in V_{k+i} if the length of $R[z, y]$ equals i modulo p. Note that if R is a path, then its terminal vertex z' will be put in the same set V_j, where it has been already, since otherwise we could find a pair of (z, z')-paths, whose lengths are not equal modulo p. We proceed with ear composition as above and in the end we will have V partitioned into $V_1, V_2, \ldots, V_p$ such that every arc with tail in V_i has its head in V_{i+1} for every $i = 1, 2, \ldots, p$ (by the way we have formed V_i's). □

Clearly, when the period of the digraph of a Markov chain is larger than 1, the limit introduced above does not exist; instead the Markov chain moves 'cyclically'. Theorem 17.8.1 shows that a strong digraph D of order n and period $p \geq 2$ is a spanning subdigraph of $\vec{C}_p[\overline{K}_{n_1}, \ldots, \overline{K}_{n_p}]$, for some $n_1, n_2, \ldots, n_p$ such that $\sum_{i=1}^{p} n_i = n$.

There are two algorithms to compute the period of a strong digraph in optimal time $O(n+m)$. The first algorithm is by Balcer and Veinott [59] and based on the following idea. If, for a vertex x of $d^+(x) \geq 2$, we contract all vertices in $N^+(x)$ and delete any parallel arcs obtained, then the resulting digraph has the same period as the original digraph by Theorem 17.8.1.

[4] We have been unable to trace the first paper, where this result was proved. Our proof of this theorem makes use of some results considered in previous chapters.

Repeating this iteration, we will finally obtain a cycle C (see Exercise 17.6). Clearly, the length of C is the desired period. For example, the digraph H obtained from a 3-cycle and a 6-cycle by identifying one of their vertices after five iterations above becomes a 3-cycle (see Figure 17.8). The second algorithm is due to Knuth (see [49]) and based on DFS-trees.

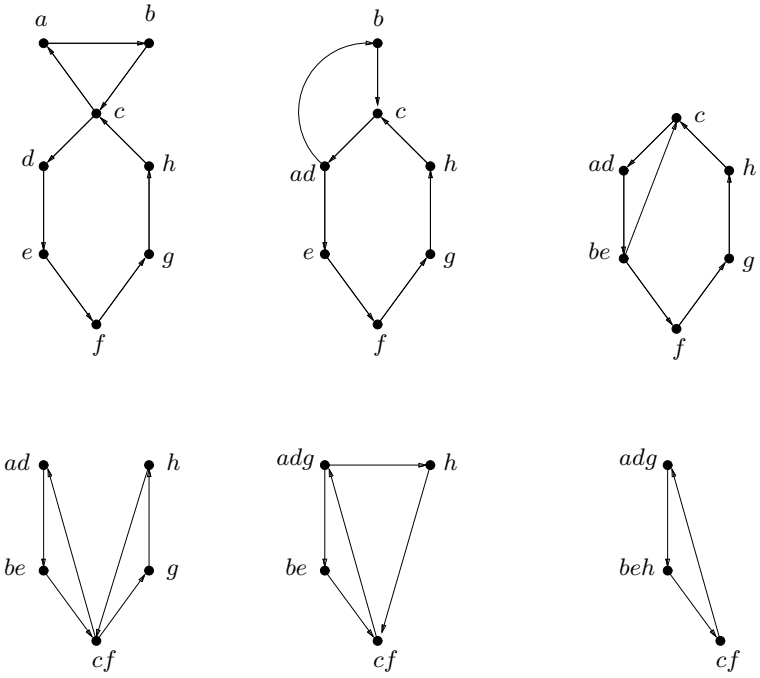


Figure 17.8 Illustrating the Balcer-Veinott algorithm.

17.9 List Edge-Colourings

The topic of this section may seem to have nothing to do with directed graphs, but as we will see, directed graphs have been a useful tool for solving the so-called Dinitz problem which we now describe. Our discussion in this section is inspired by the book [14] by Aigner and Ziegler and Galvin's paper [392].

An $n \times n$ matrix M over the integers $\{1, 2, \ldots, n\}$ is a **Latin square** (of size n) if no two entries in the same row and no two entries in the same column are equal. It is an easy exercise to show that for every integer $n \geq 1$ there exists a Latin square (Exercise 17.8).

A **proper edge-colouring** of an undirected graph $G = (V, E)$ is an assignment of integers to the edges in such a way that no two edges with a

common end-vertex receive the same colour. The smallest k such that a graph G has a proper edge-colouring using only colours from the set $\{1, 2, \ldots, k\}$ is called the **chromatic index** of G. Thus it is easy to see that there is a 1-1 correspondence between the set of Latin squares of size n and the set of proper edge-colourings of the complete bipartite graph $K_{n,n}$ using colours $\{1, 2, \ldots, n\}$.

Proper edge-colourings are useful for various practical applications such as time table construction, see e.g. the book by Jensen and Toft [564]. In the rest of this section we omit the word 'proper' since only proper edge-colourings will be considered.

In 1979 Dinitz raised the following problem (see e.g. [297, 299]): suppose we are given an $n \times n$ matrix whose (i, j) entry is a set $C(i, j)$ of n integers, $1 \leq i, j \leq n$, is it always possible to choose from each set $C(i, j)$ one element c_{ij} in such a way that the elements in each row are distinct and the elements in each column are distinct?

The Dinitz problem can be reformulated in terms of edge-colourings of complete bipartite graphs. Suppose that we are given, for each edge ij of the complete bipartite graph $K_{n,n}$, a set $C(i, j)$ of possible colours for that edge. Does there always exist an edge-colouring of $K_{n,n}$ so that for each edge ij the colour c_{ij} of ij belongs to $C(i, j)$? In this formulation the Dinitz problem is just a special case of the more general **list colouring conjecture** (see e.g. the book by Jensen and Toft [564]) which states that if a graph G has an edge-colouring with k colours, then no matter how we assign to each edge e of G a set C_e of k arbitrary colours, G has an edge colouring such that the colour of the edge e belongs to the set C_e for each $e \in E$. Such a colouring is called a **list edge-colouring** of G. An important step towards settling the Dinitz conjecture was made by Janssen [563] who proved that if all lists have length $n + 1$ (instead of n), then a solution always exists.

In order to apply results on kernel in digraphs we study the line graph of $K_{n,n}$. The definition of a line graph is analogous to that of a line digraph: $L(G)$ contains a vertex for each edge of G and two vertices in $L(G)$ are joined by an edge if and only if the corresponding edges have an end-vertex in common. It is easy to see that every list edge-colouring of $K_{n,n}$ corresponds to a list vertex colouring (in short a **list colouring**) of $L(K_{n,n})$ using the same sets (lists). Hence, in order to solve the Dinitz problem, it suffices to prove that no matter which sets $C_{11}, C_{12}, \ldots, C_{nn}$, each of size n, we associate with the n^2 vertices of $L(K_{n,n})$, there exists a proper vertex colouring of $L(K_{n,n})$ such that the colour of the vertex i is chosen from the corresponding set C_i.

Now we return to digraphs. The following lemma is attributed to Bondy, Boppana and Siegel in [30, Remark 2.4, p. 129] (see also [392]).

Lemma 17.9.1 *Let $D = (V, A)$ be a digraph and suppose that for each vertex $v \in V$ we are given a prescribed set $C(v)$ of colours satisfying $|C(v)| > d^+(v)$. If D is kernel perfect (i.e., every induced subdigraph of D has a kernel), then*

there exists a list colouring of $UG(D)$ which uses a colour from $C(v)$ for each $v \in V$.

Proof: The proof is by induction on n, the case $n = 1$ being trivially true. Fix a colour c which belongs to at least one of the sets $C(v)$, $v \in V$ and let $X(c) := \{v \in V | c \in C(v)\}$. By the assumption of the lemma the induced subdigraph $D\langle X(c)\rangle$ has a kernel Y. Now colour each vertex of V which belongs to Y by colour c (which is a proper choice by the definition of $X(c)$) and consider the digraph $D' = D - Y$ with colour sets $C'(v) = C(v) - \{c\}$. Notice that for each vertex $v \in X(c) - Y$ the out-degree of v in D' is at least one smaller than the out-degree of v in D and hence we have $|C'(v)| > d^+_{D'}(v)$ for all $v \in V(D')$. Furthermore, every vertex u that does not belong to $X(c)$ has $|C(u)| = |C'(u)|$. Thus, by the induction hypothesis, there is a list colouring of D' which uses a colour from $C'(v)$ for each $v \in V(D')$. Using that colouring along with the colour c for vertices in Y we achieve the desired colouring. □

From Lemma 17.9.1 we see that if we can establish the existence of an orientation D of $L(K_{n,n})$ such that every induced subgraph of D has a kernel and $d^+_D(v) \leq n - 1$ for each vertex v, then we have proved that $L(K_{n,n})$ has list chromatic number at most n as desired.

We show below that in order to obtain such an orientation we can use any n-edge-colouring of $K_{n,n}$ and orient appropriately. To prove the existence of a kernel in each induced subgraph we use the concept of stable matchings which we discuss below.

Below we assume that we are given a bipartite graph $B = (X \cup Y, E)$ and that for each vertex $u \in X \cup Y$ there is a fixed ordering $>_u$ on the neighbours of u. That is, $>_u$ induces an ordering $v_1 >_u v_2 >_u \ldots >_u v_{d_B(u)}$ on $N_B(u)$.

A matching M in $B = (X \cup Y, E)$ is **stable** with respect to the family of orderings $\{>_u | u \in X \cup Y\}$ if the following holds for all $uv \in E - M$: either $uy \in M$ for some y such that $y >_u v$ or $xv \in M$ for some x with $x >_v u$.

Stable matchings have an amusing real-life interpretation. Consider X as a set of men and Y as a set of women and let the existence of an edge $xy \in E$, $x \in X, y \in Y$ mean that person x and y might marry. As we saw in Theorem 4.11.2, given B we can determine in polynomial time the maximum number of men and women who can marry without anybody committing bigamy. However, in practice the fact that a man x and a woman y might marry does not mean that this particular choice is the optimal one for x or y. Hence, in a more realistic setting each person has a list of possible spouses and some ranking among these as to who would be the favorite choice down to the least wanted spouse (but still a possible choice). Now we see that this description corresponds to the orderings described above. Furthermore, stability of a given matching corresponds to saying that among the men and women who are paired for marriage there is no pair xy for which x prefers some other woman y' to y and at the same time woman y prefers some other

man x' to x. So in some sense a stable matching corresponds to a situation where no pair is highly likely to split up.

The concept of stable matchings was introduced by Gale and Shapley who proved the following slightly surprising fact. We leave the proof as Exercise 17.9.

Theorem 17.9.2 *[379] For every bipartite graph $B = (X \cup Y, E)$ and every family of orderings $\{>_u | u \in X \cup Y\}$ which arises from a local linear ordering of the neighbours of each vertex in B, there exists a stable matching with respect to $\{>_u | u \in X \cup Y\}$.* □

In Exercise 17.10 the reader is asked to show by an example that it is not always true that there exists a maximum matching which is stable.

For more information about stable matchings see e.g. the papers [60, 61] by Balinski and Ratier. Now we are ready to describe Galvin's proof of the Dinitz conjecture.

Theorem 17.9.3 *[392] For every $n \geq 1$ the complete bipartite graph $K_{n,n}$ has list chromatic index n.*

Proof: Denote the vertices of $L(K_{n,n})$ by (i,j), $1 \leq i,j \leq n$, where (i,j) is adjacent to (i',j') if and only if $i = i'$ or $j = j'$, but not both. Let Q be any Latin square of size n (recall that this corresponds to a proper edge-colouring of $K_{n,n}$) and denote by Q_{ij} the ijth entry of Q. Let D_n be the oriented graph obtained from $L(K_{n,n})$ by orienting the edges as follows:

$(i,j) \rightarrow (i,j')$ if and only if $Q_{ij} < Q_{ij'}$ and
$(i,j) \rightarrow (i',j)$ if and only if $Q_{ij} > Q_{i'j}$ (see Figure 17.9).

1	2	3
2	3	1
3	1	2

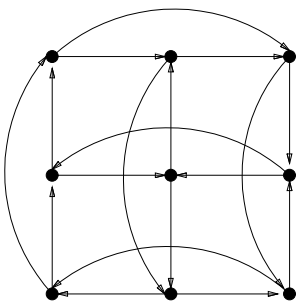


Figure 17.9 The orientation of $L(K_{3,3})$ based on a Latin square of size 3.

It is easy to see that D is $(n-1)$-regular (Exercise 17.11). Thus, by Lemma 17.9.1 we just have to prove that every induced subdigraph of D has a kernel. To prove this we use Theorem 17.9.2.

Let D' be an arbitrary induced subdigraph of D and let $B = (X, Y, E)$ be the corresponding bipartite subgraph of $K_{n,n}$ induced by those edges for which the corresponding vertex (i, j) belongs to D'. For each vertex $i \in X$ we define an ordering $>_i$ of the neighbours of i in B by letting $j' >_i j$ whenever $(i, j) \rightarrow (i, j')$ in D. Similarly, for each $j \in Y$ we define the ordering $>_j$ of the neighbours of j in B by letting $i' >_j i$ whenever $(i, j) \rightarrow (i', j)$ in D.

According to Theorem 17.9.2 B has a stable matching M with respect to $\{>_u: \ u \in X \cup Y\}$. Since M is also a matching in $K_{n,n}$ the corresponding vertices are independent in D. Furthermore, it follows from the fact that M is stable with respect to $\{>_u: \ u \in X \cup Y\}$ that for every (i, j) such that $ij \notin M$, either there exist $j' \in Y$ such that $ij' \in M$ and $j' >_i j$ or there exists an $i' \in X$ such that $i'j \in M$ and $i' >_j i$. In the first case we have $(i, j) \rightarrow (i, j')$ and in the second case we have $(i, j) \rightarrow (i', j)$ in D. Thus we have shown that every vertex of D' which is not in M dominates a vertex in M. Hence M is a kernel and the proof is complete. □

The idea of orienting $L(K_{n,n})$ as we did above is due to Maffray [678].

17.10 Digraph Models of Bartering

Recently Özturan [737, 739, 738] developed several models of bartering that use digraphs or their generalizations. We restrict ourselves to the two models that use digraphs.

In [737, 739, 738] Özturan discusses the importance of bartering and certain combinations of bartering and auctions in electronic commerce and other applications. One example where bartering is more appropriate than usual trading is in the area of Web domain names. Indeed, there is a high demand for Web domain names and many domain names are put for sale. Nevertheless, the volume of sales is relatively low [738]. One possible reason for this phenomenon is that the price of the domain names is rather high and the buyers do not have enough money to buy them. However, often the buyers also try to sell some domain names, so the deadlock could be broken if no-money barter deals come into play. This seems to be possible because often the domain names have similar prices. Other examples of practical necessity of bartering are seen in countries with high inflation overall (recent examples are Argentina and Turkey) or just in the housing market (one recent example is UK). In such cases, selling a product may mean not being able to buy a similar value product in just a few months. In bartering, if a person/company sells a product, then it buys another one at the same time.

Consider Model 1 introduced by Özturan [737]. This model is simpler than Model 2 which we consider afterwards. In both models we assume that all products (called **items**) are of equal or almost equal value (i.e., people or companies involved in bartering are ready to compensate for the difference in value between any two items). The items can be viewed as vertices of a

digraph D; there is an arc (i, j) in D if somebody is willing to exchange i for j. It is natural to find as many items as possible that can be involved in an exchange in which every person who gives an item receives another one instead. (For example, in the case of a property agent who gets a fee for every house he or she sells.) In terms of graph theory, we wish to find a cycle subdigraph of maximum order. Such a subdigraph can be found in time $O(n^3)$, see Theorem 13.8.1.

In Model 2, we consider a more complicated situation where people or companies (called **barterers**) may have several **instances** of the same item and each of the barterers wishes to receive as many instances as possible. We will show how to find a solution in which the total number of exchanged instances is maximum. Let $I = \{i_1, i_2, \ldots, i_p\}$ be the set of items, let $B = \{b_1, b_2, \ldots, b_q\}$ be the set of barterers, let $g(t, s)$ and $r(t, s)$ denote the number of copies of item i_s that barterer b_t is ready to give and receive, respectively (often one of two parameters is 0, but there is no need to assume that). The network we introduce here for Model 2 is simpler than the one described by Özturan [737], but the two are essentially equivalent.

Construct a bipartite network N with partite sets I and B. There is an arc (b_t, i_s), $t \in [q]$, $s \in [p]$, in N of capacity $g(t, s)$ if $g(t, s) > 0$ and there is an arc (i_s, b_t) in N of capacity $r(t, s)$ if $r(t, s) > 0$. We assign cost -1 to each vertex in I and 0 to each vertex in B. Observe that the minimum cost of a circulation in N is equal to the maximum total number of exchanged instances and a minimum cost circulation can be transformed into an optimal bartering. The network N is of interest since there are many efficient algorithms for finding a minimum cost circulation, see Chapter 4. Figure 17.10 provides an example of the bipartite network N. The first number on each arc gives its capacity and the second number the amount of flow in the minimum cost circulation. The costs of the vertices are not shown.

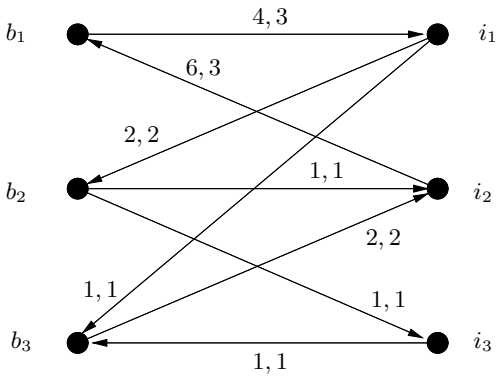


Figure 17.10 An example of network N.

17.11 PERT/CPM in Project Scheduling

Often a large project consists of many activities some of which can be done in parallel, others can start only after certain activities have been accomplished. In such cases, the **critical path method (CPM)** and **Program Evaluation and Review Technique (PERT)** are of interest. They allow one to predict when the project will be finished and monitor the progress of the project. They allow one to identify certain activities which should be finished on time if the predicted completion time is to be achieved.

CPM and PERT were developed independently in the late 1950s. They have many features in common and several others that distinguish them. However, over the years the two methods have practically merged into one combined approach often called PERT/CPM. This approach has been used in a large number of projects including a new plant construction, NASA space exploration, movie production and ship building (see, e.g., the book [528] by Hillier and Lieberman). PERT/CPM has many tools for project management, but we will restrict ourselves only to a brief introduction and refer the reader to various books on operations research such as [528, 535, 710] for more information on the method.

We will introduce PERT/CPM using an example. Suppose the tasks to complete construction of a house are as follows (in brackets we give their duration in days): Wiring (5), Plumbing (8), Walls & Ceilings (10), Floors (4), Exterior Decorating (3) and Interior Decorating (12). We cannot start doing Walls & Ceilings and Floors before Wiring and Plumbing are accomplished, we cannot do Exterior Decorating before Walls & Ceilings are completed and we cannot do Interior Decorating before Walls & Ceilings and Floors are accomplished. How much time do we need to accomplish the construction?

To solve the problem we first construct a digraph N, which is called an **activity-on-node (AON)** project network[5]. We associate the vertices of N with the starting and finishing points of the projects (vertices S and F) and with the activities described above, i.e., Wiring (Wi), Plumbing (Pl), Floors (Fl), Walls & Ceiling (WC), Interior Decorating (ID) and Exterior Decorating (ED). The network N is a vertex-weighted digraph, where the weights of S and F are 0 and the weight of any other vertex is the duration of the corresponding activities. Observe that the duration of the house construction project equals the maximum weight of an (S, F)-path.

As in the example above, in the general case, an AON network D is a vertex-weighted digraph with the starting and finishing vertices S and F. Our initial aim is to find the maximum weight of an (S, F)-path in D. Since D is an acyclic digraph, this can be done in linear time using the algorithm mentioned in Theorem 3.3.5 after the vertex splitting procedure (see Subsection 4.2.4).

[5] Original versions of PERT and CPM used another type of network, **activity-on-arc (AOA)** project network, but AOA networks are significantly harder to construct and change than AON networks and it makes more sense to use AON networks rather than AOA ones.

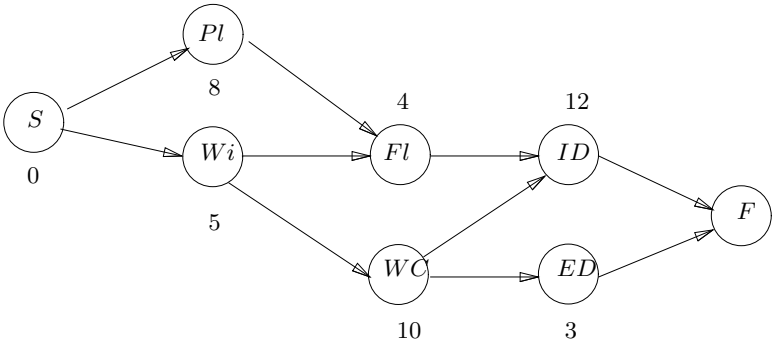


Figure 17.11 House construction network.

We can also use dynamic programming directly: for a vertex x of D let $t(x)$ be the earlier time when the activity corresponding to x can be accomplished. Then $t(S) = 0$ and for any other vertex x, we have $t(x) = \ell(x) + \max\{t(y) : y \in N^-(x)\}$, where $\ell(x)$ is the duration of the activity associated with x. To ensure that we know the value of $t(y)$ for each in-neighbour of y of x, we consider the vertices of D in an acyclic ordering (see Section 2.1).

It is easy to see that the maximum weight of an (S, F)-path in N (our example) is 27 days and the path is S, Wi, WC, ID, F. Every maximum weight (S, F)-path is called **critical** and every vertex (and the corresponding activity) belonging to a critical path is **critical**. Observe that to ensure that the project takes no longer than required, no critical activity should be delayed. At the same time, delay with non-critical activities may not affect the duration of the project. For example, if we do Plumbing 13 days instead of 8 days, the project will be finished in 27 days anyway. This means that the project manager has to monitor mainly critical activities and may delay non-critical activities in order to enforce critical ones (e.g., by moving workforce from a non-critical activity to a critical one).

The manager may want to expedite the project (if, for example, earlier completion will result in a considerable bonus) by spending more money on it. This issue can be investigated using linear programming, see Hillier and Lieberman [528].

17.12 Finite Automata

In computer program compilation, we need to know which words in the program are allowed by the grammar of the language the program is written in and which are not allowed. This kind of recognition can be done efficiently because the allowed words are strings of a regular language and one can efficiently check whether a string belongs to a regular language (in other words, to a regular expression) [12].

An **alphabet** (usually denoted by Σ) is any set of symbols called **letters**. **Strings** (or, **words**) in an alphabet Σ are ordered sequences of letters. The **empty string** is denoted by ϵ. For an alphabet Σ, a set of strings R is called a **regular language** if either $R = \emptyset$ or $R = \{\epsilon\}$ or $R = \{s\}$, where $s \in \Sigma$ or there are regular languages P and Q such that either $R = PQ$ (the set of strings obtained by concatenating a string from P with a string from Q) or $R = P^*$ (the set of all strings obtained as concatenation of zero or more strings from P) or $R = P \cup Q$ (this union is often denoted by $P|Q$). For example, the set of integers can be given by the following regular language (expression):

$$(-|\epsilon)(1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)^*|0.$$

A **(deterministic) finite automaton** M is a 5-tuple $(Q, q_0, F, \Sigma, \delta)$, where Q is a finite set of **states**, $q_0 \in Q$ is a **start state**, $F \subset Q$ is a set of **final (or, accepting) states**, Σ is a finite **input alphabet** and δ is a function from $Q \times \Sigma$ to Q called the **transition function** of M. A finite automaton M scans an input string S of letters from Σ and moves according to the transition function; M starts in q_0 and moves from q_0 to the state $\delta(q_0, s_1)$, where s_1 is the first letter in S. In Step $i \geq 1$, if M is in a state q, then it moves to the state $\delta(q, s_i)$, where s_i is the ith letter of S; M stops after the last letter of S has been read and the corresponding move has been performed. We say that a string S is **accepted** if M stops in a state belonging to F. We denote by $L(M)$ the set (language) of all strings accepted by M.

Consider, for example, a finite automation A with $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$, $\Sigma = \{a, b\}$ and δ defined as follows: $\delta(q_0, a) = q_2$, $\delta(q_0, b) = q_1$, $\delta(q_1, a) = q_1$, $\delta(q_1, b) = q_2$, $\delta(q_2, a) = q_1$, $\delta(q_2, b) = q_2$. The automation A can be depicted as a directed pseudograph D with arcs labeled by letters from Σ, see Figure 17.12 (left). One can check (Exercise 17.13) that $L(A) = ab^*(aa^*bb^*)^*|ba^*b(aa^*bb^*)^*$.

In general, a finite automaton M can be represented by a directed pseudograph $D(M)$ called the **state diagram** of M. Observe that $D(M)$ may have parallel arcs and/or loops as there may be a state $q \in Q$, and a letter s_i in S and/or two distinct letters s_j, s_k in S such that $\delta(q, s_i) = q$ and $\delta(q, s_j) = \delta(q, s_k)$. Let S be an input string for an automaton M. The moves of M from q_0 to the state where M stops form a walk, which we call the **S-walk**. Notice that S is accepted if and only if the S-walk terminates in a vertex from F.

Figure 17.12 Deterministic (left) and non-deterministic (right) finite automata.

It is well-known that a set R of strings is a regular language if and only if there is a finite automaton M such that $R = L(M)$ [533, 642]. Moreover, for each finite automaton M, $L(M)$ is a regular language.

Let M be a finite automaton. We call a string S **synchronizing** for M if no matter which state of M is proclaimed a start state the S-walks terminates in the same state. If an automaton M has a synchronizing string, M is resistant against input errors in the following sense: after detection of an error, we can reset M back to its original state using a synchronizing string. The problem of existence of a synchronizing string is related to the following **Road Colouring Conjecture**: Let D be a strong digraph in which every vertex has the same out-degree k and let the period[6] of D equal 1. Then there is an assignment $f : A(D) \rightarrow [k]$ such that (D, f) is the state diagram of a finite automaton for which there exists a synchronizing string. This conjecture was stated by Adler, Goodwyn and Weiss [7] and solved recently by Trahtman [875] (earlier Kari [583] solved the conjecture for the special case of eulerian digraphs).

Figure 17.12 (right) depicts the state diagram of a non-deterministic finite automaton B. In B, we do not demand that for every letter c and every vertex (state) q_i there is a unique arc from q_i labelled by c. There may be several arcs from q_i labelled by c or none. Moreover, some arcs from q_i may be labelled by ϵ, the empty string, which means that the automaton is allowed to move from q_i along those arcs without reading the input string S. Formally, a **non-deterministic finite automaton** M is a 5-tuple $(Q, q_0, F, \Sigma, \Delta)$, where Q is a finite set of states, $q_0 \in Q$ is a start state, $F \subset Q$ is a set of final (or, accepting) states, Σ is a finite alphabet and Δ is a subset of $Q \times (\Sigma \cup \{\epsilon\}) \times Q$ called the **transition relation** of M. One can easily define the state diagram $D(M)$ of a non-deterministic finite automaton M.

Let us now replace every arc label ϵ in $D(M)$ by a letter $\omega \notin \Sigma$ and denote the resulting directed pseudograph by $D^*(M)$. Since $D^*(M)$ may contain arcs labelled by ω, accepted strings are defined in a more sophisticated way. Perhaps the clearest way to define accepted strings is by considering S'-walks

[6] Recall that the period of a digraph D is the greatest common divisor of the cycle lengths of D, see Section 17.8.

in $D^*(M)$, where S' is a string of letters from $\Sigma \cup \{\omega\}$ (S'-walks can be easily defined). An input string $S = s_1 s_2 \dots s_r$ is **accepted** by M if there exist non-negative integers $n_1, n_2, \dots, n_r$ such that the S'-walk of $D^*(M)$ terminates in a vertex of F, where $S' = s_1 \omega^{n_1} s_2 \omega^{n_2} \dots \omega^{n_{r-1}} s_r \omega^{n_r}$ and ω^k denotes the string consisting of k letters ω (in particular, $\omega^0 = \epsilon$). We denote by $L(M)$ the set of strings accepted by M.

One may think that non-deterministic automata are more powerful than deterministic ones, but this is not true.

Theorem 17.12.1 *For every non-deterministic automaton M there is a deterministic automaton M' such that $L(M) = L(M')$.* □

A complete proof of this theorem can be found in [642]; the proof is not difficult, but rather lengthy and we will only consider briefly how to build M' from $M = (Q, q_0, F, \Sigma, \Delta)$. Let $D_\epsilon(M)$ denote the transitive closure of the subdigraph of the state diagram $D(M)$ induced by the arcs labelled ϵ (for the definition and results on transitive closure, see Section 2.3). For a state q_i let $N_\epsilon^+[q_i]$ be the set consisting of all states dominated by q_i in $D_\epsilon(M)$ and the state q_i itself. Then the states of M' are all subsets of Q, the start state of M' is $N_\epsilon^+[q_0]$, the accepting states of M' are all subsets P of Q such that $P \cap F \neq \emptyset$, and

$$\delta_{M'}(P, a) = \bigcup \{N_\epsilon^+[q_j] : \ q_j \in Q \text{ and } (q_i, a, q_j) \in \Delta \text{ for some } q_i \in P\}.$$

The above construction gives $2^{|Q|}$ states in M' (including $\emptyset$). In fact, we may well need less than $2^{|Q|}$ states in M' because all vertices of $D(M')$ not reachable from the initial vertex $N_\epsilon^+[q_0]$ can be deleted from $D(M')$ without changing $L(M')$. There are examples of non-deterministic finite automata M for which any equivalent deterministic finite automata must have exponential (in $|Q|$) number of states [533, 642].

17.13 Puzzles and Digraphs

There are several puzzles which can be solved using graph theory. Let us consider an example of such a puzzle various versions of which can be found in the literature and folklore. Suppose we have three buckets, X, Y and Z of volumes 1 liter, 3 liters and 5 liters, respectively. Suppose that bucket X is filled with water to its capacity, Y is empty and Z contains 4 liters of water. In each move, we are allowed to pour water from a bucket A to a bucket B ($A, B \in \{X, Y, Z\}$) such that in the end of the move either A is empty or B is filled to its capacity. We wish to determine the minimum number of moves required to get X empty, Y with 2 liters of water and Z with 3 liters of water.

To solve the puzzle we will use the digraph D depicted in Figure 17.13. Each vertex of D is denoted (i, j), where i is the amount of water in X and

j is the amount of water in Y (clearly, Z contains $5-i-j$ liters of water). The vertices (i,j) are states of the puzzle. In D, we have an arc from (i,j) to (i',j') if the move from (i,j) to (i',j') is allowed. When there are arcs in both directions between a pair of vertices, we replace the 2-cycle by an edge. Notice that D cannot be replaced by an undirected graph as there are arcs which are not in 2-cycles.

Clearly, to solve the puzzle it suffices to find a shortest path from $(1,0)$ to $(0,2)$. It is not difficult to see that a shortest such path is of length 3 and, thus, three is the minimum number of moves.

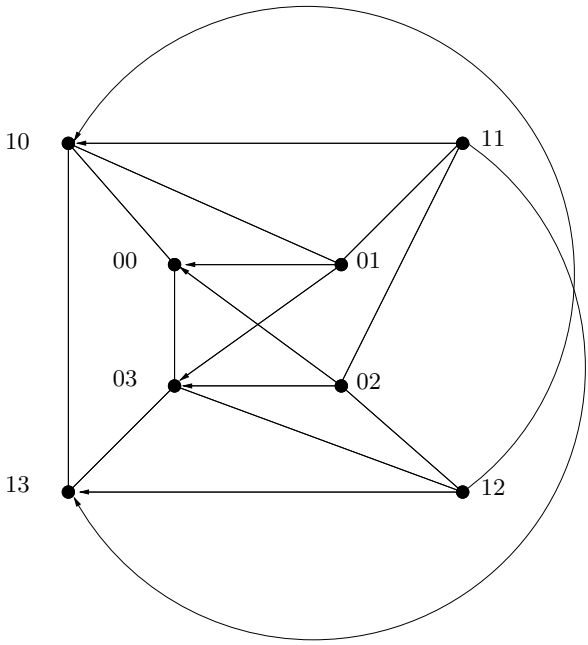


Figure 17.13 Digraph model of a puzzle. Here (i,j) is denoted by ij.

17.14 Gossip Problems

'There are n ladies, and each one of them knows an item of scandal which is not known to any of the others. They communicate by telephone, and whenever two ladies make a call, they pass on to each other, as much scandals as they know at the time. How many calls are needed before all ladies know every scandal?' This is the way the so-called gossip problem (apparently due to A. Boyd) was stated by Hajnal, Milner and Szemerédi [491] in 1972. Since

then numerous research papers on the topic have been published (see e.g. surveys by Fraigniaud and Lazard [333], Hedetniemi, Hedetniemi and Liestman [511], Hromkovič, Klasing, Monien and Peine [536]). The main reason for this popularity is a high applicability of the gossip problem, especially in computer networks.

Actually the above quotation captures only a special case of the gossip problem. In a more general setting, this problem can be formulated as follows. Let G be a connected graph of order n. Every vertex v of G holds initially an item $I(v)$ (different from the items of other vertices). A vertex v can pass all items it currently has to all or some of its neighbours at one step. The aim is to calculate the minimum number of steps required to pass to every vertex u the set $\{I(v) : \ v \in V(G)\}$ of all items.

The problem can be specified by allowing only one-way communications (like in radio communications over one frequency or email) when at every given step, for every pair u, v of adjacent vertices, either u can pass all items it holds to v, or v can pass all items it holds to u, but not both [333]. This specification is often called **half-duplex**. The half-duplex gossip problem is $\mathcal{NP}$-hard [333]. On the other hand, this problem is normally of interest, from the applications point of view, only for some special families of graphs such as the Cartesian products of cycles used in practice to build the Intel Δ-prototype (see Rattner [764]) and many transputer-based machines (see May [687]). Several important families of graphs are discussed by Fraigniaud and Lazard [333]. The solutions obtained for them are based on an upper bound that includes, as the main term, the minimum diameter of an orientation of a given undirected graph [333].

In the half-duplex gossip problem, we may consider symmetric digraphs $\overleftrightarrow{G}$ instead of undirected graphs G. The half-duplex model can be extended from symmetric to arbitrary digraphs D, where a vertex v can pass all its items only to vertices u such that vu is an arc in D. The use of arbitrary digraphs may well be of interest when security concerns dictate that some of the directions of communications are forbidden.

We consider only the half-duplex model for a strong digraph D. Let $s(D)$ stand for the minimum number of steps for gossiping in this model. Since the minimum number of steps to pass all items of vertex u to another vertex v is $\mathrm{dist}(u, v)$, we have $s(D) \geq \mathrm{diam}(D)$.

Gutin and Yeo (see [91]) proved the following simple upper bound on $s(D)$, which is an improvement on the similar upper bound in [333] even in the case of symmetric digraphs.

Theorem 17.14.1 *Let $D = (V, A)$ be a strong digraph. Then*

$$s(D) \leq \min\{2\,\mathrm{rad}(D), \mathrm{diam}_{\min}(D)\},$$

where $\mathrm{diam}_{\min}(D)$ is the minimum diameter of an orientation of D.

Proof: Let H be an orientation of D of minimum diameter. Let every vertex in D pass its items to all out-neighbours in H. Repeat this iteration till every

vertex holds all items. Clearly, the number of iterations required is the length of the longest path in H, i.e., $s(D) \leq \mathrm{diam}(H) = \mathrm{diam}_{\min}(D)$.

Let x be a vertex of D such that $\mathrm{rad}(D) = (\mathrm{dist}(x,V) + \mathrm{dist}(V,x))/2$. Let F_x^+ (F_x^-) be a BFS tree of D rooted at x (the converse of a BFS tree of the converse of D rooted at x). In the first $\mathrm{dist}(V,x)$ steps pass items from vertices to their out-neighbours along arcs of F_x^-. Thus, in the end, x holds all items. During the next $\mathrm{dist}(x,V)$ steps pass items from vertices to their out-neighbours along arcs of F_x^+. Hence, in the end, every vertex holds all items. Thus, $s(D) \leq 2\,\mathrm{rad}(D)$. □

The bound of Theorem 17.14.1 is of special interest when D satisfies $\mathrm{diam}(D) = \mathrm{diam}_{\min}(D)$. In this case, a minimum diameter orientation of D provides an optimal solution to the gossip problem. Thus, an orientation H of diameter possibly exceeding $\mathrm{diam}(D)$ by a small constant leads to a good approximate solution for the gossip problem (see Section 3.6).

17.15 Deadlocks of Computer Processes

The (computer) process is a fundamental concept in all operation systems [841]. A process is just an executing computer program including the current values of registers, variables and program counter. To complete its task each process requests some resources such as printers, scanners, plotters and files. Sometimes, a process cannot receive the requested resource. For example, suppose that two processes, p_1 and p_2, are working as photocopiers: each of them scans and prints documents; when a process works with the scanner or printer, it requires exclusive access to it. Each of the processes works as follows: it grabs the scanner and the printer, scans the document and prints it immediately (the data from the scanner are not stored but sent to the printer directly).

Suppose also that the two processes are programmed in the following unfortunate way: A first requests the printer and then the scanner, but B first requests the scanner and then the printer. Suppose that A has requested the printer and by the time it requests the scanner, it has been assigned to B. Then we have **deadlock**, i.e., none of the two processes can complete its task.

Let us first consider the case when each resource has only one copy (i.e., there is only one printer, scanner, etc.). To check whether process deadlock will occur, one can use the following bipartite digraph model D [841]: Let P and R be the sets of processes and resources, respectively; P and R are bipartition of D. For resource $r \in R$ and process $p \in P$, bipartite digraph D has an arc rp when p holds r and D has an arc pr when p requests r. Clearly, we have no deadlock if and only if D is acyclic. Using Proposition 2.1.5, we can check whether D is acyclic in time $O(n+m)$ ($n = |P|+|R|$ and $m = |A(D)|$).

Algorithm DFSA from Section 2.1 gives an acyclic ordering of $V(D)$. Restricting this ordering to P, we get $p_1, p_2, \ldots, p_s$. Notice that the resources will always be available for the current process, if we execute the processes as follows: $p_s, p_{s-1}, \ldots, p_1$. This ordering of P will allow the processor not to move from one process to another without executing the former due to non-availability of the requested resources.

Suppose D is not acyclic. Then it is natural to find the minimal number of processes whose deletion will eliminate all directed cycles of D (to resolve the deadlock optimally). We will call this problem the Optimal Deadlock Resolution Problem (ODR). Observe that ODR is $\mathcal{NP}$-hard due to the $\mathcal{NP}$-hardness of the Feedback Vertex Set Problem (FVS) (i.e., the problem of finding the minimum number of vertices in a digraph H whose deletion will make H acyclic; for more details, see Section 15.3). Indeed, let H be an instance of FVS and let us split every vertex of H to obtain a new digraph D. Recall that D has two vertices x_1, x_2 instead of every vertex x of H, $x_1 \rightarrow x_2$ in D for each $x \in V(H)$ and $x_2y_1 \in A(D)$ if and only if $xy \in A(H)$. Now assign x_1 to R and x_2 to P for each $x \in V(H)$. Then D is an instance of the processes-resources model considered above (each resource x_1 is given to just one process x_2). Clearly, a set X is an optimal solution of FVS on H if and only if $\{x_2 : x \in X\}$ is an optimal process deletion.

It is not difficult to prove that (the following natural parameterization of) ODR is fixed-parameter tractable: check whether deletion of k vertices in P from D makes D acyclic. Indeed, consider an extension[7] D^* of D in which each vertex in R is replaced by $k+1$ independent vertices. Clearly, there is a set $X \subset V(D^*)$ of size at most k such that $D^* - X$ is acyclic if and only if $D - X$ is acyclic. Also, we may assume that $X \subseteq P$. Since FVS is fixed-parameter tractable (see Section 15.3), we conclude that so is ODR.

If there are several copies of the same resource (e.g., two or more printers), the situation with deadlocks becomes more complicated. Consider the following simple example: Let $V(D) = \{p_1, p_2, r_1, r_2', r_2''\}$ (where r_2' and r_2'' are copies of the same resource r_2) and $A(D) = \{r_1p_1, p_1r_2', p_1r_2'', p_2r_1, r_2''p_2\}$. Suppose that p_1 requests only one copy of r_2. Observe that D has a directed cycle, but there is no deadlock: first p_1 uses resources r_1 and r_2' and releases them such that p_2 can use r_1 (together with r_2'') afterwards.

The deadlock detection problem, even for the case when processes may need several copies of the same resource, is polynomial time solvable [841]: at each iteration we simply find a process that can have all requested resources. Such a process is executed and afterwards it releases all resources it has used. If, as a result, all processes have been executed, there is no deadlock. Otherwise, there is a deadlock. It is easy to show that this algorithm is correct: observe that the set of the processes which can be executed or have been executed does not decrease after each iteration and, thus, it does not

[7] For the definition of a digraph extension, see Section 1.3.

matter which one of the possible processes is executed first. However, the following problem remains open:

Problem 17.15.1 *Is the ODR problem fixed-parameter tractable when each resource may have multiple copies and each process may use several copies of the same resource?*

17.16 Exercises

17.1. Show that Theorem 17.3.3 is not true for $r \leq \lceil \frac{n-1}{2} \rceil$. (Bang-Jensen, Gutin and Yeo [99])

17.2. Prove Theorem 17.2.2. Hint: Assume that D has a path of length 2 and prove that $\mathrm{cc}(D) \leq f(n)$. (Gutin and Yeo [481])

17.3. Prove Lemma 17.2.4.

17.4. Construct an alternating hamiltonian cycle in the 2-edge-coloured graph of Figure 17.6.

17.5. (−) Check which of the following 4×4 matrices $A = [a_{ij}]$ have hidden block-triangular structure (the entries not specified equal zero). Only simultaneous permutations of rows and columns are allowed.
(a) $a_{1i} = i + 1$ for $i = 1, 2, 3$, $a_{2i} = a_{3i} = i$ for $i = 2, 3$ and $a_{4i} = 2$ for $i = 2, 3, 4$;
(b) $a_{12} = a_{21} = a_{14} = a_{41} = a_{34} = a_{43} = 2$ and $a_{ii} = 1$ for $i = 1, 2, 3, 4$.

17.6. (−) Prove that the Balcer-Veinott algorithm (in Section 17.8) terminates with a cycle, whose length is the period of the input digraph.

17.7. (−) Prove that the period of a strong non-bipartite digraph D with $\delta^0(D) \geq 3$ equals 1. Hint: use Theorem 8.3.7.

17.8. (−) Give a construction of a Latin square of size n for each integer $n \geq 1$.

17.9. (+) Prove Theorem 17.9.2.

17.10. Construct a bipartite graph $B = (X \cup Y, E)$ with a family $\{>_u | u \in X \cup Y\}$ of orderings induced from local orderings of the neighbours of each vertex, such that no maximum matching of B is stable.

17.11. (−) Argue that the oriented graph D in the proof of Theorem 17.9.3 is $(n-1)$-regular.

17.12. Prove Theorem 17.12.1.

17.13. (−) For an automaton A depicted in Figure 17.12, show that $L(A) = ab^*(aa^*bb^*)^*|ba^*b(aa^*bb^*)^*$.

17.14. For the non-deterministic automaton B depicted in Figure 17.12 (right), construct a deterministic automaton B' such that $L(B) = L(B')$.

18. Algorithms and Their Complexity

In this book we often describe and analyze algorithms on digraphs. We concentrate more on graph-theoretical aspects of these algorithms than on their actual implementation on a computer. Thus, in many cases only the most basic knowledge on algorithms and complexity is required and many readers are familiar with it. However, sometimes we use less familiar terminology and notation. In particular, we sometimes say that some problem is fixed-parameter tractable or W[1]-hard.

The objective of this chapter is to help the reader to refresh basic knowledge on data structures and algorithms and give a brief introduction to less familiar areas of algorithmics which are used in this book. We also briefly discuss the satisfiability problem and its important subproblems as well as matroids.

It is well-known that most computational optimization problems are $\mathcal{NP}$-hard. Thus, we cannot hope to design polynomial algorithms for them and should try to find ways around the intractability. In fact, there are a number of ways around the intractability. They can be divided into two categories. The approaches of the first category try to solve interesting instances of the problem under consideration to optimality either by considering only polynomial-time-solvable special subproblems in the hope that the instances belong to these subproblems (we consider many polynomial-solvable subproblems in this book), or by parameterizing the problem in such a way that the computational complexity is mainly confined to some parameter and the parameter is relatively small (for more details, see Section 18.4) or by using exponential-time algorithms in the hope that certain instances of the problem can be solved relatively quickly (see Section 18.5).

The approaches of the first category have proved to be very useful, but they are somewhat unreliable in terms of running time especially for practical problems and, thus, approaches of the second category are of interest. Here we relax the optimality condition and consider approximation algorithms (see Section 18.6) or their much less theoretically based, yet much more widely used cousins, heuristics (see Section 18.7).

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 695
Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_18,

18.1 Combinatorial Algorithms

Recall that unless specified otherwise, n (m) denotes the number of vertices (arcs) in the directed multigraph under consideration. In the following, all logarithms whose base is unspecified are of base 2. For a pair of given functions $f(k), g(k)$ of a non-negative integer argument k, we say that $f(k) = O(g(k))$ if there exist positive constants c and k_0 such that $0 \leq f(k) \leq cg(k)$ for all $k \geq k_0$. If there exist positive constants c and k_0 such that $0 \leq cf(k) \leq g(k)$ for all $k \geq k_0$, we say that $g(k) = \Omega(f(k))$. Clearly, $f(k) = O(g(k))$ if and only if $g(k) = \Omega(f(k))$. If both $f(k) = O(g(k))$ and $f(k) = \Omega(g(k))$ hold, then we say that $f(k)$ and $g(k)$ are of the **same order** and denote it by $f(k) = \Theta(g(k))$.

In the analysis of an algorithm, first of all we will be interested in its **time complexity** which must reflect the running time of the corresponding computer program on various computers. In order to make the time complexity measure sufficiently universal, it is usually assumed that computations are performed by some abstract computer. The computer executes **elementary operations**, that is, arithmetical operations, comparisons, data movements and control branching, each in constant time. Since we are interested only in the asymptotics of the execution time, the number of elementary operations of an algorithm will be considered as its time complexity. In the vast majority of cases, the time complexity (which will often be called just the **complexity**) of an algorithm depends on the size of its input. An algorithm $\mathcal{A}$ is an $O(g(n))$ algorithm for some function $g(n)$ of its input size if the running time of $\mathcal{A}$ on inputs of size n never exceeds $cg(n)$ for some constant c (depending only on $\mathcal{A}$).

Since the typical inputs to the algorithms considered in this book are (weighted) directed multigraphs, the size of inputs will be measured by the numbers of vertices and arcs, that is, by n and m, and, for digraphs with weights on the arcs (vertices), by $\log |c_{\max}|$, where $|c_{\max}|$ is the maximum of the absolute values of the weights of arcs (vertices). An algorithm of complexity $O(p(n, m, \log |c_{\max}|))$, where $p(n, m, \log |c_{\max}|)$ is a polynomial in n, m and $\log |c_{\max}|$, is a **polynomial-time** (or just **polynomial**) algorithm. The notion of equating efficient algorithms with polynomial algorithms is due to Edmonds [282] and is at present the most popular formalization for the intuitive notion of 'efficient' algorithms. Although we would normally not call an algorithm of complexity $\Theta(n^{1000})$, where n is the size of the input, an efficient algorithm, it is very rarely the case that polynomial algorithms have such a high degree of their associated polynomials.

There are two well-known and often-used ways to represent a digraph $D = (V, A)$ for computational purposes: as a collection of adjacency lists and as an adjacency matrix.

For the **adjacency matrix representation** of a directed multigraph $D = (V, A)$, we assume that the vertices of D are labelled $v_1, v_2, \ldots, v_n$ in some arbitrary but fixed manner. The **adjacency matrix** $M(D) = [m_{ij}]$ of

a digraph D is an $n \times n$ matrix such that $m_{ij} = 1$ if $v_i \rightarrow v_j$ and $m_{ij} = 0$ otherwise. For directed pseudographs we let $m_{ij} = \mu(v_i, v_j)$, that is, m_{ij} is the number of arcs from v_i to v_j. The adjacency matrix representation is a very convenient and fast tool for checking whether there is an arc from a vertex to another one. A drawback of this representation is the fact that to check all adjacencies, without using any other information besides the adjacency matrix, one needs $\Omega(n^2)$ time. Thus, the majority of algorithms using the adjacency matrix cannot have complexity lower than $\Omega(n^2)$ (this holds in particular if we include the time needed to construct the adjacency matrix).

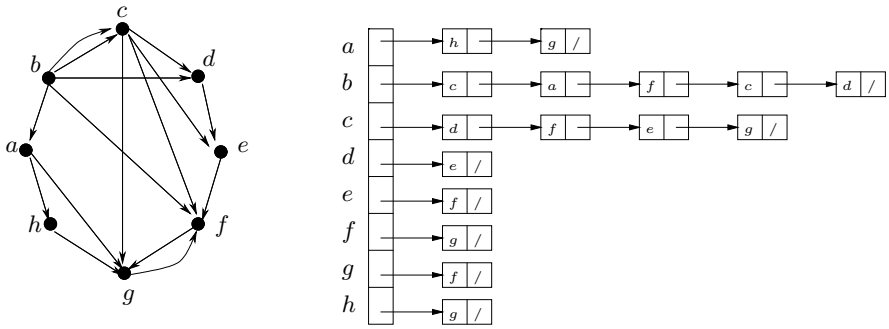


Figure 18.1 A directed multigraph and a representation by adjacency lists Adj^+.

The **adjacency list representation** of a directed pseudograph $D = (V, A)$ consists of a pair of arrays Adj^+ and Adj^-. Each of Adj^+ and Adj^- consists of $|V|$ (linked) lists, one for every vertex in V. For each $x \in V$, the linked list $Adj^+(x)$ ($Adj^-(x)$, respectively) contains all vertices dominated by x (dominating x, respectively) in some fixed order (see Figure 18.1). Using the adjacency list $Adj^+(x)$ ($Adj^-(x)$) one can obtain all out-neighbours (in-neighbours) of a vertex x in $O(|Adj^+(x)|)$ ($O(|Adj^-(x)|)$) time. A drawback of the adjacency list representation is the fact that one needs, in general, more than constant time to verify whether $x \rightarrow y$. Indeed, to decide this we have to search sequentially through $Adj^+(x)$ (or $Adj^-(x)$) until we either find y (x) or reach the end of the list.

To illustrate the concepts described in this section, let us consider the Hamilton path problem in tournaments. Theorem 1.4.2 states that every tournament is traceable. However, the proof that we have presented is non-constructive, i.e., it does not provide us with a polynomial algorithm to find a Hamilton path in a tournament. Now we give two constructive proofs of Theorem 1.4.2 and show how these lead to polynomial algorithms to construct a Hamilton path in a tournament.

Inductive Proof of Theorem 1.4.2: Clearly, the one-vertex tournament has a Hamilton path (the vertex itself). Assume that the theorem holds for

every tournament with less than $n(\geq 2)$ vertices. Consider a tournament T with n vertices and a vertex $x \in V(T)$. By induction, the tournament $T - x$ has a Hamilton path, $P = y_1 y_2 \dots y_{n-1}$. If $x \to y_1$, then xP is a Hamilton path in T; if $y_{n-1} \to x$, then Px is a Hamilton path in T. Assume that $y_1 \to x$ and $x \to y_{n-1}$. Then, it is easy to show that there exists an index $i < n-1$ such that $y_i \to x$ and $x \to y_{i+1}$. Thus, $P[y_1, y_i]xP[y_{i+1}, y_{n-1}]$ is a Hamilton path in T. □

This constructive proof gives rise to the following simple algorithm to find a Hamilton path in a tournament.

HamPathTour
Input: A tournament T on n vertices labelled $x_1, x_2, \dots, x_n$ and its adjacency matrix $M = [m_{ij}]$.
Output: A Hamilton path in T.

1. Let $P := x_1$ and $i := 2$.
2. If $i > n$ go to Step 7.
3. Let $P = y_1 y_2 \dots y_{i-1}$ be the current path.
4. If $x_i \to y_1$ then $P := x_i P$. Let $i := i + 1$ and go to Step 2.
5. If $y_{i-1} \to x_i$ then $P := Px_i$. Let $i := i + 1$ and go to Step 2.
6. For $j = 1$ to $i - 2$ do: If $y_j \to x_i \to y_{j+1}$ then $P := P[y_1, y_j]x_i P[y_{j+1}, y_{i-1}]$. Let $i := i + 1$ and go to Step 2.
7. Return the path P.

The correctness of this algorithm follows from the above proof. To see that this algorithm can be implemented as an $O(n^2)$ algorithm, observe that the algorithm has two nested loops, each of which performs $O(n)$ operations (we count queries to the adjacency matrix as constant time) and all other steps take constant time. Thus, the complexity is $O(n^2)$.

The reader who is familiar with algorithms for sorting numbers might have noticed that HamPathTour is very similar to the algorithm *Insertion-Sort* which sorts numbers by inserting one at a time in a list (see e.g. [231, pp. 2-4]). This resemblance is no coincidence. In fact, given any set $\mathcal{S} = \{a_1, \dots, a_n\}$ of n distinct real numbers we can form an acyclic tournament $T(\mathcal{S})$ with $V(T(\mathcal{S})) = \mathcal{S}$ and $A(T(\mathcal{S})) = \{a_i a_j : a_i < a_j, 1 \leq i \neq j \leq n\}$. The correct (sorted) increasing order on $\mathcal{S}$ corresponds to the unique Hamilton path $a_{\pi(1)} a_{\pi(2)} \dots a_{\pi(n)}$ of $T(\mathcal{S})$ which again is the unique acyclic ordering of $V(T(\mathcal{S}))$ (see also Exercise 2.1). Thus any algorithm for finding a Hamilton path in a tournament can be used for sorting numbers (we compare numbers, by looking at the orientation of the arc between the corresponding vertices in[1] $T(\mathcal{S})$). Conversely, several sorting algorithms can be translated into algorithms for solving the more general problem of finding Hamilton paths in

[1] Note that this is only a virtual description, since we do not need to construct the adjacency matrix in this case. We simply compare the two numbers x and y and $x \to y$ holds if and only if $x < y$.

tournaments. One such example is the classical **Mergesort** algorithm (see e.g.[232, pp. 28-36]), which we now translate into the language of tournaments. For simplicity we shall assume that the number of vertices of the input tournament is a power of two. The reader can easily extend the algorithm to the general case, see Exercise 18.1. It is convenient to state the algorithm as a recursive algorithm (which is the reason why we specify a parameter for the algorithm). We assume that the tournament is available through its adjacency matrix.

MergeHamPathTour(T)

1. Split T into two tournaments T_1 and T_2 on the same number of vertices.
2. P_i:= **MergeHamPathTour**(T_i), $i = 1, 2$.
3. P:= **MergePaths**(P_1, P_2).
4. Return P.

Here **MergePaths** is a procedure which, given two disjoint paths P, P' in tournament T, merges these two into one path P^* such that $V(P^*) = V(P) \cup V(P')$. This can be done in the same way as one would merge two sorted lists of numbers into one sorted list.

Procedure **MergePaths**(P, P')
Input: Paths $P = x_1x_2 \dots x_k$ and $P' = y_1y_2 \dots y_r$.
Output: A path P^* such that $V(P^*) = V(P) \cup V(P')$.

1. If P' is empty then P^*:=P.
2. If P is empty then P^*:= P'.
3. If x_1 dominates y_1 then P^*:=x_1**MergePaths**$(P - x_1, P')$.
4. If y_1 dominates x_1 then P^*:=y_1**MergePaths**$(P, P' - y_1)$.
5. Return P^*.

The classical analysis of the Mergesort algorithm (see e.g. [231]) shows that the algorithm uses $O(n \log n)$ comparisons to sort n real numbers. Similarly it follows from our description above that the algorithm MergeHamPathTour will find a Hamilton path in a tournament T with n vertices after making $O(n \log n)$ queries about adjacencies of vertices in T. Note that to implement the algorithm we do not need to construct the adjacency matrices of each of the tournaments considered in the recursive calls. Indeed, all adjacencies can be checked using the adjacency matrix of the original tournament. Hence, if we only count the number of times we need to check the direction of an arc, then MergeHamPathTour is a faster algorithm than HamPathTour.

For more details on design and analysis of combinatorial algorithms, the reader is directed to numerous books on the subject, e.g., to Aho, Hopcroft and Ullman [11], Brassard and Bratley [180], Cormen, Leiserson, Rivest and Stein [232] and Kleinberg and Tardos [599].

18.2 $\mathcal{NP}$-Complete and $\mathcal{NP}$-Hard Problems

There are many interesting algorithmic problems concerning (di)graphs for which no polynomial algorithm is known. Many of those problems (formulated in their decision form) belong to the class $\mathcal{NPC}$ of so-called **$\mathcal{NP}$-complete** problems. For a detailed introduction to the class of $\mathcal{NP}$-complete problems, see the book by Garey and Johnson [393]. A problem is a **decision** problem if it requires the answer 'yes' or 'no'. By a **problem** we understand actually a family of instances. For example, we will consider the Hamilton Cycle Problem in a digraph: given a digraph, decide whether or not it has a Hamilton cycle. Every digraph provides an **instance** of this problem. The so-called Traveling Salesman Problem (TSP) is similar: given a weighted complete digraph D and a real number B, decide whether D contains a Hamilton cycle of weight at most B. An instance of the last problem consists of a complete digraph and a specification of the weights of its arcs. For theoretical and algorithmic results on TSP and its variations, see, e.g., an edited volume [484].

A decision problem $\mathcal{S}$ belongs to the complexity class $\mathcal{P}$ if and only if there exists a polynomial algorithm $\mathcal{A}$ which, given any instance of $\mathcal{S}$, produces an answer in the set {'yes','no'} such that the answer from $\mathcal{A}$ on input x is 'yes' if and only if x is a 'yes' instance for[2] $\mathcal{S}$. Since $\mathcal{A}$ is polynomial, it follows that it produces its answer after at most $p(|x|)$ steps, where $|x|$ is the size of the input x and p is a fixed polynomial (depending on $\mathcal{S}$).

A decision problem belongs to the class $\mathcal{NP}$ (co-$\mathcal{NP}$) if, for every 'yes' instance ('no' instance) of the problem, there exists a short 'proof', called a **certificate**, of polynomial size (in n, m and $\log |c_{max}|$) such that, using the certificate, one can verify in polynomial time that the instance is indeed a 'yes' ('no') instance. The certificate depends on the instance of the problem, but it must have the same structure for all instances of the problem. To illustrate this definition, let us show that both the Hamilton cycle problem and traveling salesman problem are in $\mathcal{NP}$. Given a permutation π of the vertices in a digraph D (π is the certificate for hamiltonicity of D), it is easy to verify whether this permutation corresponds to a Hamilton cycle in D (note that this certificate has the same structure for each instance of the problem, namely, it is a permutation of the vertices). Indeed, assuming that $V(D) = [n]$, we simply have to check that $\pi(i)\pi(i+1)$ is an arc of D for every $i \in [n]$, where the vertex $n+1$ is the same as the vertex 1. If we also have weights on the arcs, then it is easy to verify that the weight of the proposed Hamilton cycle is no more than B. Notice that the situation here is not symmetric: it is unknown if the 'complement' of the Hamilton cycle problem (given a digraph, check whether it has no Hamilton cycle) is in $\mathcal{NP}$. Indeed, it is difficult to imagine what kind of certificate will enable a

[2] Thus a hypothetical polynomial algorithm for the Hamilton cycle problem must produce the answer 'yes' precisely when the input digraph has a Hamilton cycle.

polynomial algorithm to check that a digraph is not hamiltonian. Actually, such a certificate would answer in the affirmative the well-known complexity question: whether $\mathcal{NP} = \text{co-}\mathcal{NP}$ (see e.g. [393, Theorem 7.2]). A positive answer to this question seems to be unlikely with our current knowledge of algorithms.

Given a pair of decision problems $\mathcal{S}$, $\mathcal{T}$, we say that $\mathcal{S}$ is **polynomially reducible** to $\mathcal{T}$ (denoted $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$) if there is a polynomial algorithm $\mathcal{A}$ that transforms an instance x of $\mathcal{S}$ into an instance $\mathcal{A}(x)$ of $\mathcal{T}$ such that the second instance has the same answer as the first one. That is, x is a 'yes' instance of $\mathcal{S}$ if and only if $\mathcal{A}(x)$ is a 'yes' instance of $\mathcal{T}$. Some polynomial reductions are quite easy. For example, we can readily reduce the Hamilton cycle problem to the traveling salesman problem: given a digraph D consider a copy of a $\overleftrightarrow{K}_n$ such that $V(D) = V(\overleftrightarrow{K}_n)$, and, for every arc e in $\overleftrightarrow{K}_n$, its weight is 1 if $e \in A(D)$ and 2 otherwise. Let also $B = n$. Clearly, D is hamiltonian if and only if with the prescribed weights $\overleftrightarrow{K}_n$ has a Hamilton cycle of weight not exceeding B. Obviously, the above transformation can be carried out by a polynomial algorithm.

A decision problem is **$\mathcal{NP}$-hard** if all problems in $\mathcal{NP}$ can be polynomially reduced to this problem. If the problem is $\mathcal{NP}$-hard and also belongs to $\mathcal{NP}$, then it is **$\mathcal{NP}$-complete**. The class $\mathcal{NPC}$ consists of all $\mathcal{NP}$-complete problems. In order to show that a decision problem $\mathcal{W}$ is $\mathcal{NP}$-hard, we must show that every problem in $\mathcal{NP}$ can be polynomially reduced to $\mathcal{W}$ – a seemingly impossible task. However, polynomial transformations are closed under composition, that is, $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$ and $\mathcal{T} \leq_{\mathcal{P}} \mathcal{K}$ implies that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{K}$ (see Exercise 18.4). Hence, in order to prove that W is $\mathcal{NP}$-hard, it suffices to prove that there is *some* $\mathcal{NP}$-complete problem which is polynomially reducible to $\mathcal{W}$ (see Exercise 18.6). Of course this only works if we already have established that there is some problem that belongs to the class $\mathcal{NPC}$ of $\mathcal{NP}$-complete problems. This extremely important and non-trivial step was provided by Cook in 1971 [228] (independently, a similar discovery was made by Levin [640]).

Since there are a huge number of known $\mathcal{NP}$-complete problems, the task to prove that a given problem is $\mathcal{NP}$-complete is sometimes not too difficult. On the other hand, it is also highly non-trivial in many cases. We will give a number of examples of $\mathcal{NP}$-completeness and $\mathcal{NP}$-hardness proofs throughout this book. It is well-known that the Hamilton cycle problem is $\mathcal{NP}$-complete as shown by Karp in his classical paper [585]. It follows from the above transformation that the traveling salesman problem is $\mathcal{NP}$-complete as well.

Quite often we will deal with **optimization problems** rather than decision problems. Since an optimization problem consists of finding an optimal solution to a prescribed problem, such a problem very often has a decision analogue. For example, in the usual formulation of the traveling salesman problem the goal is to find a minimum weight Hamilton cycle in a weighted

complete digraph. The decision analogue was stated above. If the decision analogue of an optimization problem is $\mathcal{NP}$-hard, then we will also say that the optimization problem is $\boldsymbol{\mathcal{NP}}$**-hard**. So, the optimization version of the traveling salesman problem is $\mathcal{NP}$-hard. For a wealth of information on $\mathcal{NP}$-hard optimization problems and their approximability properties, see, e.g., the books [53] by Ausiello, Crescenzi, Gambosi, Kann, Marchetti-Spaccamela and Protasi and [884] by Vazirani.

From a complexity point of view, there is no significant difference between a decision problem and its optimization analogue (if it exists). To illustrate this statement, let us consider the problem of deciding whether a strong digraph has a cycle of length at least k (here k is part of the input). The optimization analogue is the problem of finding a cycle of maximum length in a strong digraph. If we solve the optimization problem, we easily obtain a solution to the decision problem: just check whether the length of the longest cycle is at least k. On the other hand, using binary search one can find an answer to the optimization problem by solving a number of decision problems. In our example, we first check whether or not the digraph under consideration has a cycle of length at least $n/2$. Then, solve the analogous problem with $n/4$ (if D has no cycle of length at least $n/2$) or $3n/4$ (if D has a cycle of length at least $n/2$) instead of $n/2$, etc. So, we would need to solve $O(\log n)$ decision problems, in order to obtain an answer to the optimization problem.

18.3 The Satisfiability Problem

In this subsection we deal with a problem called the **Satisfiability Problem** or, briefly, **SAT** that is not a problem on digraphs, but it has applications to many problems on graphs including several ones given in the book. SAT is a classical NP-complete problem and contains subproblems 3-SAT and 2-SAT defined later. Both subproblems are often used to establish computational complexity of problems on graphs and other objects: While 3-SAT is NP-complete and is very often used to prove that a certain problem is also NP-complete, 2-SAT is polynomial time solvable and there are many examples where 2-SAT is applied to show that a certain problem admits a polynomial algorithm as well. In Chapter 17, we prove that 2-SAT is polynomial time solvable, and in Chapter 11, we give examples of how 2-SAT can be used to prove that a certain problem is polynomial time solvable.

A **boolean variable** x is a variable that can assume only two values 0 and 1. The **sum** of boolean variables $x_1 + x_2 + \ldots + x_k$ is defined to be 1 if at least one of the x_i's is 1 and 0 otherwise. The **negation** $\overline{x}$ of a boolean variable x is the variable that assumes the value $1 - x$. Hence $\overline{\overline{x}} = x$. Let X be a set of boolean variables. For every $x \in X$ there are two **literals**, over x, namely, x itself and $\overline{x}$. A **clause** C over a set of boolean variables X is a sum of literals over the variables from X. The **size** of a clause is the number of literals it contains. For example, if u, v, w are boolean variables with values

$u = 0, v = 0$ and $w = 1$, then $C = (u + \overline{v} + \overline{w})$ is a clause of size 3, its value is 1 and the literals in C are u, $\overline{v}$ and $\overline{w}$. An assignment of values to the set of variables X of a boolean expression is called a **truth assignment**. If the variables are $x_1, \ldots, x_k$, then we denote a truth assignment by $t = (t_1, \ldots, t_k)$. Here it is understood that x_i will be assigned the value t_i for $i \in [k]$.

The **satisfiability problem**, also called **SAT**, is the following problem. Let $X = \{x_1, \ldots, x_k\}$ be a set of boolean variables and let $C_1, \ldots, C_r$ be a collection of clauses for which every literal is over X. Decide if there exists a truth assignment $t = (t_1, \ldots, t_k)$ to the variables in X such that the value of every clause will be 1. This is equivalent to asking whether or not the boolean expression $\mathcal{F} = C_1 * \ldots * C_p$ can take the value 1. Depending on whether this is possible or not, we say that $\mathcal{F}$ is **satisfiable** or **unsatisfiable**. Here '$*$' stands for **boolean multiplication**, that is, $1*1 = 1$, $1*0 = 0*1 = 0*0 = 0$. For a given truth assignment $t = (t_1, \ldots, t_k)$ and literal q we denote by $q(t)$ the value of q when we use the truth assignment t (i.e., if $q = \overline{x_3}$ and $t_3 = 1$, then $q(t) = 1 - 1 = 0$).

To illustrate the definitions, let $X = \{x_1, x_2, x_3\}$ and let $C_1 = (\overline{x_1} + \overline{x_3})$, $C_2 = (x_2 + \overline{x_3})$, $C_3 = (\overline{x_1} + x_3)$ and $C_4 = (x_2 + x_3)$. Then it is not difficult to check that $\mathcal{F} = C_1 * C_2 * C_3 * C_4$ is satisfiable and that taking $x_1 = 0, x_2 = 1, x_3 = 1$ we obtain $\mathcal{F} = 1$.

In this example every clause is of size 2, so the instance of SAT given in the example is also an instance of 2-SAT: **2-SAT** is SAT in which every clause must be of size 2. If we require that every clause in SAT has size exactly 3, we obtain **3-SAT**. We have already mentioned that while 2-SAT is polynomial time solvable, 3-SAT is $\mathcal{NP}$-complete. For more details, see the book by Garey and Johnson [393].

Consider an instance $\mathcal{F}$ of 2-SAT. It is interesting to note that if, instead of asking whether $\mathcal{F}$ is satisfiable, we ask whether there exists some truth assignment such that at least ℓ clauses will get the value 1, then this problem, which is called **MAX-2-SAT**, is $\mathcal{NP}$-complete as shown by Garey, Johnson and Stockmeyer [394] (here ℓ is part of the input for the problem).

18.4 Fixed-Parameter Tractability and Intractability

Fixed-Parameter Algorithmics (FPA) is a relatively new approach for dealing with intractable computational problems. In the framework of FPA we try to introduce a parameter k, which is often a positive integer (but may be a vector, graph or any other object for some problems) such that the problem at hand can be solved in time $O(f(k)n^c)$, where n is the size of the problem instance, c is a constant not dependent on n or k, and $f(k)$ is an arbitrary computable function not dependent on n. The ultimate goal is to obtain $f(k)$ and c such that for small or even moderate values of k the problem under consideration can be completely solved in a reasonable amount of time.

As an example, consider the **Vertex Cover problem (VC)**: given an undirected graph G (with n vertices and m edges), find a minimum number of vertices such that every edge is incident to at least one of these vertices. VC with parameter k, (an upper bound on) the number of vertices in a vertex cover, admits an algorithm of running time $O(1.2738^k + kn)$ obtained by Chen, Kanj and Xia [204] that allows us to solve VC with k up to several hundred. Without using FPA, we would be likely to end up with the obvious algorithm of complexity $O(mn^k)$. This algorithm is far too slow even for small values of k such as $k = 10$.

Parameterized problems that admit algorithms of complexity $O(f(k)n^c)$ are called **fixed-parameter tractable (FPT)**. Not all parameterized problems are FPT. Indeed, k-COL, the problem of checking whether a graph G has a proper colouring with at most k colours, cannot be FPT as k-COL is NP-complete for each fixed $k \geq 3$. However, there are many parameterized problems admitting an algorithm of complexity $O(n^{f(k)})$, but very likely not being FPT. A typical example is k-IS, the problem of checking whether a graph G has an independent set with at least k vertices. k-IS is proved (see, e.g., the book [274] by Downey and Fellows) to be in a special wide class W[1] of parameterized problems and k-IS is among the hardest problems in W[1], namely, k-IS is W[1]-complete. This means that every parameterized problem in W[1] can be reduced to k-IS.

We do not define the class W[1] and the above-mentioned reduction as we will not use them in this book (for these and other definitions and a large number of results on FPA, see the monographs [274, 325, 727] by Downey and Fellows, Flum and Grohe, and Niedermeier, respectively). Instead, we would like to convey the following important message: we know that it is very unlikely that $\mathcal{P} = \mathcal{NP}$, it is also highly unlikely that all parameterized problems in W[1] are FPT. Thus, the fact that a certain problem is W[1]-complete implies that it is highly unlikely that the problem is FPT. To this end, it is sufficient to know that the parameterized problem under consideration is W[1]-hard, i.e., every problem in W[1] can be reduced to the problem.

We have a good understanding of many parameterized problems on undirected graphs. However, research of FPA on digraphs is less advanced. However, recently several papers on the topic have been written: (i) Gutin and Yeo [482] provided an overview of the topic, (ii) Gutin, Kloks, Lee and Yeo [466] proved that the problem of checking whether a digraph has a kernel with at most k vertices is FPT when restricted to planar digraphs and W[1]-hard (in fact, W[2]-hard) for general digraphs, (iii) Chen, Liu, Lu, O'Sullivan and Razgon [205] showed that the problem of checking whether a digraph has a feedback vertex set of size k is FPT (see Subsection 15.3.4 for details), (iv) Alon, Fomin, Gutin, Krivelevich and Saurabh [22] proved that the problem of verifying whether a digraph has an out-tree with at least k leaves (i.e., vertices of out-degree zero) is FPT, (v) Bonsma and Dorn [173] (Gutin,

Razgon and Kim [472], respectively) showed that the problems of checking whether a digraph has an out-branching with at least k leaves (with at least k non-leaves) is FPT and (vi) Bang-Jensen and Yeo [122] proved that the problem of verifying where a strong digraph of order n has a strong spanning subdigraph of size at most $2(n-1)-k$ is FPT. The results obtained in [22] and [173] were improved by Bonsma and Dorn [174].

FPA provides us with some practical algorithms to solve $\mathcal{NP}$-hard problems (e.g., in bioinformatics [727]), a significantly more refined view of the class $\mathcal{NPC}$ (the classes W[1], W[2], etc.) and some theoretical basis of preprocessing, a tool important for practical exact algorithms and heuristics.

18.5 Exponential Algorithms

In the VC problem given in Section 18.4, the parameter $k \leq n$. Thus, the complexity $O(1.2738^k + kn)$ of the algorithm by Chen, Kanj and Xia [204] can be bounded by $O(1.2738^n)$ giving us an exponential-time algorithm for VC, which is significantly faster than the obvious $O(2^n)$-time algorithm. Of course, if we could further reduce the time complexity to say $O(1.1^n)$, we would have an algorithm that may well be more efficient than, say, an $O(n^5)$-time algorithm even for moderate values of n. This already indicates that research into exponential-time algorithms is not a purely theoretical exercise. The first interesting theoretical results are overviewed in the survey paper [906] of Woeginger and there are several interesting further results, see, e.g., the papers [130] by Beigel and Eppstein and [238] by Dahllöf, Jonsson and Wahlström.

In fact, the vast majority of exponential-time algorithms used in computational practice have never been analyzed theoretically with respect to their running time. Nevertheless, such algorithms are often used to solve even moderate-to-large instances of some $\mathcal{NP}$-hard problems. One well-known example is the symmetric TSP , where the Concorde TSP Solver (produced by Applegate, Bixby, Chvátal and Cook [44]) can routinely solve TSP instances of order from several hundred to a couple of thousands. The **symmetric (asymmetric) TSP** is the problem to find a minimum weight Hamilton cycle in a weighted complete undirected (directed) graph. Interestingly, currently the best way to solve the asymmetric TSP is to transform it to the symmetric TSP, see the chapter [711] by Naddef.

Most practical exponential-time algorithms are based on the **branch-and-bound** method (see, e.g., the book [742] by Papadimitriou and Steiglitz). In its simplest form, branch-and-bound is just an organized way of taking a hard problem and splitting it into two or more smaller (and hence easier) subproblems. If these subproblems are still too hard, we 'branch' again and further subdivide the problems. The process is repeated until each of the subproblems can be easily solved. Branching is done in such a way that solving each of the subproblems (and selecting the best answer found) is equivalent

to solving the original problem. The process of branching can be viewed as a tree.

Bounds are used to eliminate some branches of the tree. Consider a minimization problem such as TSP. If a lower bound of the cost of a solution of one of the subproblems is higher than the currently best solution, then the subproblem can be **fathomed**, i.e., no further branching from the subproblem is needed. Lower bounds are often easy to obtain. For example, for TSP the minimum weight of a cycle factor is clearly a lower bound for the minimum weight of a Hamilton cycle.

There are several variants of branch-and-bound such as branch-and-cut [711], branch-and-price, branch-and-peg (see [419] by Goldengorin, Ghosh and Sierksma), branch-and-win (see [744] by Pastor and Corominas) and cut-and-solve (see [225] by Climer and Zhang). There are other methods to construct practical and theoretical exponential algorithms. Perhaps the most prominent of them is dynamic programming (also used to build polynomial-time algorithms, see Section 3.3).

Computational experience shows that to construct a relatively fast exponential algorithm one should study very well the underlying problem and, possibly, some related problems as well. Among the best-known success stories in the area are TSP branch-and-cut algorithms obtained as a result of extensive combinatorial studies, see e.g., the survey [711] by Naddef.

18.6 Approximation Algorithms

There are several situations when the use of exact optimization algorithms does not seem to be a good idea. One is when the time is greatly limited or the problem should be solved online. Another is when the data are not exact or the objective function is not well-defined and, thus, we cannot get an optimal solution even by exhaustive search. In such situations, we can use approximation algorithms for finding a solution that is often not optimal, but we have some performance guarantee in each case.

Let P be a combinatorial optimization problem, and let $\mathcal{A}$ be an approximation algorithm for P. Let $X(I)$ denote the set of all feasible solutions for some instance $I \in P$ and let $|I|$ be the size of I. We denote the solution obtained by $\mathcal{A}$ for an instance I of P by $x(I)$. Furthermore let $opt(I)$ denote the optimal solution of I. When considering the weight of a solution y we write $w(y)$.

The theoretical performance of approximation algorithms is normally measured by the (worst-case) **performance ratio**. Usually, upper or lower bounds for the worst-case performance ratio are obtained, where the performance ratio is defined as

$$\max_{I \in P:|I|=n} \left\{ \frac{w(x(I))}{w(opt(I))}, \frac{w(opt(I))}{w(x(I))} \right\}.$$

The performance ratio defined in this way has its advantage in the fact that it is always at least 1 (for both minimization and maximization problems).

We normally require that an approximation algorithm has a polynomial running time. Some approximation algorithms provide a good performance guarantee. For example, the well-known Christofides algorithm [215] for the symmetric TSP[3] with triangle inequality (i.e., $w_{ij}+w_{jk} \geq w_{ik}$ for every triple i, j, k of vertices, where w_{ij} is the weight of an edge ij) has performance ratio 1.5. There are no approximation algorithms of constant performance ratio for the (general) symmetric TSP. This is because the existence of such an algorithm would imply that $\mathcal{P}$=$\mathcal{NP}$ (see Exercise 18.17).

Sometimes, we can overcome the in-approximability, by using another measure of performance guarantee. One such measure was defined by Zemel [927] who provided some mathematical arguments to show that his measure is better, in some sense, than the traditional performance ratio. Let $\mathcal{A}$ be an approximate algorithm for TSP and I a problem instance. Then $w_{\min}(I)$, $w_{\max}(I)$, $w_{\mathcal{A}}(I)$ denote the weights, respectively, of an optimal Hamilton cycle, a heaviest Hamilton cycle and a Hamilton cycle produced by algorithm $\mathcal{A}$ for instance I. The **Zemel measure** of algorithm $\mathcal{A}$, denoted $\rho_z(\mathcal{A})$, is the supremum of $(w_{\mathcal{A}}(I)-w_{\min}(I))/(w_{\max}(I)-w_{\min}(I))$, taken over all TSP instances I for which $w_{\max}(I) \neq w_{\min}(I)$. The following theorem was proved by Hassin and Khuller [504].

Theorem 18.6.1 *There is a polynomial-time approximate algorithm $\mathcal{A}$ for TSP with $\rho_z(\mathcal{A}) \leq \frac{1}{2}$, and one for the symmetric TSP with $\rho_z(\mathcal{A}) \leq \frac{1}{3}$.* □

For many results on approximation algorithms and in-approximability, see the book [53] by Ausiello, Crescenzi, Gambosi, Kann, Marchetti-Spaccamela and Protasi.

18.7 Heuristics and Metaheuristics

While approximation algorithms are of interest to practical applications, we often cannot show any performance guarantee of some very useful practical algorithms and, moreover, we know that several very useful practical algorithms are of very poor performance guarantee in general case. It happens that these algorithms perform quite well for the vast majority of interesting instances. When we do not consider any performance guarantee, we speak of **heuristics**. **Metaheuristics** are classes of heuristics based on a general idea or approach.

Heuristics may not seem very interesting to the theoretician who wishes to consider only methods that provably obtain the optimum or some approximation guarantee for the solution. However, in practice the situation may

[3] Recall that the symmetric TSP is the problem of finding a minimum weight Hamilton cycle in a weighted complete undirected graph.

well be entirely different: the engineer who has been asked to find a reasonable solution to an instance of some optimization problem cannot really use this attitude. What (s)he needs is a way to get a good solution and some indication that this solution is much better than a random solution and cannot be easily improved on (recall the discussion concerning the domination number of algorithms for the TSP problem in Section 17.4).

There are many thousands papers on heuristics and metaheuristics as well as several books treating the subject in great detail. In this section, we will only consider a small number of ideas and give some references for further reading. As an illustration, we will use the Feedback Arc Set (FAS) problem as defined in Subsection 15.3.3: given a digraph $D = (V, A)$, find an ordering $s = v_1, v_2, \ldots, v_n$ of V with minimum number of **backward arcs**, i.e., arcs of the form $v_i v_j$ with $i > j$. We will consider some implemented FAS heuristics and comment on their implementation and performance. We supplement our comments on FAS heuristics with some comments on heuristics for TSP and other optimization problems.

We start with a very simple method for finding an ordering which is locally optimal. Let $s = v_1, v_2, \ldots, v_n$ be an ordering of V. Now suppose that there are indices i, j such that by deleting the vertex v_j and reinserting it between v_i and[4] v_{i+1} we obtain a smaller backward arc set. The effect on the value of the backward arc set can be calculated easily without reconsidering all arcs (Exercise 18.30).

To use a more general terminology, we call an ordering of V, a **solution**. The **value** $v(s)$ of a solution s is the number of backward arcs with respect to s. We say that two solutions s, s' are **neighbours** if we can obtain one from the other by deleting one vertex and reinserting it somewhere else in the ordering of the remaining vertices. The **neighbourhood** $N(s)$ of s is the set of solutions that are neighbours of s. Now we can describe a very simple heuristic which we call **1-OPT** for the FAS problem:

1-OPT
Input: A directed multigraph $D = (V, A)$;
Output: An ordering of V (for which the backward arcs form a feedback arc set in D).

1. Start with a solution s corresponding to a random permutation of V;
2. If there exists a neighbour s' of s such that $v(s') < v(s)$; then take $s := s'$ as the new current solution and repeat this step;
3. Output the locally optimal solution s and halt.

It is easy to show (Exercise 18.31) that the 1-OPT algorithm will halt after finitely many steps with a solution that is locally optimal. Here **locally optimal** means that the number of backward arcs cannot be decreased by moving a single vertex.

[4] We allow $i = n$ and $i = 0$ with the obvious meaning of v_{i+1} and v_0.

There are several other ways of defining interesting neighbourhoods of a solution to the FAS problem. For example, one could consider all solutions that can be obtained by interchanging the positions of two vertices in the given ordering. Experimental evidence found by Olsen [732] suggests that this last way of choosing the neighbourhood does not produce as high quality solutions as the one above. Coleman and Wirth [226] came up with the same conclusion for FAS on tournaments.

Although 1-OPT produces solutions that are in general much better than a random choice, it only guarantees that the final solution found is locally optimal. Furthermore, since a new solution is only taken if it improves the objective function, the algorithm cannot escape a local minimum. This can be remedied somewhat by restarting the algorithm several times from new random orderings of V. Since the algorithm is usually very fast it is possible to restart it many times (from different random solutions) and then take the best solution among the local optima which have been found. Computational experience with TSP and other optimization problems demonstrates that **chained local search** is a better choice; in chained local search, each iteration of local search starts from a somewhat perturbed local optimum obtained in one of the previous iterations. It seems Baum [128] was the first to introduce chained local search; the method proved to be very successful for the symmetric TSP (see, e.g., Johnson and McGeoch [572]).

Another method to escape local minima would be to allow a neighbour s' of the current solution s with $v(s') > v(s)$ to be chosen with some positive probability. However, unless this probability decreases as the number of steps increases the method may never converge towards a local minimum.

This problem is handled in the next method called **simulated annealing**. The basic idea is to allow a neighbouring solution s' with $v(s') > v(s)$ to be chosen with a probability p which depends both on $\tau = v(s') - v(s)$ and the number of steps taken by the algorithm so far.

Below we describe the generic simulated annealing method for a minimization problem over the set S of possible solutions and with objective function f and neighbourhood structure N. Note that this is a **metaheuristic**, i.e., it is a scheme that can be applied to many types of combinatorial optimization problems rather than just one specific problem.

Generic Simulated Annealing

1. Select an initial solution s_0;
2. Initialize control parameter t to a value t_0;
3. Select a reduction method M for the control parameter t;
4. Repeat $K(n)$ times:
5. Choose randomly a neighbour $s \in N(s_0)$;
6. Let $\eta := f(s) - f(s_0)$;
7. If $\eta \leq 0$ then $s_0 := s$
8. Else let $s_0 := s$ with probability $\exp(-\eta/t)$;
9. Let $t := M(t)$;

10. If the stopping condition is satisfied then return the best solution encountered and halt. Otherwise go to Step 4.

Although we did not write it above, it is understood that the algorithm also keeps track of the best solution found so far (note that this may not coincide with the current solution s_0).

It is evident from the (loose) description above that any implementation of the method involves making several choices about how to perform the various steps. We discuss briefly the general idea below and refer to the survey [275] by Dowsland and the experimental evaluation of simulated annealing by Johnson, Aragon, McGeoch and Schevon described in [570] for more details. It is important to note that finding a good set of values/methods to implement the algorithm is by no means always a trivial task. Part of this process consists of **tuning** the parameters t_0, $K(n)$, the method M for decreasing t and the stopping criterion. This is done by performing a number of runs with all but one parameter fixed and then selecting values that look promising. After some stages of this process, one may arrive at a choice for the parameters which does not seem easy to improve (based on the test data used). However, experimental evidence reported by Hansen [497] and Olsen [732] indicate that for a problem such as FAS it is not too hard to make a set of choices which will make the algorithm perform quite well.

The initial solution can be chosen arbitrarily or it may be a local optima found by 1-OPT, say. The control parameter t should be initialized so that in the beginning there is a fair chance that the algorithm will accept a neighbour with a higher f value than the current solution s_0. Normally this is done by starting from a random solution and then performing, say, 1000 steps of the algorithm while keeping track of the number of neighbours who are accepted as the new current solution[5]. The **initial acceptance rate** is the fraction of accepted solutions over the total number of neighbours tested (1000 above). Experiments reported in e.g. [570] suggest that acceptance rates in the interval $[0.3, 0.9]$ all work well (these experiments were not for the FAS problem, but the conclusion also seems to hold for FAS [732]).

Experiments show that the actual reduction method used to reduce t after every cycle of $K(n)$ steps is not as important as the rate at which t is reduced. This rate should be as slow as possible (that is, as time allows) [570]. In fact, whereas in general no theoretical guarantee exists for the quality of a solution found by local search heuristics such as 1-OPT, it can actually be shown (see e.g. the book [51] by Arts and Korst) that under ideal conditions (such as reducing the parameter t infinitely slowly, taking a very large number of steps for each value of t and using a neighbour structure that allows one to reach some optimal solution from an arbitrary solution) simulated annealing will in fact find an optimal solution. Of course such a result is only of theoretical interest, but the nice thing is that, since some of these results are based

[5] This includes those that have a better (or equal) value than the current solution as well as those that are worse, but are chosen in the probabilistic step 8.

on Markov chains, the results suggest that the slower one reduces t and the higher $K(n)$ (as a function of the size of the neighbourhood), the better results one should obtain. This thesis seems to be true for several applications of simulated annealing (see e.g. [275, 570]).

It is common to use a simple geometric reduction method where we set $t := rt$ for some fixed number $0 < r < 1$ which is close to one. Experiments by Johnson et al. suggest that $r = 0.95$ is generally a good choice [570]. The number of steps $K(n)$ for each value assumed by t should be at least a linear function in the size of the neighbourhood of an arbitrary solution. Finally it is common to use as a stopping condition that there has been no improvement in the current solution for some number N of moves. Another possibility is to use the **current acceptance rate** (calculated similarly as the initial acceptance rate by keeping track of the number of accepted moves over the last, say, 1000 steps) as a measure and stop when this rate gets below, say 1 percent. One may also decide to stop when the control parameter becomes smaller than a prescribed value t_s. Note that in the last case, the number of steps performed by the algorithm is always the same (for $K(n)$ and M fixed).

Tabu search described in detail by Glover and Laguna [408] is a metaheuristic somewhat similar to simulated annealing, but based on an absolutely different principle. Another popular metaheuristic is evolutionary method, where many solutions are generated and used to produce new solutions (see the book [291] by Eiben and Smith). For more information on heuristics and metaheuristics, see, e.g., [291, 408, 410, 532, 572, 771, 772].

18.8 Matroids

In this section we give a short introduction to matroids. The motivation for this is that algorithms for matroids are a useful tool for solving various graph theoretical problems. For an example of this we refer to Section 9.2 and Exercise 18.27. Due to lack of space we will not be able to describe the algorithms for 2-matroid intersection and matroid partition (those are the ones used in the applications mentioned above). We refer the reader to the books [229] by Cook, Cunningham, Pulleyblank and Schrijver and [766] by Recski for detailed descriptions of these algorithms.

Definition 18.8.1 *Let S be a finite set and let $\mathcal{I}$ be a collection of subsets of S. Recall that $M = (S, \mathcal{I})$ is an* **independence system** *if (I1) and (I2) below hold (see Subsection 17.3.1). If also (I3) holds, M is called a* **matroid**.

(I1) $\emptyset \in \mathcal{I}$.
(I2) If $Y \in \mathcal{I}$ *and* $X \subseteq Y$, *then* $X \in \mathcal{I}$.
(I3) If $X, Y \in \mathcal{I}$ *and* $|X| < |Y|$, *then there exists an element* $y \in Y - X$ *such that* $X \cup \{y\} \in \mathcal{I}$.

Let $M = (S, \mathcal{I})$ be a matroid. A set $X \subseteq S$ such that $X \in \mathcal{I}$ is called **independent**. All other sets are **dependent**. A **base** of M is a maximal independent set. A **circuit** is a minimal dependent set. Let $\mathcal{B}$ denote the set of bases of M and $\mathcal{C}$ the set of circuits of M.

It follows directly from (I3) and the definition of a base that

$$\text{all bases of a matroid have the same size.} \tag{18.1}$$

Below we list some important properties of the bases of a matroid: (B1) follows from (I1), (B2) follows from (I3) and (B3) is left to the reader as Exercise 18.18.

Proposition 18.8.2 *Let $M = (S, \mathcal{I})$ be a matroid. The set $\mathcal{B}$ of bases of M satisfy the following:*

(B1) $\mathcal{B} \neq \emptyset$.
(B2) For all $B, B' \in \mathcal{B}$ *we have* $|B| = |B'|$.
(B3) Let $B, B' \in \mathcal{B}$. *For every* $b \in B$ *there exists an element* $b' \in B'$ *such that* $(B - b) \cup \{b'\} \in \mathcal{B}$. □

The other direction holds as well:

Proposition 18.8.3 *Let S be a finite set and $\mathcal{B}$ a collection of subsets of S which satisfies (B1)-(B3) above. Then there exists a matroid $M = (S, \mathcal{I})$ whose set of bases is precisely $\mathcal{B}$.* □

If $M = (S, \mathcal{I})$ is a matroid and $X \subseteq S$, then we say that a subset $Y \subseteq X$ is a **maximal independent subset of** X if $Y \in \mathcal{I}$ and $Y \subset Z \subseteq X$ implies $Z \notin \mathcal{I}$. Assertion (18.1) can be generalized as follows:

Proposition 18.8.4 *Let $M = (S, \mathcal{I})$ be a matroid and let $X \subseteq S$. All maximal independent subsets of X have the same size.* □

The following formula defines the **rank function** of a matroid:

$$r(X) = \max\{|Y| : Y \subseteq X \text{ and } Y \in \mathcal{I}\}.$$

The **rank** of a matroid $M = (S, \mathcal{I})$ is the number $r(S)$, the size of a base in M.

Examples of matroids:

(1) Let $G = (V, E)$ be an undirected graph. Define $M(G)$ as $M(G) = (E, \mathcal{I})$, where $E' \in \mathcal{I}$ if and only if $G_{E'} = (V, E')$ has no cycle. Then $M(G)$ is a matroid (called the **circuit matroid of G**). To see this, it suffices to check (I3), since (I1),(I2) trivially hold. Let X, Y be subsets of E such that none of $G\langle X\rangle$ and $G\langle Y\rangle$ has a cycle and $|X| < |Y|$. It is easy to show that if Z is independent in $M(G)$, then the number of connected

components in $G\langle Z\rangle$ is $n - |Z|$, where n is the number of vertices in G. Thus $|X| < |Y|$ implies that the number of connected components of $G\langle X\rangle$ is larger than that of $G\langle Y\rangle$. Hence Y contains an edge y such that y joins two vertices which are in distinct components of $G\langle X\rangle$. This implies that $G\langle X \cup \{y\}\rangle$ is acyclic and hence $X \cup \{y\} \in \mathcal{I}$.
The bases of $M(G)$ are the (sets of edges of) maximal forests of G and a cycle of $M(G)$ is a fundamental cycle of G with respect to a maximal forest of G. The rank of $M(G)$ is $|V|$ minus the number of connected components of G.

(2) Let S be a set on n elements, and define $U_{n,k}$ for $k \leq n$ as follows: $U_{n,k} = (S, \{X \subseteq S : |X| \leq k\})$. This trivially gives a matroid called a **uniform matroid**. If $k = n$, we obtain a very special case in which all subsets are independent. This matroid is called the **free matroid** on n elements.

(3) Let $D = (V, A)$ be a digraph such that $\delta^-(D) > 0$ and define $\mathcal{B}$ as those subsets A' of A for which every vertex $v \in V$ has in-degree precisely one in $D\langle A'\rangle$. We show that $\mathcal{B}$ satisfies (B1)-(B3) of Proposition 18.8.2 and hence, by Proposition 18.8.3, $\mathcal{B}$ forms the set of bases of a matroid $M^-(D)$. Indeed, (B1) holds since $\delta^-(D) > 0$ and (B2) holds by the definition of $\mathcal{B}$. To see that (B3) is true consider sets $A', A'' \in \mathcal{B}$ and let $a' \in A'$. The arc a' enters a vertex x and in A'' there is exactly one arc a'' with head x. Now we see that $(A' - a') \cup \{a''\} \in \mathcal{B}$.
Similarly, if $\delta^+(D) > 0$, then we may define a matroid $M^+(D)$ whose bases are those subsets X of the arcs for which every vertex $v \in V$ has out-degree precisely one in $D\langle X\rangle$. This follows from the argument above by considering the converse of D.

The next result shows, in particular, that the rank function of a matroid is submodular. This is one of the main reasons for the usefulness of matroids.

Proposition 18.8.5 *The rank function of $M = (S, \mathcal{I})$ satisfies the following:*
(R1) $0 \leq r(X) \leq |X|$ *for every* $X \in S$.
(R2) $X \subseteq Y$ *implies* $r(X) \leq r(Y)$.
(R3) For all $X, Y \subseteq S$: $r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$.

Proof: (R1) and (R2) follow from the definitions. To see that (R3) holds consider two subsets X, Y of S. We may assume that $X \neq Y$. Let A be a maximal independent subset of $X \cap Y$ and let B be an extension of A to a maximal independent subset of $X \cup Y$. Now using (R2) we have

$$r(X) + r(Y) \geq |B \cap X| + |B \cap Y| = |B| + |A| = r(X \cup Y) + r(X \cap Y).$$

□

18.8.1 The Dual of a Matroid

The **dual** of a matroid $M = (S, \mathcal{I})$ is the pair $M^* = (S, \mathcal{I}^*)$, where $\mathcal{I}^* = \{X \subseteq S : X \cap B = \emptyset$ for some base B of M$\}$. In Exercise 18.19 the reader

is asked to prove that M^* is a matroid. Note that the bases of M^* form precisely the set $\mathcal{B}^* = \{S - B : B \text{ is a base of } M\}$.

Proposition 18.8.6 *For any matroid M we have*

(a) $(M^*)^* = M$.
(b) $r^*(X) = |X| + r(S - X) - r(S)$.

Proof: Exercise 18.20. □

A circuit in M^* is called a **cutset** or a **cocircuit** in M. It follows from the definition of M^* that a cocircuit of M is a minimal subset of S which has a non-empty intersection with all bases of M.

18.8.2 The Greedy Algorithm for Matroids

Let $M = (S, \mathcal{I})$ be a matroid. In Section 17.3, we described the greedy algorithm $\mathcal{GA}$ for independence systems. The aim of this section is to prove that $\mathcal{GA}$ always solves the $(S, \mathcal{I}, \mathbb{R}_0)$-minimization problem (i.e., the problem of finding a base of minimum weight) to optimality.

As in Section 17.3, for every $X \in \mathcal{I}$ we define $\text{ext}(X) = \{y \in S - X : X \cup \{y\} \in \mathcal{I}\}$. Suppose that we are given a weight function $w : S \to \mathbb{R}_0$ on the elements of S. We let $w(X) = \sum_{x \in X} w(x)$. The goal of the $(S, \mathcal{I}, \mathbb{R}_0)$-minimization problem is to find a base of M with minimum weight. An **optimal base** is a base B such that $w(B) \leq w(B')$ for every $B' \in \mathcal{B}$.

The greedy algorithm tries to construct a minimum weight base as follows: it starts from an empty set X, and at every step it takes the current set X and adds to it a minimum weight element $e \in \text{ext}(X)$; the algorithm stops when a base is built.

The following theorem was obtained by Rado [759].

Theorem 18.8.7 *Applied to matroids the greedy algorithm always finds an optimal base.*

Proof: Let $M = (S, \mathcal{I})$ be a matroid and let w be a weight function. Let G be a base returned by the greedy algorithm and let B be another base. Let $G = \{g_1, g_2, \ldots, g_r\}$ and $B = \{b_1, b_2, \ldots, b_r\}$, where $w(g_i) \leq w(g_{i+1})$ and $w(b_i) \leq w(b_{i+1})$ for every $i \in [r-1]$. Let $G_j = \{g_1, g_2, \ldots, g_j\}$. We may assume that $\mathcal{GA}$ builds G starting from $G_0 = \emptyset$ and forming first G_1 then G_2, G_3, etc.

To prove that $w(G) \leq w(B)$, we will prove that, in fact, $w(g_i) \leq w(b_i)$ for each $i \in [r]$. For the definition of $\mathcal{GA}$, the claim holds for $i = 1$. Suppose that $w(g_i) \leq w(b_i)$ for each $i \in [j]$, but $w(g_{j+1}) > w(b_{j+1})$.

Consider the set $A = \{s \in S : \ w(s) \leq w(b_{j+1})\}$. Observe that G_j is a maximal independent subset of A as otherwise there would be $a \in \text{ext}(G_j)$ with $w(a) \leq w(b_{j+1}) < w(g_{j+1})$ and $\mathcal{GA}$ would append a instead of g_{j+1} to

G_j. On the other hand, B_{j+1} is an independent subset of A and $|B_{j+1}| > |G_j|$, a contradiction with Proposition 18.8.4. □

It is well-known that matroids are the only independence systems for which the greedy algorithm always provides an optimal base, see Exercise 18.21.

18.8.3 Independence Oracles

What is a fast algorithm for matroids? How do we represent a matroid efficiently? These are important questions. In particular, it should be clear that in general it is infeasible to store information about a given matroid by a list of its independent sets. For example, if M is the uniform matroid $U_{n,k}$, we would have to store all subsets of size at most k of $[n]$. On the other hand, for $U_{n,k}$ it is very easy to decide whether a given subset of $[n]$ is independent: simply calculate its size and check whether this is at most k. This illustrates that what is important is not having a list of all independent sets, but rather to be able to determine whether a given subset X of the ground set S is independent in M.

We shall assume that our matroids are always given in terms of the ground set S and a subroutine O_M which given $X \subseteq S$ decides whether X is independent in M or not. Such a subroutine O_M is called an **independence oracle** for $M = (S, \mathcal{I})$. We say that a matroid algorithm $\mathcal{A}$ for a matroid $M = (S, \mathcal{I})$ with independence oracle O_M is **fast** if the number of steps of $\mathcal{A}$ is polynomial in $|S|$ and any other inputs (such as a weight function), provided that we consider each call to O_M as taking constant time. With this assumption, the greedy algorithm is a fast matroid algorithm.

18.8.4 Union of Matroids

Let $M_i = (S, \mathcal{I}_i)$, $i \in [k]$, be matroids on the same ground set S. Define $\vee_{i=1}^k M_i = (S, \vee_{i=1}^k \mathcal{I}_i)$ as follows. A set $X \subseteq S$ is independent in $\vee_{i=1}^k M_i$ if and only if X can be partitioned as $X = X_1 + X_2 + \cdots + X_k$, where $X_i \in \mathcal{I}_i$ for each $i \in [k]$. It is a non-trivial exercise (Exercise 18.23) to prove the following:

Proposition 18.8.8 *Let $M_i = (S, \mathcal{I}_i)$, $i \in [k]$, be matroids on the same ground set S. Then $\vee_{i=1}^k M_i$ is a matroid.* □

The Matroid Partition Problem: Let $M_i = (S, \mathcal{I}_i)$, $i \in [k]$, be matroids on the same ground set S and a set $X \subseteq S$. Do there exist subsets $X_1, X_2, \ldots, X_k$ of S such that $X = \sum_{i=1}^k X_i$ and $X_i \in \mathcal{I}_i$ for $i \in [k]$?

In Exercise 18.26 the goal is to show that the question of deciding whether an undirected graph has k edge-disjoint spanning trees can be formulated as a matroid partition problem. Hence the following theorem implies the existence

of a polynomial algorithm for deciding whether an undirected graph has k edge-disjoint spanning trees (see Exercise 18.27).

Theorem 18.8.9 *The Matroid Partition Problem can be solved in polynomial time, provided we are given polynomial time realizable independence oracles for each of the matroids M_i, $i \in [k]$.* □

We refer the reader to Recski's book [766] for a description of a fast algorithm for the Matroid Partition Problem. Note that if $M = (S, \mathcal{I})$ is a matroid and X is a subset of S, then $M\langle X\rangle = (X, \mathcal{I}_X)$, where $\mathcal{I}_X = \{Y \in \mathcal{I} : Y \subseteq X\}$ is also a matroid (Exercise 18.24). Hence, the Matroid Partition Problem is equivalent to the problem of deciding whether the ground set S is independent in $\vee_{i=1}^{k} M_i$. This is the problem solved in [766].

18.8.5 Intersection of Two Matroids

Another very useful topic on matroids is matroid intersection. By this we do not mean that if M_1, M_2 are matroids on the same ground set S, then $M = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ is also a matroid. This is false as the reader can easily show by an example (Exercise 18.28). Instead we are interested in the following:

Matroid Intersection Problem: Given matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$, find a maximum cardinality subset $T \subseteq S$ which is independent in each of M_1, M_2.

We may assume that $r_1(S) = r_2(S)$ as if $t = r_1(S) - r_2(S) > 0$, then we can add t new elements to S and every independent set of M_2. If we follow this by appending to M_2 every subset of the t elements, we get a new matroid M_2 such that $r_1(S) = r_2(S)$.

The next result shows that the Matroid Intersection Problem and the Matroid Partition Problem are closely related.

Theorem 18.8.10 *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids on the same ground set S with $r_1(S) = r_2(S) = r$ and let $n = |S|$. There is a common base of M_1, M_2 if and only if $M_1 \vee M_2^* = U_{n,n}$.* □

The following result is due to Edmonds [284].

Theorem 18.8.11 *The Matroid Intersection Problem can be solved in polynomial time, if polynomial time realizable independence oracles for M_1, M_2 are given. Furthermore, under the same assumptions, one can find in polynomial time a maximum (or minimum) weight common independent subset with respect to any given real-valued weight function w on S.* □

For a description of a polynomial algorithm for (weighted) matroid intersection see e.g. [229, 766]. Matroid intersection is a very useful tool for modeling (and solving) many combinatorial optimization problems. For instance,

the problem of finding a minimum weight cycle factor in an arc weighted digraph can be formulated as a weighted two-matroid intersection problem. Consider the intersection of the matroids $M^-(D), M^+(D)$ which were defined in the beginning of this section. There is a common base of these matroids if and only if D has a cycle factor and furthermore, the minimum weight of a common base equals the minimum weight of a cycle factor. Two more examples are given in Section 9.2 and Exercise 18.29.

18.8.6 Intersections of Three or More Matroids

If we consider three or more matroids all on the same ground set and ask for a common base of these, then this problem contains quite a few difficult problems as special cases as we shall see below.

The K-matroid Intersection Problem: Given matroids $M_i = (S, \mathcal{I}_i)$, $i \in [k]$, on the same ground set. Does there exist a set $X \subseteq S$ such that X is a base of M_i for $i = 1, 2, \ldots, k$?

Theorem 18.8.12 *The k-matroid intersection problem is $\mathcal{NP}$-complete for each $k \geq 3$.*

Proof: It suffices to prove the theorem for $k = 3$ since the proof can easily be extended to higher k by using several copies of the same matroid. We will prove that the $\mathcal{NP}$-complete problem of deciding the existence of a hamiltonian path which starts in a prescribed vertex u and ends in a prescribed vertex v in a digraph (see Exercise 7.2) can be reduced to the 3-matroid intersection problem in polynomial time.

Let $D = (V, A)$ be a digraph with specified vertices $u, v \in V$. Define $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, 3$, as follows:

$S = A$; $M_1 = M(UG[D])$;
$X \in \mathcal{I}_2$ if and only if there is no arc entering u in $D_X = (V, X)$ and every other vertex has at most one arc entering it in D_X.
$Y \in \mathcal{I}_3$ if and only if there is no arc leaving v in $D_Y = (V, Y)$ and every other vertex has at most one arc leaving it in D_Y.

We argued in Section 9.2 that $M_2 = (A, \mathcal{I}_2)$ is a matroid and, similarly, $M_3 = (A, \mathcal{I}_3)$ is a matroid. It is easy to see that D has a Hamilton path P from u to v if and only if M_1, M_2, M_3 have a common base (the arcs of a Hamilton path correspond to a common base of M_1, M_2, M_3). □

18.9 Exercises

18.1. Show how to extend the algorithm MergeHamPathTour (see Section 18.1) so that it works for tournaments with an arbitrary number of vertices.

18.2. Based on the proof of Theorem 1.5.1, give a polynomial algorithm to find cycles of lengths $3, 4, \ldots, n$ through a given vertex in a strong tournament T. What is the complexity of your algorithm and how do you store information about T and the cycles you find?

18.3. **(+) Fast algorithm for Euler trails.** Demonstrate how to implement the algorithm in the proof of Theorem 1.7.2 as an $O(n + m)$ algorithm. Hint: use adjacency lists along with a suitable data structure to store the trail constructed so far.

18.4. Suppose $\mathcal{S}, \mathcal{T}, \mathcal{K}$ are decision problems such that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$ and $\mathcal{T} \leq_{\mathcal{P}} \mathcal{K}$. Prove that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{K}$.

18.5. The **independent set problem** is as follows: Given a graph $G = (V, E)$ and natural number k, decide whether G has an independent set of size at least k. Show that the independent set problem belongs to the complexity class $\mathcal{NP}$.

18.6. Suppose $\mathcal{W}$ is an $\mathcal{NP}$-complete problem and that $\mathcal{T}$ is a decision problem such that $\mathcal{W} \leq_{\mathcal{P}} \mathcal{T}$. Prove that $\mathcal{T}$ is $\mathcal{NP}$-hard.

18.7. **The acyclic subdigraph problem.** Let $\mathcal{S}$ be the following decision problem. Given a digraph D and a natural number k, does D contain an induced acyclic subdigraph on at least k vertices? Show that the independent set problem polynomially reduces to $\mathcal{S}$ (the independent set problem is: given a graph G and a number k, does G contain an independent set of size at least k?).

18.8. Show that if a decision problem $\mathcal{S}$ belongs to the complexity class $\mathcal{P}$, then it also belongs to $\mathcal{NP}$.

18.9. Show that $\mathcal{P} \subseteq \mathcal{NP} \cap co\text{–}\mathcal{NP}$.

18.10. Show that if there is some decision problem $\mathcal{S}$ which belongs to both of the classes $\mathcal{P}$ and $\mathcal{NPC}$, then $\mathcal{P} = \mathcal{NP}$.

18.11. **(+) Reducing the Hamilton cycle problem to Satisfiability.** Describe a polynomial reduction from the Hamilton cycle problem to the Satisfiability problem. Hint: model different attributes by different sets of clauses. For example, you should use one family of clauses to ensure that every vertex is the tail of at least one arc.

18.12. Describe a polynomial reduction from the problem of deciding whether an undirected graph has a matching of size k to the problem MAX-2-SAT.

18.13. **(+) A special case of the maximum independent set problem**. The maximum independent set problem is as follows. Given an undirected graph G, find an independent set of maximum cardinality in G. The purpose of this exercise is to show that a special case of the maximum independent set problem is equivalent to the 2-satisfiability problem and hence can be solved using any algorithm for 2-SAT.

(a) Let $G = (V, E)$ be a graph on $2k$ vertices and suppose that G has a perfect matching (i.e., a collection $e_1, \ldots, e_k$ of edges with no common end-vertex). Construct an instance $\mathcal{F}$ of 2-SAT which is satisfiable if and only if G has an independent set of k vertices. Hint: fix a perfect

matching M of G and let each edge in M correspond to a variable and its negation.

(b) Prove the converse, namely, if $\mathcal{F}$ is any instance of 2-satisfiability, then there exists a graph $G = (V, E)$ with a perfect matching such that G has an independent set of size $|V(G)|/2$ if and only if $\mathcal{F}$ is satisfiable.

(c) Prove that it is $\mathcal{NP}$-complete to decide if a given graph has an independent set of size at least ℓ, even if G has a perfect matching. Hint: use a reduction from MAX-2-SAT.

18.14. **Finding a 1-maximal cycle.** A cycle C in a digraph D is **1-maximal** if D has no cycle C' such that $C - a$ is a subpath of C' for some arc a of C. Describe a polynomial algorithm for finding a 1-maximal cycle in a strong digraph. What is the complexity of your algorithm? Hint: compare it with the proof of Theorem 1.5.1.

18.15. Describe a linear time algorithm to check whether a given acyclic digraph has more than one acyclic ordering. Hint: use the result of Exercise 2.1.

18.16. **Transitive subtournaments in tournaments**. Show that every tournament on 8 vertices contains a transitive tournament on 4 vertices (as an induced subdigraph). Hint: start from a vertex of maximum out-degree. Use the idea above to prove that every tournament on n vertices contains a transitive tournament of size $\Omega(\log n)$.

18.17. Let $r > 1$ be a real number. Prove that there is no polynomial-time approximation algorithm of performance ratio r for the symmetric TSP unless $\mathcal{P}$=$\mathcal{NP}$.

18.18. Prove that (B3) holds for any matroid.

18.19. Prove that if M is a matroid, then the dual M^* is also a matroid.

18.20. Prove Proposition 18.8.6.

18.21. (+) Prove the following result:

Theorem 18.9.1 *Let $M = (S, \mathcal{I})$ satisfy (I1),(I2). The greedy algorithm $\mathcal{GA}$ finds an optimal base for M for every choice of non-negative real-valued weight function w on S if and only if M is a matroid.*

Hint: show that if $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k, b_{k+1}\}$ both belong to $\mathcal{I}$, then one can choose a weight function w on the elements of S so that $\mathcal{GA}$ will always choose A as the first k elements and unless there is a $b_i \in B$ such that $A \cup \{b_i\} \in \mathcal{I}$, $\mathcal{GA}$ will not reach an optimal base.

18.22. Describe an $O(n + m)$ algorithm for deciding whether an undirected graph on n vertices and m edges has a cycle.

18.23. (+) Prove Proposition 18.8.8. Hint: it suffices to prove the claim for two matroids. Consider a counterexample X, Y to (I3) with $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$, $X_1, Y_1 \in \mathcal{I}_1$, $X_2, Y_2 \in \mathcal{I}_2$ and $|X_1 \cap Y_2| + |X_2 \cap Y_1|$ is maximum.

18.24. Prove that $M\langle X\rangle$ defined in Section 18.8 is a matroid.

18.25. Let $D = (V, A)$ be a digraph with two vertices s, t such that $\lambda(s, t) \geq k$ for some k. Define $\mathcal{I}$ by $\mathcal{I} = \{X \subseteq A : \lambda_{D-X}(s, t) \geq k\}$. Show by an example that $(A, \mathcal{I})$ is not always a matroid. (+) Can you characterize those digraphs for which $(A, \mathcal{I})$ is actually a matroid?

18.26. **(+) Testing for k edge-disjoint spanning trees in graphs.** Show how to formulate the problem of deciding whether an undirected graph G has k edge-disjoint spanning trees as a matroid partition problem.

18.27. **(+) An algorithm for deciding the existence of k edge-disjoint spanning trees.** Use the formulation in Exercise 18.26 to derive a polynomial algorithm for deciding whether an undirected graph has k edge-disjoint spanning trees. Remember to justify that the required oracles can be implemented as polynomial algorithms.

18.28. Give an example of two matroids M_1, M_2 on the same ground set S for which $M = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ is not a matroid.

18.29. **(+) Formulating the maximum (weight) matching problem for a bipartite graph as a (weighted) matroid intersection problem.**
(a) Show how to formulate the question of deciding the existence of a matching of size n in a bipartite graph $G = (U, V, E)$ on $2n$ vertices as a matroid intersection problem.
(b) Show how to solve the problem of finding a maximum weight matching of size n in the graph G above if we are given non-negative weights on the edges of G.
(c) Argue that one can in fact find a maximum matching in any bipartite graph in polynomial time, using an algorithm for the matroid intersection problem.

18.30. Consider the 1-OPT method for the FAS problem. Describe how to determine, in linear time, the number of backward arcs with respect to the ordering we obtain from $v_1, v_2, \ldots, v_n$ after removing one vertex from position j and reinserting it between v_i and v_{i+1}.

18.31. Prove that the 1-OPT algorithm applied to the feedback arc set problem will always halt. Then give a good bound on the number of steps taken by the algorithm.

References

1. A. Abouelaoualim, K.Ch. Das, L. Faria, Y. Manoussakis, C.A. Martinhon, and R. Saad. Paths and Trails in Edge-Colored Graphs. In *Proc. LATIN'08*, volume 4957 of *Lect. Notes Comput. Sci.*, pages 723–735. Springer-Verlag, 2008.
2. A. Abouelaoualim, K.Ch. Das, W. Fernandez de la Vega, M. Karpinski, Y. Manoussakis, C.A. Martinhon, and R. Saad. Cycles and paths in edge-colored graphs with given degrees. Manuscript, 2007.
3. A. Ádám. Problem, In *'Theory Graphs Applications', Proc. Coll. Smolenice*, pages 12–18, Czech. Acad. Sci. Publ., 1964.
4. A. Ádám. Bemerkungen zum graphentheoretischen Satze von I. Fidrich. *Acta Math. Acad. Sci. Hung.*, 16:9–11, 1965.
5. A. Ádám. On some cyclic connectivity properties of directed graphs. *Acta Cybernet.*, 14:1–12, 1999.
6. L. Addario-Berry, F. Havet, and S. Thomassé. Paths with two blocks in n-chromatic digraphs. *J. Combin. Theory Ser. B*, 97:620–626, 2007.
7. R.L. Adler, L.W. Goodwyn, and B. Weiss. Equivalence of topological Markov shifts. *Israel J. Math.*, 27:48–63, 1977.
8. R. Aharoni and R. Holzman. Fractional kernels in digraphs. *J. Combin. Theory Ser. B*, 73(1):1–6, 1998.
9. R. Aharoni and C. Thomassen. Infinite, highly connected digraphs with no two arc-disjoint spanning trees. *J. Graph Theory*, 13(1):71–74, 1989.
10. A.V. Aho, M.R. Garey, and J.D. Ullman. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1(2):131–137, 1972.
11. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, Reading, Mass., 1975.
12. A.V. Aho, R. Sethi, and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, Mass., 1986.
13. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows*. Prentice Hall, Englewood Cliffs, NJ, 1993.
14. M. Aigner and G. Ziegler. *Proofs from the book*. Springer-Verlag, Berlin, 1998.
15. N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *Proc. STOC'05: the 37th Annual ACM Symp. on Theory of Computing*, pages 684–693. ACM Press, 2005.
16. A. Ainouche. An improvement of Fraisse's sufficient condition for hamiltonian graphs. *J. Graph Theory*, 16:529–543, 1992.
17. N. Alon. Disjoint directed cycles. *J. Combin. Theory Ser. B*, 68(2):167–178, 1996.
18. N. Alon. Ranking tournaments. *SIAM J. Discrete Math.*, 20:137–142, 2006.
19. N. Alon. Splitting digraphs. *Combin. Probab. Comput.*, 15:933–937, 2006.
20. N. Alon, B. Bollobàs, A. Gyàrfàs, J. Lehel, and A. Scott. Maximum directed cuts in acyclic digraphs. *J. Graph Theory*, 55:1–13, 2007.

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1,

21. N. Alon, F.V. Fomin, G. Gutin, M. Krivelevich, and S. Saurabh. Better algorithms and bounds for directed maximum leaf problems. In *Proc. Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2007*, volume 4855 of *Lect. Notes Comput. Sci.*, pages 316–327. Springer-Verlag, 2007.
22. N. Alon, F.V. Fomin, G. Gutin, M. Krivelevich, and S. Saurabh. Parameterized Algorithms for Directed Maximum Leaf Problems. In *Proc. ICALP'07: 34th Int. Colloquium on Automata, Languages and Programming*, volume 4596 of *Lect. Notes Comput. Sci.*, pages 352–362. Springer-Verlag, 2007.
23. N. Alon, F.V. Fomin, G. Gutin, M. Krivelevich, and S. Saurabh. Spanning directed trees with many leaves. Preprint arXiv:0803.0701, March 2008.
24. N. Alon and G. Gutin. Properly colored Hamilton cycles in edge colored complete graphs. *Random Struct. Algor.*, 11:179–186, 1997.
25. N. Alon, G. Gutin, and M. Krivelevich. Algorithms with large domination ratio. *J. Algor.*, 50(1):118–131, 2004.
26. N. Alon and N. Linial. Cycles of length 0 modulo k in directed graphs. *J. Combin. Theory Ser. B*, 47(1):114–119, 1989.
27. N. Alon, C. McDiarmid, and M. Molloy. Edge-disjoint cycles in regular directed graphs. *J. Graph Theory*, 22(3):231–237, 1996.
28. N. Alon and J.H. Spencer. *The Probabilistic Method*. John Wiley & Sons, New York, 1992. With an appendix by Paul Erdős.
29. N. Alon and J.H. Spencer. *The Probabilistic Method*. John Wiley & Sons, New York, 2nd edition, 2000.
30. N. Alon and M. Tarsi. Colourings and orientations of graphs. *Combinatorica*, 12:125–134, 1992.
31. N. Alon, R. Yuster, and U. Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs, In *Proc. 26th Annual ACM Symp. Theory Computing*, pages 326–335, Montreal, Canada, 1994, ACM Press.
32. N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. Assoc. Comput. Mach.*, 42:844–856, 1995.
33. N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17:209–223, 1997.
34. B. Alspach. Cycles of each length in regular tournaments. *Can. Math. Bull.*, 10:283–285, 1967.
35. B. Alspach. On point-symmetric tournaments. *Can. Math. Bull.*, 13:317–323, 1970.
36. B. Alspach, J.-C. Bermond, and D. Sotteau. Decomposition into cycles. I. Hamilton decompositions. In *Cycles and rays (Montreal, PQ, 1987)*, pages 9–18. Kluwer, Dordrecht, 1990.
37. B. Alspach and M. Rosenfeld. Realization of certain generalized paths in tournaments. *Discrete Math.*, 34:199–202, 1981.
38. B. Alspach and C. Tabib. A note on the number of 4-circuits in a tournament. In *Theory and practice of combinatorics*, volume 60 of *North-Holland Math. Stud.*, pages 13–19. North-Holland, Amsterdam, 1982.
39. H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Inf. Process. Lett.*, 37(4):237–240, 1991.
40. D. Amar and A. Raspaud. Covering the vertices of a digraph by cycles of prescribed length. *Discrete Math.*, 87:111–118, 1991.
41. A. Andersson. Sublogarithmic Searching without Multiplications. In *Proc. 36th Symposium on Foundations of Computer Science*, pages 655–663. IEEE Computer Society Press, 1995.

42. A. Apartsin, E. Ferapontova, and V. Gurvich. A circular graph - counterexample to the Duchet kernel conjecture. *Discrete Math.*, 178:229–231, 1998.
43. K. Appel and W. Haken. Every planar map is four colorable. *Bull. Amer. Math. Soc.*, 82(5):711–712, 1976.
44. D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Implementing the Dantzig-Fulkerson-Johnson algorithm for large salesman problems. *Math. Program. Ser. B*, 97(1-2):91–153, 2003.
45. H. Ariyoshi. Feedback arc sets of directed star polygons, In D.E. Kirk, editor, *IEEE Conf. Record of the 14th Asilomar Conf. on Circuits, Systems and Computers*, pages 55–59, IEEE Computer Soc., New York, 1981.
46. E. M. Arkin and R. Hassin. A note on orientations of mixed graphs. *Discrete Appl. Math.*, 116(3):271–278, 2002.
47. E. M. Arkin, R. Hassin, and S. Shahar. Increasing digraph arc-connectivity by arc addition, reversal and complement. *Discrete Appl. Math.*, 122(1-3):13–22, 2002.
48. E.M. Arkin and C.H. Papadimitriou. On negative cycles in mixed graphs. *Oper. Res. Lett.*, 4:113–116, 1985.
49. E.M. Arkin, C.H. Papadimitriou, and M. Yannakakis. Modularity of cycles and paths in graphs. *J. Assoc. Comput. Mach.*, 31:255–274, 1991.
50. S. Arora, A. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *Proc. 37th Annual Symposium on Foundations of Computer Science*, pages 24–33. IEEE Computer Society Press, 1996.
51. E.H.L. Arts and J.H.M. Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, Chicester, 1989.
52. A.A. Assad. Multicommodity network flows – A survey. *Networks*, 8:37–91, 1978.
53. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer-Verlag, Berlin, 1999.
54. J. N. Ayoub and I.T. Frisch. Optimally invulnerable directed communication networks. *IEEE Trans. Commun. Technol.*, 18:484–489, 1970.
55. L. Baffi and R. Petreschi. Parallel maximal matching on minimal vertex series parallel digraphs. In *Algorithms, concurrency and knowledge (Pathumthani, 1995)*, pages 34–47. Springer-Verlag, 1995.
56. A. Bagchi, A. Bhargava, and T. Suel. Approximate maximum weight branchings. *Inf. Process. Lett.*, 99(2):54–58, 2006.
57. R.C. Baker, G. Harman, and J. Pintz. The difference between consecutive primes, II. *Proc. London Math. Soc.*, 83:532–562, 2001.
58. E. Balas and M.J. Saltzman. An algorithm for the three-index assignment problem. *Oper. Res.*, 39:150–161, 1991.
59. Y. Balcer and A.F. Veinott. Computing a graph's period quadratically by node condensation. *Discrete Math.*, 4:295–303, 1973.
60. M. Balinski and G. Ratier. On stable marriages and graphs, and strategy and polytopes. *SIAM Rev.*, 39(4):575–604, 1997.
61. M. Balinski and G. Ratier. Graphs and marriages. *Amer. Math. Mon.*, 105(5):430–445, 1998.
62. P. Balister, S. Gerke, and G. Gutin. Convex sets in acyclic digraphs. Preprint arXiv:0712.2678v1, December 2007.
63. P. Balister, S. Gerke, G. Gutin, A. Johnstone, J. Reddington, E. Scott, A. Soleimanfallah, and A. Yeo. Algorithms for Generating Convex Sets in Acyclic Digraphs. *J. Discrete Algorithms*, to appear:10 pp., 2008.

64. E. Bampis, P. Hell, Y. Manoussakis, and M. Rosenfeld. Finding an antidirected hamiltonian path starting with a forward arc from a given vertex in a tournament. In *Proc. 8th Franco-Japanese and 4th Franco-Chinese Conference on Combinatorics and Computer Science*, volume 1120 of *Lect. Notes Comput. Sci.*, pages 67–73. Springer, 1995.
65. J. Bang-Jensen. On the 2-linkage problem for semicomplete digraphs. In *Graph theory in memory of G. A. Dirac (Sandbjerg, 1985)*, volume 41 of *Ann. Discrete Math.*, pages 23–37. North-Holland, 1989.
66. J. Bang-Jensen. Locally semicomplete digraphs: a generalization of tournaments. *J. Graph Theory*, 14(3):371–390, 1990.
67. J. Bang-Jensen. A note on a special case of the 2-path problem for semicomplete digraphs. In *Graph theory, combinatorics, and applications, Vol. 1 (Kalamazoo, MI, 1988)*, pages 77–86. Wiley, 1991.
68. J. Bang-Jensen. Edge-disjoint in- and out-branchings in tournaments and related path problems. *J. Combin. Theory Ser. B*, 51(1):1–23, 1991.
69. J. Bang-Jensen. On the structure of locally semicomplete digraphs. *Discrete Math.*, 100(1-3):243–265, 1992.
70. J. Bang-Jensen. Arc-local tournament digraphs: a generalization of tournaments and bipartite tournaments, Technical report 2, Department of Mathematics and Computer Science, Odense University, Denmark, 1993.
71. J. Bang-Jensen. A reformulation of Huang's structure theorem for local tournaments with some consequences, Technical report 13, Department of Mathematics and Computer Science, Odense University, Denmark, 1994.
72. J. Bang-Jensen. Digraphs with the path-merging property. *J. Graph Theory*, 20(2):255–265, 1995.
73. J. Bang-Jensen. Disjoint Paths with Prescribed Ends and Cycles through Given Arcs in Locally Semicomplete Digraphs and Quasi-Transitive Digraphs, Technical Report 22, Department of Mathematics and Computer Science, Odense University, Denmark, 1996.
74. J. Bang-Jensen. Linkages in locally semicomplete digraphs and quasi-transitive digraphs. *Discrete Math.*, 196(1-3):13–27, 1999.
75. J. Bang-Jensen. The structure of strong arc-locally semicomplete digraphs. *Discrete Math.*, 283(1-3):1–6, 2004.
76. J. Bang-Jensen. Problems and conjectures concerning connectivity, paths, trees and cycles in tournament-like digraphs. *Discrete Math.*, to appear.
77. J. Bang-Jensen and S. Brandt. Expansion and hamiltonicity in digraphs. Submitted.
78. J. Bang-Jensen, A. Frank, and B. Jackson. Preserving and increasing local edge-connectivity in mixed graphs. *SIAM J. Discrete Math.*, 8:155–178, 1995.
79. J. Bang-Jensen and Y. Guo. A note on vertex pancyclic oriented graphs. *J. Graph Theory*, 31:313–318, 1999.
80. J. Bang-Jensen, Y. Guo, G. Gutin, and L. Volkmann. A classification of locally semicomplete digraphs. *Discrete Math.*, 167/168:101–114, 1997. 15th British Combinatorial Conference (Stirling, 1995).
81. J. Bang-Jensen, Y. Guo, and L. Volkmann. Weakly Hamiltonian-connected locally semicomplete digraphs. *J. Graph Theory*, 21(2):163–172, 1996.
82. J. Bang-Jensen, Y. Guo, and A. Yeo. A new sufficient condition for a digraph to be Hamiltonian. *Discrete Appl. Math.*, 95:61–72, 1999.
83. J. Bang-Jensen, Y. Guo, and A. Yeo. Complementary cycles containing prescribed vertices in tournaments. *Discrete Math.*, 214:77–87, 2000.
84. J. Bang-Jensen and G. Gutin. Paths, trees and cycles in tournaments. *Congr. Numer.*, 115:131–170, 1996. Surveys in graph theory (San Francisco, 1995).

85. J. Bang-Jensen and G. Gutin. Alternating paths and cycles in edge-coloured multigraphs: a survey. *Discrete Math.*, 165-166:39–60, 1997.
86. J. Bang-Jensen and G. Gutin. Paths and cycles in extended and decomposable digraphs. *Discrete Math.*, 164(1-3):5–19, 1997. The Second Krakow Conference on Graph Theory (Zgorzelisko, 1994).
87. J. Bang-Jensen and G. Gutin. Vertex heaviest paths and cycles in quasi-transitive digraphs. *Discrete Math.*, 163(1-3):217–223, 1997.
88. J. Bang-Jensen and G. Gutin. Alternating cycles and trails in 2-edge-coloured complete multigraphs. *Discrete Math.*, 188:61–72, 1998.
89. J. Bang-Jensen and G. Gutin. Generalizations of tournaments: A survey. *J. Graph Theory*, 28:171–202, 1998.
90. J. Bang-Jensen and G. Gutin. On the complexity of hamiltonian path and cycle problems in certain classes of digraphs. *Discrete Appl. Math.*, 95:41–60, 1999.
91. J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, London, 2000.
92. J. Bang-Jensen, G. Gutin, and J. Huang. Weakly Hamiltonian-connected ordinary multipartite tournaments. *Discrete Math.*, 138(1-3):63–74, 1995. 14th British Combinatorial Conference (Keele, 1993).
93. J. Bang-Jensen, G. Gutin, and J. Huang. A sufficient condition for a semicomplete multipartite digraph to be Hamiltonian. *Discrete Math.*, 161(1-3):1–12, 1996.
94. J. Bang-Jensen, G. Gutin, and H. Li. Sufficient conditions for a digraph to be Hamiltonian. *J. Graph Theory*, 22(2):181–187, 1996.
95. J. Bang-Jensen, G. Gutin, and A. Yeo. On k-strong and k-cyclic digraphs. *Discrete Math.*, 162(1-3):1–11, 1996.
96. J. Bang-Jensen, G. Gutin, and A. Yeo. Hamiltonian cycles avoiding prescribed arcs in tournaments. *Combin. Probab. Comput.*, 6(3):255–261, 1997.
97. J. Bang-Jensen, G. Gutin, and A. Yeo. A polynomial algorithm for the Hamiltonian cycle problem in semicomplete multipartite digraphs. *J. Graph Theory*, 29:111–132, 1998.
98. J. Bang-Jensen, G. Gutin, and A. Yeo. Steiner type problems for digraphs that are locally semicomplete or extended semicomplete. *J. Graph Theory*, 44(3):193–207, 2003.
99. J. Bang-Jensen, G. Gutin, and A. Yeo. When the greedy algorithm fails. *Discrete Optim.*, 1:121–127, 2004.
100. J. Bang-Jensen, G. Gutin, and A. Yeo. Finding a cheapest cycle in a quasi-transitive digraph with real-valued vertex costs. *Discrete Optim.*, 3:86–94, 2006.
101. J. Bang-Jensen and P. Hell. Fast algorithms for finding Hamiltonian paths and cycles in in-tournament digraphs. *Discrete Appl. Math.*, 41(1):75–79, 1993.
102. J. Bang-Jensen, P. Hell, and J. Huang. Optimal recognition of local tournaments. *Congr. Numer.*, 100:141–146, 1994.
103. J. Bang-Jensen and J. Huang. Quasi-transitive digraphs. *J. Graph Theory*, 20(2):141–161, 1995.
104. J. Bang-Jensen and J. Huang. Kings in quasi-transitive digraphs. *Discrete Math.*, 185(1-3):19–27, 1998.
105. J. Bang-Jensen, J. Huang, and E. Prisner. In-tournament digraphs. *J. Combin. Theory Ser. B*, 59(2):267–287, 1993.
106. J. Bang-Jensen, J. Huang, and A. Yeo. Strongly connected spanning subgraphs with the minimum number of arcs in quasi-transitive digraphs. *SIAM J. Discrete Math.*, 16:335–343, 2003.

107. J. Bang-Jensen, J. Huang, and A. Yeo. Spanning k-arc-strong subdigraphs with few arcs in k-arc-strong tournaments. *J. Graph Theory*, 46(4):265–284, 2004.
108. J. Bang-Jensen and T. Jordán. Spanning 2-strong subtournaments in 3-strong semicomplete digraphs. Unpublished manuscript, November 1995. Department of Mathematics and Computer Science, Odense University, Denmark.
109. J. Bang-Jensen and T. Jordán. Adding and reversing arcs in semicomplete digraphs. *Combin. Probab. Comput.*, 7(1):17–25, 1998.
110. J. Bang-Jensen and Y. Manoussakis. Weakly Hamiltonian-connected vertices in bipartite tournaments. *J. Combin. Theory Ser. B*, 63(2):261–280, 1995.
111. J. Bang-Jensen, Y. Manoussakis, and C. Thomassen. A polynomial algorithm for Hamiltonian-connectedness in semicomplete digraphs. *J. Algor.*, 13(1):114–127, 1992.
112. J. Bang-Jensen and M. H. Nielsen. Finding complementary cycles in locally semicomplete digraphs. *Discrete Appl. Math.*, 146(3):245–256, 2005.
113. J. Bang-Jensen and M.H. Nielsen. Minimum cycle factors in quasi-transitive digraphs. *Discrete Optim.*, 5:121–137, 2008.
114. J. Bang-Jensen, M.H. Nielsen, and A. Yeo. Longest path partitions in generalizations of tournaments. *Discrete Math.*, 306(16):1830–1839, 2006.
115. J Bang-Jensen and S. Poljak. Eulerian trails through a set of terminals in specific, unique and all orders. *Contemp. Math.*, 147:247–258, 1993.
116. J. Bang-Jensen and S. Thomassé. Highly connected hypergraphs containing no two edge-disjoint spanning connected subhypergraphs. *Discrete Appl. Math.*, 131(2):555–559, 2003.
117. J. Bang-Jensen, S. Thomassé, and A. Yeo. Small degree out-branchings. *J. Graph Theory*, 42(4):297–307, 2003.
118. J. Bang-Jensen and C. Thomassen. A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM J. Discrete Math.*, 5:366–376, 1992.
119. J. Bang-Jensen and A. Yeo. The minimum spanning strong subdigraph problem for extended semicomplete digraphs and semicomplete bipartite digraphs. *J. Algor.*, 41(1):1–19, 2001.
120. J. Bang-Jensen and A. Yeo. Decomposing k-arc-strong tournaments into strong spanning subdigraphs. *Combinatorica*, 24(3):331–349, 2004.
121. J. Bang-Jensen and A. Yeo. Making a tournament k-arc-strong by reversing or deorienting arcs. *Discrete Appl. Math.*, 136(2-3):161–171, 2004. The 1st Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW 2001).
122. J. Bang-Jensen and A. Yeo. The minimum spanning subdigraph problem is fixed parameter tractable. *Discrete Appl. Math.*, to appear.
123. M. Bankfalvi and Zs. Bankfalvi. Alternating hamiltonian circuit in two-coloured complete graphs. In *Proc. Colloq. Tihany 1968*, pages 11–18. Academic Press, 1968.
124. M. Bárász, J. Becker, and A. Frank. An algorithm for source location in directed graphs. *Oper. Res. Lett.*, 33:221–230, 2005.
125. J.-P. Barthélémy, O. Hudry, G. Isaak, F.S. Roberts, and B. Tesman. The reversing number of a digraph. *Discrete Appl. Math.*, 60(1-3):39–76, 1995. ARIDAM VI and VII (New Brunswick, NJ, 1991/1992).
126. E.T. Baskoro, M. Miller, J. Plesník, and S. Znám. Digraphs of degree 3 and order close to the Moore bound. *J. Graph Theory*, 20:339–349, 1995.
127. G. Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing*. Prentice Hall, Englewood Cliffs, NJ, 1999.
128. E.B. Baum Iterated descent: A better algorithm for local search in combinatorial optimization problems, 1986. Unpublished manuscript.

129. J. Beck. On 3-chromatic hypergraphs. *Discrete Math.*, 24(2):127–137, 1978.
130. R. Beigel and D. Eppstein. 3-coloring in time $O(1.3289^n)$. *J. Algor.*, 54(2):168–204, 2005.
131. L.W. Beineke and C.H.C. Little. Cycles in bipartite tournaments. *J. Combin. Theory Ser. B*, 32(2):140–145, 1982.
132. L.W. Beineke and M.D. Plummer. On the 1-factors of a nonseparable graph. *J. Combin. Theory Ser. B*, 2:285–289, 1967.
133. L.W. Beineke and C.M. Zamfirescu. Connection digraphs and second order line graphs. *Discrete Math.*, 39:237–254, 1982.
134. R.E. Bellman. On a routing problem. *Quart. Appl. Math.*, 16:87–90, 1958.
135. D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo, and A. Zverovich. Transformations of Generalized ATSP into ATSP: experimental and theoretical study. *Oper. Res. Lett.*, 31:357–365, 2003.
136. A. Benhocine and A.P. Wojda. On the existence of specified cycles in a tournament. *J. Graph Theory*, 7:469–473, 1983.
137. A. Benkouar, Y. Manoussakis, V. Paschos, and R. Saad. On the Complexity of Some Hamiltonian and Eulerian Problems in Edge-Colored Complete Graphs. In *Proc. the 2nd International Symposium on Algorithms*, volume 557 of *Lect. Notes Comput. Sci.*, pages 190–198. Springer-Verlag, 1991.
138. M.D. Bennett. Nucleotypic basis of the spacial ordering of chromosomes in eucariotes and the implications of the order for genome and phenotypic variation. In *Genome Evolution*, pages 239–261. Academic Press, London, 1982.
139. D. Berend, S. Skiena, and Y. Twitto. Combinatorial dominance guarantees for heuristic algorithms, In *Proc. International Conference on Analysis of Algorithms*, 2007.
140. A. R. Berg, B. Jackson, and T. Jordán. Edge splitting and connectivity augmentation in directed hypergraphs. *Discrete Math.*, 273(1-3):71–84, 2003. EuroComb'01 (Barcelona).
141. A. R. Berg and T. Jordán. Minimally k-edge-connected directed graphs of maximal size. *Graphs Combin.*, 21(1):39–50, 2005.
142. A. R. Berg and T. Jordán. Two-connected orientations of Eulerian graphs. *J. Graph Theory*, 52(3):230–242, 2006.
143. C. Berge. *Graphs and Hypergraphs*. North-Holland, Amsterdam, 2nd edition, 1976.
144. C. Berge. *Graphs*. North-Holland, Amsterdam, 1985. Second revised edition of part 1 of the 1973 English version.
145. C. Berge and P. Duchet. Recent problems and results about kernels in directed graphs. *Discrete Math.*, 86(1-3):27–31, 1990.
146. C. Berge and A.R. Rao. A combinatorial problem in logic. *Discrete Math.*, 17:23–26, 1977.
147. K.A. Berman and X. Liu. Cycles through large degree vertices in digraphs: a generalization of Meyniel's theorem. *J. Combin. Theory Ser. B*, 74:20–27, 1998.
148. J.-C. Bermond, A. Germa, M.-C. Heydemann, and D. Sotteau. Girth in digraphs. *J. Graph Theory*, 4(3):337–341, 1980.
149. J.-C. Bermond and P. Hell. On even factorizations and the chromatic index of the Kautz and de Bruijn digraphs. *J. Graph Theory*, 17:647–655, 1993.
150. J.-C. Bermond, X. Munos, and A. Marchetti-Spaccamela. A Broadcasting Protocol in Line Digraphs. *J. Parallel Distributed Comput.*, 61(8):1013–1032, 2001.
151. J.-C. Bermond and C. Peyrat. De Bruijn and Kautz networks: A competitor for the hypercube? In F. André and J.P. Verjus, editors, *Hypercube and distributed computers*, pages 279–493. Elsevier, North-Holland, 1989.

152. J.-C. Bermond and C. Thomassen. Cycles in digraphs—a survey. *J. Graph Theory*, 5(1):1–43, 1981.
153. P. Bertolazzi, R.F. Cohen, G. Di Battista, R. Tamassia, and I.G. Tollis. How to draw a series-parallel digraph. *Int. J. Comput. Geom. Appl.*, 4(4):385–402, 1994.
154. D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-width and parity games. In *STACS 2006, Proc. 23rd Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lect. Notes Comput. Sci.*, pages 524–436. Springer-Verlag, 2006.
155. S. Bessy. *Stabilité et décomposition en circuits d'un digraphe*, Phd thesis, Université Lyon 1, December 2003.
156. S. Bessy. Paths partition with prescribed beginnings in digraphs: A Chvátal Erdős condition approach. *Discrete Math.*, 2007.
157. S. Bessy and S. Thomassé. Every strong digraph has a spanning strong subgraph with at most $n+2\alpha-2$ arcs. *J. Combin. Theory Ser. B*, 87(2):289–299, 2003.
158. S. Bessy and S. Thomassé. Spanning a digraph by $\alpha(D)$ circuits: a proof of Gallai's conjecture. *Combinatorica*, 27:659–667, 2007.
159. W. Bienia, L. Goddyn, P. Gvozdjak, A. Sebö, and M. Tarsi. Flows, view obstructions, and the lonely runner. *J. Combin. Theory Ser. B*, 72(1):1–9, 1998.
160. N.L. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory 1736-1936*. Clarendon Press, London, 1976.
161. H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25:1305–1317, 1996.
162. F. Boesch and R. Tindell. Robbins's theorem for mixed multigraphs. *Amer. Math. Mon.*, 87(9):716–719, 1980.
163. B. Bollobás and P. Erdős. Alternating Hamiltonian cycles. *Israel J. Math.*, 23:126–131, 1976.
164. B. Bollobás, D.L. Goldsmith, and D.R. Woodall. Indestructive deletions of edges from graphs. *J. Combin. Theory Ser. B*, 30(3):263–275, 1981.
165. B. Bollobás and R. Häggkvist. Powers of Hamilton cycles in tournaments. *J. Combin. Theory Ser. B*, 50(2):309–318, 1990.
166. J. A. Bondy. Short proofs of classical theorems. *J. Graph Theory*, 44(3):159–165, 2003.
167. J.A. Bondy. Diconnected orientations and a conjecture of Las Vergnas. *J. London Math. Soc. (2)*, 14(2):277–282, 1976.
168. J.A. Bondy. A short proof of the Chen-Manalastas theorem. *Discrete Math.*, 146(1-3):289–292, 1995.
169. J.A. Bondy. Basic graph theory: paths and circuits. In *Handbook of combinatorics, Vol. 1, 2*, pages 3–110. Elsevier, Amsterdam, 1995.
170. J.A. Bondy and U.S.R. Murty. *Graph theory with applications*. American Elsevier Publishing Co., New York, 1976.
171. J.A. Bondy and C. Thomassen. A short proof of Meyniel's theorem. *Discrete Math.*, 19(2):195–197, 1977.
172. J.A. Bondy and A. Vince. Cycles in a graph whose lengths differ by one or two. *J. Graph Theory*, 27(1):11–15, 1998.
173. P. Bonsma and F. Dorn. An FPT Algorithm for Directed Spanning k-Leaf. Preprint arXiv:0711.4052, November 2007.
174. P. Bonsma and F. Dorn. Tight Bounds and Faster Algorithms for Directed Max-Leaf Problems. In *Proc. ESA'08*, volume to appear of *Lect Notes Computer Sci.* Springer, 2008.

175. E. Boros and V. Gurvich. Perfect graphs are kernel-solvable. *Discrete Math.*, 159:35–55, 1996.
176. E. Boros and V. Gurvich. A corrected version of the Duchet kernel conjecture. *Discrete Math.*, 179(1-3):231–233, 1998.
177. E. Boros and V. Gurvich. Perfect, Partitionable, and Kernel-Solvable Graphs. In *The Perfect Graph Conjecture*. American Institute of Mathematics, Palo Alto, www.aimath.org/pastworkshops/perfectgraph.html, Oct-Nov 2002.
178. A. Brandstädt. *Graphen und Algorithmen*. B. G. Teubner, Stuttgart, 1994.
179. S. Brandt, H. Broersma, R. Diestel, and M. Kriesell. Global connectivity and expansion: long cycles and factors in f-connected graphs. *Combinatorica*, 26:17–36, 2006.
180. G. Brassard and P. Bratley. *Fundamentals of algorithmics*. Prentice Hall Inc., Englewood Cliffs, NJ, 1996.
181. W.G. Bridges and S. Toueg. On the impossibility of directed Moore graphs. *J. Combin. Theory Ser. B*, 29:339–341, 1980.
182. D.E. Brown, A.H. Busch, and J.R. Lundgren. Interval tournaments. *J. Graph Theory*, 56:72–81, 2007.
183. S.A. Burr. Subtrees of directed graphs and hypergraphs. *Congr. Numer.*, 28:227–239, 1980.
184. R.G. Busacker and P.J. Gowen. A procedure for determining a family of minimal cost network flow patterns, Technical Report 15, ORO Tech. Report, Johns Hopkins University, 1961.
185. A.H. Busch, M.S. Jacobson, and K.B. Reid. On arc-traceable tournaments. *J. Graph Theory*, 53:157–166, 2006.
186. L. Caccetta and R. Häggkvist. On minimal digraphs with given girth. *Congr. Numer.*, 21:181–187, 1978.
187. M. Cai, X. Deng, and L. Wang. Minimum k arborescences with bandwidth constraints. *Algorithmica*, 38:529–537, 2004.
188. K. Cameron. *Polyhedral and algorithmic ramifications of antichains*, PhD thesis, University of Waterloo, 1982.
189. K. Cameron. On Gallai-type min-max inequalities. In *Combinatorics, graph theory, algorithms and applications (Beijing, 1993)*, pages 7–16. World Sci., River Edge, NJ, 1994.
190. K. Cameron and J. Edmonds. Coflow polyhedra. *Discrete Math.*, 101:1–21, 1992.
191. K. Cameron and J. Edmonds. The travelling preacher, projection, and a lower bound for the stability number of a graph. *Discrete Optim.*, 5:290–292, 2008.
192. P. Camion. Chemins et circuits hamiltoniens des graphes complets. *C. R. Acad. Sci. Paris*, 249:2151–2152, 1959.
193. A. Cayley. A theorem on trees. *Quart. J. Pure Appl. Math.*, 23:376–378, 1889.
194. G.J. Chang, F.K. Hwang, and L.D. Tong. The Hamiltonian property of the consecutive-3 digraphs. *Math. Comput. Modelling*, 25:83–88, 1997.
195. G.J. Chang, F.K. Hwang, and L.D. Tong. The consecutive-4 digraphs are Hamiltonian. *J. Graph Theory*, 31:1–6, 1999.
196. P. Charbit and A. Sebö. Cyclic orders: equivalence and duality. *Combinatorica*, 28:131–143, 2008.
197. P. Charbit, S. Thomassé, and A. Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combin. Probab. Comput.*, 16:1–4, 2007.
198. M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *J. Algor.*, 33:73–91, 1999.
199. G. Chartrand, D. Geller, and S. Hedetniemi. Graphs with forbidden subgraphs. *J. Combin. Theory Ser. B*, 10:12–41, 1971.

200. C.C. Chen and D.E. Daykin. Graphs with Hamiltonian cycles having adjacent lines of different colors. *J. Combin. Theory Ser. B*, 21:135–139, 1976.
201. C.C. Chen and P. Manalastas, Jr. Every finite strongly connected digraph of stability 2 has a Hamiltonian path. *Discrete Math.*, 44(3):243–250, 1983.
202. G. Chen, J. Shen, and R. Yuster. Second neighborhood via first neighborhood in digraphs. *Ann. Combin.*, 7(1):15–20, 2003.
203. G.-T. Chen, R.J. Gould, and H. Li. Partitioning vertices of a tournament into independent cycles. *J. Combin. Theory Ser. B*, 83(2):213–220, 2001.
204. J. Chen, I.A. Kanj, and G. Xia. Simplicity is beauty: Improved upper bounds for vertex cover, Technical Report TR05-008, DePaul University, Chicago, 2005.
205. J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. Directed Feedback Vertex Set is Fixed-Parameter Tractable. In *Proc. STOC'08: the 40th Annual ACM Symposium on Theory of Computing*. ACM Press, 2008.
206. E. Cheng and T. Jordán. Successive edge-connectivity augmentation problems. *Math. Program. Ser. B*, 84:577–593, 1999.
207. J. Cheriyan and S.N. Maheshwari. Analysis of preflow push algorithms for maximum network flow. *SIAM J. Comput.*, 18:1057–1086, 1989.
208. J. Cheriyan and J.H. Reif. Directed *s-t* numberings, rubber bands, and testing digraph *k*-vertex connectivity. *Combinatorica*, 14(4):435–451, 1994.
209. J. Cheriyan and R. Thurimella. Approximating minimum-size *k*-connected spanning subgraphs via matching (extended abstract). In *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996)*, pages 292–301. IEEE Computer Society Press, 1996.
210. J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM J. Comput.*, 30(2):528–560, 2000.
211. B.V. Cherkassky and A.V. Goldberg. Negative-cycle detection algorithms. *Math. Program.*, 85:277–311, 1999.
212. B.V. Cherkassky, A.V. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. *Math. Program.*, 73:129–174, 1996.
213. A.G. Chetwynd and A.J.W. Hilton. Alternating Hamiltonian cycles in two colored complete bipartite graphs. *J. Graph Theory*, 16:153–158, 1992.
214. W.S. Chow, Y. Manoussakis, O. Megalakaki, M. Spyratos, and Zs. Tuza. Paths through fixed vertices in edge-colored graphs. *Math. Inf. Sci. Hum.*, 127(32):49–58, 1994.
215. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem, Technical Report CS-93-13, Carnegie Mellon University, 1976.
216. M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Ann. Math.*, 164:51–229, 2006.
217. M. Chudnovsky, P. Seymour, and B. Sullivan. Cycle in dense digraphs. Preprint arXiv:math/0702147v1, December 2007.
218. F.R.K. Chung, M.R. Garey, and R.E. Tarjan. Strongly connected orientations of mixed multigraphs. *Networks*, 15(4):477–484, 1985.
219. F.R.K. Chung, W. Goddard, and D.J. Kleitman. Even cycles in directed graphs. *SIAM J. Discrete Math.*, 7(3):474–483, 1994.
220. V. Chvátal. On Hamilton's ideals. *J. Combin. Theory Ser. B*, 12:163–168, 1972.
221. V. Chvátal and P. Erdős. A note on Hamiltonian circuits. *Discrete Math.*, 2:111–113, 1972.
222. V. Chvátal and L. Lovász. Every directed graph has a semi-kernel. *Lect. Notes Math.*, 411:175, 1974.
223. V. Chvátal and E. Szemerédi. Short cycles in directed graphs. *J. Combin. Theory Ser. B*, 35(3):323–327, 1983.

224. V. Chvátal and C. Thomassen. Distances in orientations of graphs. *J. Combin. Theory Ser. B*, 24(1):61–75, 1978.
225. S. Climer and W. Zhang. Cut-and-solve: an iterative search strategy for combinatorial optimization problems. *Artif. Intell.*, 170:714–738, 2006.
226. T. Coleman and A. Wirth. Ranking tournaments: local search and a new algorithm. In *Proc. ALENEX'08: 10th Workshop on Algorithm Engineering and Experimentation*, pages 133 – 141. SIAM, 2008.
227. V. Conitzer. Computing Slater rankings using similarities among candidates, In *21st National Conference on Artificial Intelligence (AAAI-06)*, pages 613–619, 2006.
228. S.A. Cook. The complexity of theorem-proving procedures, In *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, pages 151–158, 1971.
229. W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, New York, 1998.
230. D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions, In *Proc. 19th Ann. ACM Symp. on Theory of Computation*, pages 1–6, ACM Press, 1987.
231. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to algorithms*, *The MIT Electrical Engineering and Computer Science Series*. MIT Press, Cambridge, MA, 1990.
232. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.
233. B. Csaba. On the Bollobás-Eldridge conjecture for bipartite graphs. *Combin. Probab. Comput.*, 16:661–691, 2007.
234. W.H. Cunningham and A. Frank. A primal-dual algorithm for submodular flows. *Math. Oper. Res.*, 10(2):251–262, 1985.
235. S.J. Curran and D. Witte. Hamilton paths in Cartesian products of directed cycles. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 35–74. North-Holland, 1985.
236. A. Czumaj and W.B. Strothmann. Bounded degree spanning trees. In *Algorithms–ESA'97*, volume 1284 of *Lect. Notes Comput. Sci.*, pages 104–117. Springer, 1997.
237. G. Dahl. Directed Steiner problems with connectivity constraints. *Discrete Appl. Math.*, 47:109–128, 1993.
238. V. Dahllöf, P. Jonsson, and M. Wahlström. Counting models for 2SAT and 3SAT formulae. *Theor. Comput. Sci.*, 332(1-3):265–291, 2005.
239. M. Dalmazzo. Nombre d'arcs dans les graphes k-arc-fortement connexes minimaux. *C.R. Acad. Sci. Paris A*, 2853:341–344, 1977.
240. P. Dankelmann. The diameter of directed graphs. *J. Combin. Theory Ser. B*, 94(1):183–186, 2005.
241. P. Dankelmann, G. Gutin, and E.J. Kim. On Complexity of Minimum Leaf Out-Branching Problem. Preprint arXiv:0808.0980, August 2008.
242. S.K. Darbinyan. Cycles of any length in digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 75(4):147–152, 1982.
243. S.K. Darbinyan. Pancyclicity of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 80(2):51–54, 1985.
244. S.K. Darbinyan. Pancyclicity of digraphs with the Meyniel condition. *Stud. Sci. Math. Hung.*, 20(1-4):95–117, 1985.
245. S.K. Darbinyan. A sufficient condition for the Hamiltonian property of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 82(1):6–8, 1986.
246. S.K. Darbinyan. On the pancyclicity of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 83(3):99–101, 1986.

247. S.K. Darbinyan. Hamiltonian and strongly Hamilton-connected digraphs. *Akad. Nauk Armyan. SSR Dokl.*, 91(1):3–6, 1990.
248. S.K. Darbinyan. On hamiltonian bypasses in digraphs satisfying Meyniel-like conditions (in Russian). *Math. Probl. Comput. Sci.*, 20:7–19, 1998.
249. M. Darrah, Y.-P. Liu, and C.-Q. Zhang. Cycles of all lengths in arc-3-cyclic semicomplete digraphs. *Discrete Math.*, 173(1-3):23–33, 1997.
250. P. Das. Pan-alternating cyclic edge-partitioned graphs. *Ars Combin.*, 14:105–114, 1982.
251. D.E. Daykin. Graphs with cycles having adjacent lines of different colors. *J. Combin. Theory Ser. B*, 20:149–152, 1976.
252. N.G. de Bruijn. A combinatorial problem. *Ned. Akad. Wet. Proc.*, 49:758–764, 1946.
253. B. de Fluiter and H.L. Bodlaender. Parallel algorithms for treewidth two. In *Workshop on Graph-Theoretic Concepts in Computer Science*, volume 1335 of *Lect. Notes Comput. Sci.*, pages 157–170. Springer, 1997.
254. W.F. de la Vega. On the maximum cardinality of a consistent set of arcs in a random tournament. *J. Combin. Theory Ser. B*, 35:328–332, 1983.
255. N. Dean and B.J. Latka. Squaring the tournament—an open problem. *Congr. Numer.*, 109:73–80, 1995.
256. A. Demers and A. Downing Minimum leaf spanning tree. US Patent no. 6,105,018, August 2000.
257. X. Deng, P. Hell, and J. Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM J. Computing*, 25(2):390–403, 1996.
258. R. Diestel. *Graph theory*. Springer-Verlag, New York, 2nd edition, 2000.
259. E.W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math.*, 1:269–271, 1959.
260. R.P. Dilworth. A decomposition theorem for partially ordered sets. *Ann. Math.*, 51:161–166, 1950.
261. G. Ding, A. Schrijver, and P.D. Seymour. Disjoint paths in a planar graph – a general theorem. *SIAM J. Discrete Math.*, 5(1):112–116, 1992.
262. E.A. Dinic. An algorithm for the solution of the problem of maximal flow in a network with power estimation. *Dokl. Akad. Nauk SSSR*, 194:754–757, 1970.
263. E.A. Dinits and A.V. Karzanov. On the existence of two edge-disjoint chains in multi-graph connecting given pairs of its vertices. *Graph Theory Newsletters*, 8:2–3, 1979.
264. E.A. Dinits and A.V. Karzanov. On two integer flows of value 1. In A.V. Karzanov, editor, *Combinatorial methods for network flow problems*, pages 127–137. Institute for System Studies, Moscow, 1979.
265. I. Dinur and S. Safra. The importance of being biased. In *Proc. STOC'02: 34th Annual ACM Symposium on Theory of Computing*, pages 33–42. ACM Press, 2002.
266. G.A. Dirac. Some theorems on abstract graphs. *Proc. London Math. Soc.*, 2(3):69–81, 1952.
267. A. Dolan and J. Aldous. *Networks and algorithms*. John Wiley & Sons, Chichester, 1993. An introductory approach.
268. J. Donadelli and Y. Kohayakawa. A density result for random sparse oriented graphs and its relation to a conjecture of Woodall. *Electron. J. Combin.*, 9(1):Research Paper 45, 10 pp. (electronic), 2002.
269. E.A. van Doorn. Connectivity of circulant digraphs. *J. Graph Theory*, 10(1):9–14, 1986.
270. D. Dorninger. On permutations of chromosomes. In *Contributions to General Algebra*, volume 5, pages 95–103. Teubner-Verlag, Stuttgart, 1987.

271. D. Dorninger. Hamiltonian circuits determining the order of chromosomes. *Discrete Appl. Math.*, 50:159–168, 1994.
272. D. Dorninger and W. Timischl. Geometrical constraints on Bennett's predictions of chromosome order. *Heredity*, 58:321–325, 1987.
273. R.J. Douglas. Tournaments that admit exactly one Hamiltonian circuit. *Proc. London Math. Soc. (3)*, 21:716–730, 1970.
274. R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.
275. K. Dowsland. Simulated Annealing. In C.R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 20–69. McGraw-Hill, 1995.
276. D.-Z. Du, F. Cao, and D.F. Hsu. De Bruijn digraphs, and Kautz digraphs, and their generalizations. In D.-Z. Du and D.F. Hsu, editors, *Combinatorial network theory*, pages 65–105. Kluwer, Dordrecht, 1996.
277. D.-Z. Du and D.F. Hsu. On Hamiltonian consecutive-d digraphs. *Banach Center Publ.*, 25:47–55, 1989.
278. D.-Z. Du, D.F. Hsu, and F.K. Hwang. Hamiltonian property of d-consecutive digraphs. *Math. Comput. Modeling*, 17:61–63, 1993.
279. D.Z. Du, Y.-D. Lyuu, and D.F. Hsu. Line digraph iterations and the spread concept—with application to graph theory, fault tolerance, and routing. In *Graph-theoretic concepts in computer science (Fischbachau, 1991)*, volume 570 of *Lect. Notes Comput. Sci.*, pages 169–179. Springer, 1992.
280. I.S. Duff, A.M. Erisman, and J.K. Reid. *Direct methods for sparse matrices*. Oxford University Press, 1997.
281. R.J. Duffin. Topology of series-parallel networks. *J. Math. Anal. Appl.*, 10:303–318, 1965.
282. J. Edmonds. Paths, trees, and flowers. *Can. J. Math.*, 17:449–467, 1965.
283. J. Edmonds. Optimum branchings. *J. Res. Natl. Bur. Stand. Sect. B*, 71B:233–240, 1967.
284. J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Int. Conf., Calgary, Alta., 1969)*, pages 69–87. Gordon and Breach, 1970.
285. J. Edmonds. Edge-disjoint branchings. In B. Rustin, editor, *Combinatorial Algorithms*, pages 91–96. Academic Press, 1973.
286. J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In *Studies in integer programming (Proc. Workshop, Bonn, 1975)*, pages 185–204. Ann. Discrete Math., Vol. 1. North-Holland, 1977.
287. J. Edmonds and E.L. Johnson. Matching, Euler tours and the Chinese postman. *Math. Program.*, 5:88–124, 1973.
288. J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. Assoc. Comput. Mach.*, 19:248–264, 1972.
289. C.S. Edwards. Some extremal properties of bipartite subgraphs. *Can. J. Math.*, 25:475–485, 1973.
290. A. Ehrenfeucht, H.N. Gabow, R.M. McConnell, and S.J. Sullivan. An $O(n^2)$ divide-and-conquer algorithm for the prime tree decomposition of two-structures and modular decomposition of graphs. *J. Algor.*, 16(2):283–294, 1994.
291. A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
292. A. El-Sahili. Paths with two blocks in k-chromatic digraphs. *Discrete Math.*, 287:151–153, 2004.
293. A. El-Sahili. Trees in tournaments. *J. Combin. Theory Ser. B*, 92:183–187, 2004.

294. A. El-Sahili and M. Kouider. About paths with two blocks. *J. Graph Theory*, 55:221–226, 2007.
295. S. Enni. A note on mixed graphs and directed splitting off. *J. Graph Theory*, 27(4):213–221, 1998.
296. P. Erdős. Graph theory and probability. *Can. J. Math.*, 11:34–38, 1959.
297. P. Erdős. Some old and new problems in various branches of combinatorics. *Congr. Numer.*, 23:19–37, 1979.
298. P. Erdős and J.W. Moon. On sets of consistent arcs in a tournament. *Can. Math. Bull.*, 8:269–271, 1965.
299. P. Erdős, A.L. Rubin, and R.W. Irwing. Choosability in graphs. *Congr. Numer.*, 26:122–157, 1980.
300. P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compos. Math.*, 2:463–470, 1935.
301. P. Erdős and W.T. Trotter. When the Cartesian product of directed cycles is Hamiltonian. *J. Graph Theory*, 2:137–142, 1978.
302. A.H. Esfahanian and S.L. Hakimi. On computing the connectivities of graphs and digraphs. *Networks*, 14(2):355–366, 1984.
303. L. Euler. Solutio problematis ad geometriam situs pertinentis. *Comment. Acad. Sci. Imperialis Petropolitanae*, 8:128–140, 1736.
304. G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
305. S. Even. An algorithm for determining whether the connectivity of a graph is at least k. *SIAM J. Comput.*, 4(3):393–396, 1975.
306. S. Even. *Graph algorithms*. Computer Science Press, Woodland Hills, 1979.
307. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
308. S. Even and R.E. Tarjan. Network flow and testing graph connectivity. *SIAM J. Comput.*, 4(4):507–518, 1975.
309. T. Feder. Classification of Homomorphisms to Oriented Cycles and of k-Partite Satisfiability. *SIAM J. Discrete Math.*, 14(4):471–480, 2001.
310. U. Feige. A threshold of $\ln n$ for approximation set-cover. In *Proc. STOC 28: 28th Annual ACM Symp. on Theory of Computing*, pages 314–318. ACM Press, 1996.
311. J. Feldman and M. Ruhl. The directed Steiner network problem is tractable for a constant number of terminals. In *Proc. 40th Annual Symposium on Foundations of Computer Science*, pages 299–308. IEEE Computer Society Press, 1999.
312. W. Feller. *An introduction to probability theory and its applications. Vol. I.* John Wiley & Sons, New York, 3rd edition, 1968.
313. J. Feng, H.-E. Giesen, Y. Guo, G. Gutin, T. Jensen, and A. Rafiey. Characterization of edge-colored complete graphs with properly colored Hamilton paths. *J. Graph Theory*, 53(4):333–346, 2006.
314. P. Feofiloff and D. H. Younger. Directed cut transversal packing for source-sink connected graphs. *Combinatorica*, 7(3):255–263, 1987.
315. J.F. Fink and L. Lesniak-Foster. Graphs for which every unilateral orientation is traceable. *Ars Combin.*, 9:113–118, 1980.
316. M.A. Fiol, J.L.A. Yebra, and I. Alegre. Line digraph iteration and the (d, k) digraph problem. *IEEE Trans. Comput.*, C-33:400–403, 1984.
317. D.C. Fisher. Squaring a tournament: a proof of Dean's conjecture. *J. Graph Theory*, 23(1):43–48, 1996.
318. M.J. Fisher and A.R. Meyer. Boolean matrix multiplication and transitive closure. In *Proc. 12th Ann. ACM Symp. on Switching and Automata Theory*, pages 129–131. ACM Press, 1971.

319. H. Fleischner. Eine gemeinsame Basis für die Theorie der Eulerschen Graphen und den Satz von Petersen. *Monatsh. Math.*, 81(4):267–278, 1976.
320. H. Fleischner. *Eulerian graphs and related topics. Part 1. Vol. 1.* North-Holland, Amsterdam, 1990.
321. H. Fleischner. *Eulerian graphs and related topics. Part 1. Vol. 2.* North-Holland, Amsterdam, 1991.
322. H. Fleischner, G. Sabidussi, and E. Wegner. Transforming eulerian trails. *Discrete Math.*, 109:103–116, 1992.
323. H. Fleischner and S. Szeider. On edge-colored graphs covered by properly colored cycles. *Graphs Combin.*, 21:301–306, 2005.
324. R.W. Floyd. Algorithm 97, shortest path. *Commun. ACM*, 5:345, 1962.
325. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, 2006.
326. F.V. Fomin, M. Matamala, E. Prisner, and I. Rapaport. AT-free graphs: linear bounds for the oriented diameter. *Discrete Appl. Math.*, 141:135–148, 2004.
327. F.V. Fomin, M. Matamala, and I. Rapaport. Complexity of approximating the oriented diameter of chordal graphs. *J. Graph Theory*, 45:255–269, 2004.
328. F.V. Fomin and D.M. Thilikos. Dominating sets in planar graphs: branch-width and exponential speed-up, In *SODA'03: Proc. 14th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 168–177, Philadelphia, 2003, SIAM.
329. R. Forcade. Parity of paths and circuits in tournaments. *Discrete Math.*, 6:115–118, 1973.
330. L.R. Ford, Jr.. Network flow theory, Technical Report P-923, The Rand Corp., 1956.
331. L.R. Ford, Jr. and D.R. Fulkerson. *Flows in networks.* Princeton University Press, Princeton, NJ, 1962.
332. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
333. P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Appl. Math.*, 53:79–133, 1994.
334. P. Fraisse and C. Thomassen. Hamiltonian dicycles avoiding prescribed arcs in tournaments. *Graphs Combin.*, 3(3):239–250, 1987.
335. A. Frank. On the orientation of graphs. *J. Combin. Theory Ser. B*, 28(3):251–261, 1980.
336. A. Frank. On disjoint trees and arborescences. In *Algebraic methods in graph theory, Vol. I, II (Szeged, 1978)*, pages 159–169. North-Holland, 1981.
337. A. Frank. A note on k-strongly connected orientations of an undirected graph. *Discrete Math.*, 39(1):103–104, 1982.
338. A. Frank. An algorithm for submodular functions on graphs. In *Bonn Workshop on Combinatorial Optimization (Bonn, 1980)*, volume 16 of *Ann. Discrete Math.*, pages 97–120. North-Holland, 1982.
339. A. Frank. Finding feasible vectors of Edmonds-Giles polyhedra. *J. Combin. Theory Ser. B*, 36(3):221–239, 1984.
340. A. Frank. Submodular flows. In *Progress in combinatorial optimization (Waterloo, Ont., 1982)*, pages 147–165. Academic Press, 1984.
341. A. Frank. On connectivity properties of Eulerian digraphs. In *Graph theory in memory of G. A. Dirac (Sandbjerg, 1985)*, volume 41 of *Ann. Discrete Math.*, pages 179–194. North-Holland, 1989.
342. A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM J. Discrete Math.*, 5(1):25–53, 1992.
343. A. Frank. Applications of submodular functions. In *Surveys in combinatorics, 1993 (Keele)*, volume 187 of *London Math. Soc. Lect. Note Ser.*, pages 85–136. Cambridge University Press, 1993.

344. A. Frank. Submodular functions in graph theory. *Discrete Math.*, 111(1-3):231–243, 1993. Graph theory and combinatorics (Marseille-Luminy, 1990).
345. A. Frank. Connectivity augmentation problems in network design. In J.R. Birge and K.G. Murty, editors, *Mathematical Programming: State of the art*, pages 34–63. The University of Michigan, 1994.
346. A. Frank. Connectivity and network flows. In *Handbook of combinatorics, Vol. 1, 2*, pages 111–177. Elsevier, Amsterdam, 1995.
347. A. Frank. Orientations of graphs and submodular flows. *Congr. Numer.*, 113:111–142, 1996.
348. A. Frank. Applications of relaxed submodularity. *Doc. Math.*, Extra Vol. III:343–354, 1998.
349. A. Frank. Increasing the rooted-connectivity of a digraph by one. *Math. Program. Ser. B*, 84:565–576, 1999.
350. A. Frank and A. Gyárfás. Directed graphs and computer programs. In *Problémes Combinatoires et Théorie des Graphes*, Colloque Internationaux C.N.R.S., 260, pages 157–158, 1976.
351. A. Frank and A. Gyárfás. How to orient the edges of a graph? In *Combinatorics (Proc. 5th Hung. Colloq., Keszthely, 1976), Vol. I*, volume 18 of *Colloq. Math. Soc. János Bolyai*, pages 353–364. North-Holland, 1978.
352. A. Frank, T. Ibaraki, and H. Nagamochi. On sparse subgraphs preserving connectivity properties. *J. Graph Theory*, 17(3):275–281, 1993.
353. A. Frank, T. Ibaraki, and H. Nagamochi. Two arc-disjoint paths in Eulerian digraphs. *SIAM J. Discrete Math.*, 11(4):557–589 (electronic), 1998.
354. A. Frank and T. Jordán. Minimal edge-coverings of pairs of sets. *J. Combin. Theory Ser. B*, 65(1):73–110, 1995.
355. A. Frank and T. Jordán. Directed vertex-connectivity augmentation. *Math. Program. Ser. B*, 84:537–553, 1999.
356. A. Frank, T. Király, and Z. Király. On the orientation of graphs and hypergraphs. *Discrete Appl. Math.*, 131(2):385–400, 2003. Submodularity.
357. A. Frank and É. Tardos. Generalized polymatroids and submodular flows. *Math. Program. Ser. B*, 42(3):489–563, 1988.
358. A. Frank and É. Tardos. An application of submodular flows. *Linear Algebra Appl.*, 114/115:329–348, 1989.
359. M.L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *J. Assoc. Comput. Mach.*, 31:538–544, 1984.
360. M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. Assoc. Comput. Mach.*, 34(3):596–615, 1987.
361. S. Friedland. Every 7-regular digraph contains an even cycle. *J. Combin. Theory Ser. B*, 46(2):249–252, 1989.
362. A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.
363. G. Frobenius. Über zerlegbare Determinanten. *Sitzungsber. König. Preuss. Akad. Wiss.*, XVIII:274–277, 1917.
364. S. Fujishige. *Submodular functions and optimization, 2nd Edition*, volume 58 of *Ann. Discrete Math.* North-Holland, 2005.
365. S. Fujishige, H. Röck, and U. Zimmermann. A strongly polynomial algorithm for minimum cost submodular flow problems. *Math. Oper. Res.*, 14(1):60–69, 1989.
366. D.R. Fulkerson. Packing rooted directed cuts in a weighted directed graph. *Math. Program.*, 6:1–13, 1974.

367. M. Funke and G. Reinelt. A polyhedral approach to the feedback vertex set problem. In *Integer programming and combinatorial optimization (Vancouver, BC, 1996)*, pages 445–459. Springer, 1996.
368. Z. Fűredi, P. Horak, C.M. Pareek, and X. Zhu. Minimal oriented graphs of diameter 2. *Graphs Combin.*, 14:345–350, 1998.
369. M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *J. Algor.*, 17:409–423, 1994.
370. M.E. Furman. Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph. *Sov. Math. Dokl.*, 11:1252, 1970.
371. H. N. Gabow and T. Jordán. Incrementing bipartite digraph edge-connectivity. *J. Combin. Optim.*, 4(4):449–486, 2000.
372. H. N. Gabow and T. Jordán. Bipartition constrained edge-splitting in directed graphs. *Discrete Appl. Math.*, 115(1-3):49–62, 2001. 1st Japanese-Hungarian Symposium for Discrete Mathematics and its Applications (Kyoto, 1999).
373. H.N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. *Proc. SODA'90*, pages 434–443, 1990.
374. H.N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *J. Comput. System Sci.*, 50(2):259–273, 1995. 23rd Symposium on the Theory of Computing (New Orleans, LA, 1991).
375. H.N. Gabow. Better performance bounds for finding the smallest k-edge connected spanning subgraph of a multigraph, In *SODA*, pages 460–469, 2003.
376. H.N. Gabow. Special edges, and approximating the smallest directed k-edge connected spanning subgraph, In *SODA '04: Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 234–243, Philadelphia, 2004, SIAM.
377. H.N. Gabow. Using expander graphs to find vertex connectivity. *J. Assoc. Comput. Mach.*, 53(5):800–844, 2006.
378. D. Gale. A theorem on flows in networks. *Pac. J. Math.*, 7:1073–1082, 1957.
379. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *Amer. Math. Mon.*, 69:9–15, 1962.
380. H. Galeana-Sánchez. A counterexample to a conjecture of Meyniel on kernel-perfect graphs. *Discrete Math.*, 41:105–107, 1982.
381. H. Galeana-Sánchez. Kernels and perfectness in arc-local tournament digraphs. *Discrete Math.*, 306(19-20):2473–2480, 2006.
382. H. Galeana-Sánchez and X. Li. Semikernels and (k, l)-kernels in digraphs. *SIAM J. Discrete Math.*, 11(2):340–346, 1998.
383. H. Galeana-Sánchez and V. Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Math.*, 48:67–76, 1984.
384. Z. Galil. Finding the vertex connectivity of graphs. *SIAM J. Comput.*, 9(1):197–199, 1980.
385. T. Gallai. Maximum-minimum Sätze und verallgemeinerte faktoren in graphen. *Acta Math. Acad. Sci. Hung.*, 12:131–173, 1961.
386. T. Gallai. Problem 15. In M. Fiedler, editor, *Theory of Graphs and its Applications*, page 161. Czech. Acad. Sci. Prague, 1964.
387. T. Gallai. On directed paths and circuits. In *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, pages 115–118. Academic Press, 1968.
388. T. Gallai and A.N. Milgram. Verallgemeinerung eines graphentheoretischen Satzes von Rédei. *Acta Sci. Math. Szeged*, 21:181–186, 1960.
389. A. Galluccio and M. Loebl. Cycles of prescribed modularity in planar digraphs. *J. Algor.*, 21(1):51–70, 1996.
390. A. Galluccio and M. Loebl. Even directed graphs in H-free digraphs. *J. Algor.*, 27, 1996.

391. A. Galluccio and M. Loebl. (p, q)-odd digraphs. *J. Graph Theory*, 23(2):175–184, 1996.
392. F. Galvin. The list chromatic index of a bipartite multigraph. *J. Combin. Theory Ser. B*, 63:153–158, 1995.
393. M.R. Garey and D.S. Johnson. *Computers and intractability*. W. H. Freeman, San Francisco, 1979.
394. M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976.
395. M.R. Garey, D.S. Johnson, and R.E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5:704–714, 1976.
396. F. Gavril. Some NP-complete problems on graphs, In *Proc. 11th Conf. on Information Sciences and Systems*, pages 91–95, 1977.
397. A.M.H. Gerards. Homomorphisms of graphs into odd cycles. *J. Graph Theory*, 12(1):73–83, 1988.
398. A.M.H. Gerards. *Graphs and polyhedra. Binary spaces and cutting planes*. Stichting Mathematisch Centrum Centrum voor Wiskunde en Informatica, Amsterdam, 1990.
399. A.M.H. Gerards. An orientation theorem for graphs. *J. Combin. Theory Ser. B*, 62(2):199–212, 1994.
400. A.M.H. Gerards and F.B. Shepherd. Strong orientations without even directed circuits. *Discrete Math.*, 188(1-3):111–125, 1998.
401. D. Ghosh, B. Goldengorin, G. Gutin, and G. Jäger. Tolerance-based greedy algorithms for the traveling salesman problem. *Commun. DQM*, 10:52–70, 2007.
402. A. Ghouila-Houri. Diametre maximal d'un graphe fortement connexe. *C.R. Acad. Sci. Paris*, 250:254–256, 1960.
403. A. Ghouila-Houri. Une condition suffisante d'existence d'un circuit hamiltonien. *C.R. Acad. Sci. Paris*, 25:495–497, 1960.
404. A. Ghouila-Houri. Caractérisation des graphes non orientés dont on peut orienter les arĕtes de manière à obtenir le graphe d'une relation d'ordre. *C. R. Acad. Sci. Paris*, 254:1370–1371, 1962.
405. A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, Cambridge, 1985.
406. P. Gibbons, R. Karp, V. Ramachandran, D. Soroker, and R. Tarjan. Transitive compaction in parallel via branchings. *J. Algor.*, 12(1):110–125, 1991.
407. F. Glover, G. Gutin, A. Yeo, and A. Zverovich. Construction heuristics and domination analysis for the asymmetric TSP. *Eur. J. Oper. Res.*, 129:555–568, 2001.
408. F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, 1997.
409. F. Glover and A.P. Punnen. The traveling salesman problem: new solvable cases and linkages with the development of approximation algorithms. *J. Oper. Res. Soc.*, 48:502–510, 1997.
410. F.W. Glover and G.A. Kochenberger, editors. *Handbook of Metaheuristics*. Springer, 2003.
411. A. Godbole, Z. Cohn, and E. Wright. Probabilistic Versions of Seymour's Distance Two Conjecture. Manuscript, 2008.
412. W.D. Goddard, G. Kubicki, O.R. Oellermann, and S.L. Tian. On multipartite tournaments. *J. Combin. Theory Ser. B*, 52(2):284–300, 1991.
413. W.D. Goddard and O.R. Oellermann. On the cycle structure of multipartite tournaments. In *Graph Theory Combin. Appl. 1*, pages 525–533. Wiley, New York, 1991.
414. M.X. Goemans and D. Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 18:37–59, 1998.

415. A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. In *Proc. 18th ACM Symposium on the Theory of Computing*, pages 136–146. ACM Press, 1986.
416. A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.*, 35(4):921–940, 1988.
417. A.V. Goldberg and R.E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *J. Assoc. Comput. Mach.*, 36(4):873–886, 1989.
418. M.K. Goldberg. The diameter of a strongly connected graph. *Dokl. Akad. Nauk SSSR*, 170:767–769, 1966.
419. B. Goldengorin, D. Ghosh, and G. Sierksma. Branch and peg algorithms for the simple plant location problem. *Comput. Oper. Res.*, 31:241–255, 2004.
420. M.C. Golumbic. The complexity of comparability graph recognition and coloring. *Computing*, 18(3):199–208, 1977.
421. M.C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980. With a foreword by Claude Berge.
422. M. Gondran and M. Minoux. *Graphs and algorithms*. John Wiley & Sons, Chichester, 1984.
423. A. Goralcikova and V. Koubek. A reduct-and-closure algorithm for graphs. In *Proc. 8th Symp. on Math. Foundations of Computer Science*, volume 74 of *Lect. Notes Comput. Sci.*, pages 301–307. Springer-Verlag, 1979.
424. È.Y. Grinberg. Examples of non-Ádám multigraphs. *Latv. Mat. Ezhegodnik*, 31:128–138, 253, 1988.
425. M. Grohe and M. Grüber. Parameterized approximability of the disjoint cycle problem. In *Proc. ICALP'07: 34th Int. Colloquium on Automata, Languages and Programming*, volume 4596 of *Lect. Notes Comput. Sci.*, pages 363–374. Springer-Verlag, 2007.
426. J.W. Grossman and R. Häggkvist. Alternating cycles in edge-partitioned graphs. *J. Combin. Theory Ser. B*, 34:77–81, 1983.
427. M. Grötschel and F. Harary. The graphs for which all strong orientations are hamiltonian. *J. Graph Theory*, 3:221–224, 1979.
428. M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Oper. Res.*, 32(6):1195–1220, 1984.
429. M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
430. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, 1988.
431. B. Grünbaum. Antidirected Hamiltonian paths in tournaments. *J. Combin. Theory Ser. B*, 11:249–257, 1971.
432. Y. Guo. *Locally semicomplete digraphs*, PhD thesis, RWTH Aachen, Germany, 1995.
433. Y. Guo. Strongly Hamiltonian-connected locally semicomplete digraphs. *J. Graph Theory*, 22(1):65–73, 1996.
434. Y Guo. Path-connectivity in local tournaments. *Discrete Math.*, 167/168:353–372, 1997. 15th British Combinatorial Conference (Stirling, 1995).
435. Y. Guo. Spanning local tournaments in locally semicomplete digraphs. *Discrete Appl. Math.*, 79(1-3):119–125, 1997. 4th Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1995).
436. Y. Guo. Semicomplete Multipartite Digraphs: A Generalization of Tournaments. German Habilitation Thesis, RWTH Aachen, Germany 1998.
437. Y. Guo and J.H. Kwak. The cycle structure of regular multipartite tournaments. *Discrete Appl. Math.*, 120:109–116, 2002.

438. Y. Guo, A. Pinkernell, and L. Volkmann. On cycles through a given vertex in multipartite tournaments. *Discrete Math.*, 164(1-3):165–170, 1997. The Second Krakow Conference on Graph Theory (Zgorzelisko, 1994).
439. Y. Guo, M. Tewes, L. Volkmann, and A. Yeo. Sufficient conditions for semicomplete multipartite digraphs to be hamiltonian. *Discrete Math.*, 212(1-2):91–100, 2002.
440. Y. Guo and L. Volkmann. Pancyclic locally semicomplete digraphs. Unpublished manuscript 1992.
441. Y. Guo and L. Volkmann. Cycles in multipartite tournaments. *J. Combin. Theory Ser. B*, 62(2):363–366, 1994.
442. Y. Guo and L. Volkmann. On complementary cycles in locally semicomplete digraphs. *Discrete Math.*, 135(1-3):121–127, 1994.
443. Y. Guo and L. Volkmann. Locally semicomplete digraphs that are complementary m-pancyclic. *J. Graph Theory*, 21(2):121–136, 1996.
444. G. Gutin. Criterion for complete bipartite digraphs to be Hamiltonian. *Vestsī Akad. Navuk BSSR Ser. Fīz.-Mat. Navuk*, no. 1:109–110, 1984.
445. G. Gutin. Effective characterization of complete bipartite digraphs that have a Hamiltonian path. *Kibernetika (Kiev)*, no. 4:124–125, 1985.
446. G. Gutin. The radii of n-partite tournaments. *Mat. Zametki*, 40(3):414–417, 430, 1986.
447. G. Gutin. Finding the largest contour in a complete bipartite digraph. *Kibernetika*, no. 2:117–118, 1987.
448. G. Gutin. Characterization of complete n-partite digraphs that have a Hamiltonian path. *Kibernetika (Kiev)*, no. 1:107–108, 136, 1988.
449. G. Gutin. Characterization of vertex pancyclic partly oriented k-partite tournaments. *Vestsī Acad. Navuk BSSR Ser.Fiz.-Mat.*, no. 2:41–46, 1989.
450. G. Gutin. m-sources in complete multipartite digraphs. *Vestsī Akad. Navuk BSSR Ser. Fīz.-Mat. Navuk*, no. 5:101–106, 128, 1989.
451. G. Gutin. *Cycles and paths in directed graphs*, PhD thesis, School of Mathematics, Tel Aviv University, 1993.
452. G. Gutin. Finding a longest path in a complete multipartite digraph. *SIAM J. Discrete Math.*, 6:270–273, 1993.
453. G. Gutin. On cycles in multipartite tournaments. *J. Combin. Theory Ser. B*, 58(2):319–321, 1993.
454. G. Gutin. Polynomial algorithms for finding Hamiltonian paths and cycles in quasi-transitive digraphs. *Australas. J. Combin.*, 10:231–236, 1994.
455. G. Gutin. Minimizing and maximizing the diameter in orientations of graphs. *Graphs Combin.*, 10(3):225–230, 1994.
456. G. Gutin. Characterizations of vertex pancyclic and pancyclic ordinary complete multipartite digraphs. *Discrete Math.*, 141(1-3):153–162, 1995.
457. G. Gutin. Cycles and paths in semicomplete multipartite digraphs, theorems, and algorithms: a survey. *J. Graph Theory*, 19(4):481–505, 1995.
458. G. Gutin. Exponential neighbourhood local search for the traveling salesman problem. *Comput. Oper. Res.*, 26:313–320, 1999.
459. G. Gutin. Connected (g, f)-factors and supereulerian digraphs. *Ars Combin.*, 54:311–317, 2000.
460. G. Gutin. Note on edge-colored graphs and digraphs without properly colored cycles. *Australas. J. Combin.*, 42:137–140, 2008.
461. G. Gutin, B. Goldengorin, and J. Huang. Worst Case Analysis of Max-Regret, Greedy and Other Heuristics for Multidimensional Assignment and Traveling Salesman Problems. *J. Heuristics*, pages 169–181, 14.
462. G. Gutin, T. Jensen, and A. Yeo. Domination analysis for minimum multiprocessor scheduling. *Discrete Appl. Math.*, 154:2613–2619, 2006.

463. G. Gutin, A. Johnstone, J. Reddington, E. Scott, A. Soleimanfallah, and A. Yeo. An algorithm for finding connected convex subgraphs of an acyclic digraph, In *Algorithms and Complexity in Durham 2007*, College Publications, London, 2007.
464. G. Gutin, N. Jones, A. Rafiey, S. Severini, and A. Yeo. Mediated digraphs and quantum nonlocality. *Discrete Appl. Math.*, 150(1-3):41–50, 2005.
465. G. Gutin and E.J. Kim. Properly Coloured Cycles and Paths: Results and Open Problems. Preprint arXiv:0805.3901v3, May 2008.
466. G. Gutin, T. Kloks, C.-M. Lee, and A. Yeo. Kernels in planar digraphs. *J. Comput. Syst. Sci.*, 71(2):174–184, 2005.
467. G. Gutin, K. M. Koh, E. G. Tay, and A. Yeo. Almost minimum diameter orientations of semicomplete multipartite and extended digraphs. *Graphs Combin.*, 18(3):499–506, 2002.
468. G. Gutin, K. M. Koh, E. G. Tay, and A. Yeo. On the number of quasi-kernels in digraphs. *J. Graph Theory*, 46(1):48–56, 2004.
469. G. Gutin, A. Koller, and A. Yeo. Note on Upper Bounds for TSP Domination Number. *Algor. Oper. Res.*, 1(1):52–54, 2006.
470. G. Gutin and A. Rafiey. When n-cycles in n-partite tournaments are longest cycles. *Discrete Math.*, 289(1-3):163–168, 2004.
471. G. Gutin, A. Rafiey, and A. Yeo. On n-partite tournaments with unique n-cycle. *Graphs Combin.*, 22(2):241–249, 2006.
472. G. Gutin, I. Razgon, and E.J. Kim. Minimum leaf out-branching problems. In *Proc. AAIM'08*, volume 5034 of *Lect. Notes Comput. Sci.*, pages 235–246. Springer, 2008.
473. G. Gutin, M. Tewes, and A. Yeo. Longest paths in strong spanning oriented subgraphs of strong semicomplete multipartite digraphs. *Discrete Math.*, 222:269–274, 2000.
474. G. Gutin and A. Yeo. Kings in semicomplete multipartite digraphs. *J. Graph Theory*, 33:177–183, 2000.
475. G. Gutin and A. Yeo. Note on the path covering number of a semicomplete multipartite digraph. *J. Combin. Math. and Combin. Computing.*, 32:231–237, 2000.
476. G. Gutin and A. Yeo. Quasi-hamiltonicity: a series of necessary conditions for a digraph to be hamiltonian. *J. Combin. Theory Ser. B*, 78:232–242, 2000.
477. G. Gutin and A. Yeo. Remarks on hamiltonian digraphs. *Australas. J. Combin.*, 23:115–118, 2001.
478. G. Gutin and A. Yeo. Orientations of digraphs almost preserving diameter. *Discrete Appl. Math.*, 121(1-3):129–138, 2002.
479. G. Gutin and A. Yeo. Polynomial approximation algorithms for the TSP and the QAP with factorial domination number. *Discrete Appl. Math.*, 119:107–116, 2002.
480. G. Gutin and A. Yeo. Domination Analysis of Combinatorial Optimization Algorithms and Problems. In M.C. Golumbic and I.B.-A. Hartman, editors, *Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications*. Springer-Verlag, 2005.
481. G. Gutin and A. Yeo. On the number of connected convex subgraphs of a connected acyclic digraph. *Discrete Appl. Math.*, to appear:5 pp., 2008.
482. G. Gutin and A. Yeo. Some Parameterized Problems on Digraphs. *Comput. J.*, 51:363–371, 2008.
483. G. Gutin, A. Yeo, and A. Zverovich. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Appl. Math.*, 117:81–86, 2002.

484. G. Gutin, A. Yeo, and A. Zverovitch. Exponential neighborhoods and domination analysis for the tsp. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and its Variations*. Kluwer, Dordrecht, 2002.
485. M. Habib, M. Morvan, and J.-X. Rampon. On the calculation of transitive reduction-closure of orders. *Discrete Math.*, 111:289–303, 1993.
486. R. Häggkvist. On F-Hamiltonian graphs. In *Graph Theory and Related Topics*, pages 219–231. Academic Press, New York, 1979.
487. R. Häggkvist. Hamilton cycles in oriented graphs. *Combin. Probab. Comput.*, 2(1):25–32, 1993.
488. R. Häggkvist and Y. Manoussakis. Cycles and paths in bipartite tournaments with spanning configurations. *Combinatorica*, 9(1):33–38, 1989.
489. R. Häggkvist and A. Thomason. Trees in tournaments. *Combinatorica*, 11(2):123–130, 1991.
490. R. Häggkvist and C. Thomassen. On pancyclic digraphs. *J. Combin. Theory Ser. B*, 20(1):20–40, 1976.
491. A. Hajnal, E.C. Milner, and E. Szemerédi. A cure for the telephone disease. *Can. Math. Bull.*, 15:447–450, 1972.
492. S.L. Hakimi, E.F. Schmeichel, and N.E. Young. Orienting graphs to optimize reachability. *Inf. Process. Lett.*, 63(5):229–235, 1997.
493. P. Hall. On representation of subsets. *J. London Math. Soc.*, 10:26–30, 1935.
494. Y.O. Hamidoune. *Contribution a l' étude de la connectivité d'un graphe*, PhD thesis, Université Paris VI, 1980.
495. Y.O. Hamidoune. An application of connectivity theory in graphs to factorizations of elements in finite groups. *Eur. J. Combin.*, 2:349–355, 1981.
496. Y.O. Hamidoune. A note on minimal directed graphs with given girth. *J. Combin. Theory Ser. B*, 43:343–348, 1987.
497. S. Hansen. Heuristic and exact methods for solving the feedback arc set problem. Master's thesis, Department of Mathematics and Computer Science, University of Southern Denmark, Odense, 1999.
498. F. Harary. The maximum connectivity of a graph. *Proc. Natl. Acad. Sci. U.S.A.*, 48:1142–1146, 1962.
499. F. Harary, J.A. Kabell, and F.R. McMorris. Bipartite intersection graphs. *Comment. Math. Univ. Carolin.*, 23:739–745, 1982.
500. F. Harary, J. Krarup, and A. Schwenk. Graphs suppressible to an edge. *Can. Math. Bull.*, 15:201–204, 1971.
501. F. Harary and L. Moser. The theory of round robin tournaments. *Amer. Math. Mon.*, 73:231–246, 1966.
502. F. Harary and R.Z. Norman. Some properties of line digraphs. *Rend. Circ. Mat. Palermo*, 9(2):161–168, 1960.
503. F. Harary, R.Z. Norman, and D. Cartwright. *Structural Models*. John Wiley & Sons, 1965.
504. R. Hassin and S. Khuller. z-Approximations. *J. Algor.*, 41:429–442, 2001.
505. F. Havet. Finding an oriented hamiltonian path in a tournament. *J. Algor.*, 36:253–275, 2000.
506. F. Havet. Oriented Hamiltonian cycles in tournaments. *J. Combin. Theory*, 80:1–31, 2000.
507. F. Havet. Stable set meeting every longest path. *Discrete Math.*, 289:169–173, 2004.
508. F. Havet and S. Thomassé. Median orders of tournaments: a tool for the second neighborhood problem and Sumner's conjecture. *J. Graph Theory*, 35(4):244–256, 2000.
509. F. Havet and S. Thomassé. Oriented hamiltonian paths in tournaments: a proof of Rosenfeld's conjecture. *J. Combin. Theory Ser. B*, 78:243–273, 2000.

510. F. Havet, S. Thomassé, and A. Yeo. Hoàng-Reed Conjecture holds for tournaments. *Discrete Math.*, 308:3412–3415, 2008.
511. S.M. Hedetniemi, S.T. Hedetniemi, and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:129–134, 1988.
512. P. Hell, J. Bang-Jensen, and J. Huang. Local tournaments and proper circular arc graphs. In *Algorithms (Tokyo, 1990)*, volume 450 of *Lect. Notes Comput. Sci.*, pages 101–108. Springer, 1990.
513. P. Hell and J. Huang. Lexicographic orientation and representation algorithms for comparability graphs, proper circular arc graphs, and proper interval graphs. *J. Graph Theory*, 20(3):361–374, 1995.
514. P. Hell and M. Rosenfeld. The complexity of finding generalized paths in tournaments. *J. Algor.*, 4(4):303–309, 1983.
515. P. Hell and M. Rosenfeld. Antidirected hamiltonian paths between specified vertices of a tournament. *Discrete Appl. Math.*, 117:87–98, 2002.
516. R.L. Hemminger and L.W. Beineke. Line graphs and line digraphs. In L.W. Beineke and R.J. Wilson, editors, *Selected Topics in Graph Theory*, pages 271–305. Academic Press, London, 1978.
517. G.R.T. Hendry. Extending cycles in directed graphs. *J. Combin. Theory Ser. B*, 46(2):162–172, 1989.
518. G.R.T. Hendry. Extending cycles in graphs. *Discrete Math.*, 85(1):59–72, 1990.
519. J.S. Heslop-Harrison and M.D. Bennett. Prediction and analysis of spacial order in haploid chromosome complements. *Proc. R. Soc. London*, B:211–223, 1983.
520. J.S. Heslop-Harrison and M.D. Bennett. The spacial order of chromosomes in root-tip metaphases of Aegilops umbellulata. *Proc. R. Soc. London*, B:225–239, 1983.
521. G. Hetyei. Cyclic connectivity classes of directed graphs. *Acta Math. Acad. Paedagog. Nyházi.*, 17:47–59, 2001.
522. C. Heuchenne. Sur une certaine correspondance entre graphs. *Bull. Soc. R. Sci. Liége*, 33:743–753, 1964.
523. J. van den Heuvel and M. Johnson. Transversals of subtree hypergraphs and the source location problem in digraphs, CDAM Research Report LSE-CDAM-2004-10, London School of Economics, 2004.
524. J. van den Heuvel and M. Johnson. The external network problem with edge- or arc-connectivity requirements, In *Proc. CAAN 2004*, volume 3405 of *Lect. Notes Comput. Sci.*, pages 114–126, 2005.
525. M.-C. Heydemann and D. Sotteau. About some cyclic properties in digraphs. *J. Combin. Theory Ser. B*, 38(3):261–278, 1985.
526. M.-C. Heydemann, D. Sotteau, and C. Thomassen. Orientations of Hamiltonian cycles in digraphs. *Ars Combin.*, 14:3–8, 1982.
527. M.C. Heydemann. On cycles and paths in digraphs. *Discrete Math.*, 31:217–219, 1980.
528. F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research.* McGraw Hill, 2001.
529. A.J.W. Hilton. Alternating Hamiltonian circuits in edge-coloured bipartite graphs. *Discrete Appl. Math.*, 35:271–273, 1992.
530. C.T. Hoang and B. Reed. A note on short cycles in digraphs. *Discrete Math.*, 66(1-2):103–107, 1987.
531. A.J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. In R. Bellman and M. Hall, editors, *Combinatorial Analysis*, pages 113–128. American Mathematical Society, Providence, RI, 1960.

532. H.H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco, 2004.
533. J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, 1979.
534. J.E. Hopcroft and R.E. Tarjan. Efficient planarity testing. *J. Assoc. Comput. Mach.*, 21:549–568, 1974.
535. F.W. Horsley. *Means Scheduling Manual: On-Time, On-Budget Construction Up-To-Date Computerized Scheduling*. Roberts Means Co., 1991.
536. J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting & gossiping). In *Combinatorial network theory*, pages 125–212. Kluwer, Dordrecht, 1996.
537. H.T. Hsu. An algorithm for finding a minimal equivalent graph of a digraph. *J. Assoc. Comput. Mach.*, 22:11–16, 1975.
538. J. Huang. *Tournament-like oriented graphs*, PhD thesis, School of Computing Science, Simon Fraser University, Canada, 1992.
539. J. Huang. On the structure of local tournaments. *J. Combin. Theory Ser. B*, 63(2):200–221, 1995.
540. J. Huang. A note on spanning local tournaments in locally semicomplete digraphs. *Discrete Appl. Math.*, 89:277–279, 1998.
541. J. Huang. Which digraphs are round? *Australas. J. Combin.*, 19:203–208, 1999.
542. J. Huang and D. Ye. Sharp Bounds for the Oriented Diameters of Interval Graphs and 2-Connected Proper Interval Graphs. In *Computational Science - ICCS 2007, 7th International Conference, Beijing, China, Proceedings, Part III*, volume 4489 of *Lect. Notes Comput. Sci.*, pages 353–361. Springer, 2007.
543. A. Hubenko. On a cyclic connectivity property of directed graphs. *Discrete Math.*, 308:1018–1024, 2008.
544. P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings, In *SODA'07: Proc. 18th annual ACM-SIAM symposium on Discrete algorithm*, pages 637–644, New York, NY, USA, 2007, ACM.
545. F.K. Hwang. The Hamiltonian property of linear functions. *Oper. Res. Lett.*, 6:125–127, 1987.
546. T. Ibaraki and S. Poljak. Weak three-linking in Eulerian digraphs. *SIAM J. Discrete Math.*, 4(1):84–98, 1991.
547. M. Imase and M. Itoh. Design to minimize a diameter on building block networks. *IEEE Trans. Comput.*, C-30:439–443, 1981.
548. M. Imase and M. Itoh. Design for directed graphs with minimum diameter. *IEEE Trans. Comput.*, C-32:782–784, 1983.
549. M. Imase, I. Soneoka, and K. Okada. Connectivity of regular directed graphs with small diameter. *IEEE Trans. Comput.*, C-34:267–273, 1985.
550. M. Imase, I. Soneoka, and K. Okada. A fault tolerant processor interconnection network. *Syst. Comput. Japan*, 17:21–30, 1986.
551. M. Imori, M. Matsumoto, and H. Yamada. The line digraph of a regular and pancircular digraph is also regular and pan circular. *Graphs Combin.*, 4:235–239, 1988.
552. G. Isaak. Tournaments as feedback arc sets. *Electron. J. Combin.*, 2:19pp, 1995.
553. H. Ito, K. Makino, K. Arata, S. Honami, Y. Itatsu, and S. Fujishige. Source location problem with flow requirements in directed networks. *Optimization Methods and Software*, 18:427–435, 2003.
554. S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. Assoc. Comput. Mach.*, 48:761–777, 2001.

555. B. Jackson. Long paths and cycles in oriented graphs. *J. Graph Theory*, 5(2):145–157, 1981.
556. B. Jackson. Some remarks on arc-connectivity, vertex splitting, and orientation in graphs and digraphs. *J. Graph Theory*, 12(3):429–436, 1988.
557. B. Jackson and O. Ordaz. A Chvátal-Erdős condition for $(1, 1)$-factors in digraphs. *Discrete Math.*, 57(1-2):199–201, 1985.
558. B. Jackson and O. Ordaz. Chvátal-Erdős conditions for paths and cycles in graphs and digraphs. A survey. *Discrete Math.*, 84(3):241–254, 1990.
559. H. Jacob and H. Meyniel. About quasi-kernels in a digraph. *Discrete Math.*, 154(1-3):279–280, 1996.
560. F. Jaeger. On nowhere-zero flows in multigraphs. *Congr. Numer.*, 15:373–378, 1976.
561. F. Jaeger. On five-edge-colorings of cubic graphs and nowhere-zero flow problems. *Ars Combin.*, 20(B):229–244, 1985.
562. F. Jaeger. Nowhere-zero flow problems. In *Selected Topics in Graph Theory, 3*, pages 71–95. Academic Press, 1988.
563. J. Janssen. The Dinitz problem solved for rectangles. *Bull. Amer. Math. Soc.*, 29:243–249, 1993.
564. T.R. Jensen and B. Toft. *Graph coloring problems*. John Wiley & Sons, Inc., New York, 1995. A Wiley-Interscience Publication.
565. W.S. Jewell. Optimal flow through networks, Technical Report 8, OR Center, MIT, Cambridge, 1958.
566. J. Jirásek. On a certain class of multidigraphs, for which reversal of no arc decreases the number of their cycles. *Comment. Math. Univ. Carolin.*, 28:185–189, 1987.
567. J. Jirásek. Some remarks on Ádám's conjecture for simple directed graphs. *Discrete Math.*, 108:327–332, 1992.
568. J. Jirásek. Arc reversal in nonhamiltonian circulant oriented graphs. *J. Graph Theory*, 49(1):59–68, 2005.
569. D.B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. Assoc. Comput. Mach.*, 24:1–13, 1977.
570. D.S. Johnson, C.R. Aragon, L. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part 1, Graph partitioning. *Oper. Res.*, 37:865–892, 1989.
571. D.S. Johnson, G. Gutin, L.A. McGeoch, A. Yeo, X. Zhang, and A. Zverovitch. Experimental analysis of heuristics for the atsp. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 445–487. Kluwer, Dordrecht, 2002.
572. D.S. Johnson and L.A. McGeoch. Experimental Analysis of Heuristics for the STSP. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 369–443. Kluwer, Dordrecht, 2002.
573. T. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *J. Combin. Theory Ser. B*, 82(1):138–154, 2001.
574. N.S. Jones, N. Linden, and S. Massar. The Extent of Multi-particle Quantum Non-locality. *Phys. Rev. A*, 71:042329, 2005.
575. T. Jordán. Increasing the vertex-connectivity in directed graphs. In *Algorithms—ESA '93 (Bad Honnef, 1993)*, volume 726 of *Lect. Notes Comput. Sci.*, pages 236–247. Springer, 1993.
576. T. Jordán. *Connectivity augmentation problems in Graphs*, PhD thesis, Department of Computer Science, Eötvös University, Budapest, 1994.
577. T. Jordán. On the optimal vertex-connectivity augmentation. *J. Combin. Theory Ser. B*, 63:8–20, 1995.

578. T. Jordán. On the existence of k edge-disjoint 2-connected spanning subgraphs. *J. Combin. Theory Ser. B*, 95(2):257–262, 2005.
579. S. Jukna. *Extremal Combinatorics with Applications in Computer Science*. Springer, Berlin, 2001.
580. H.A. Jung. Eine Verallgemeinerung des n-fachen zusammenhangs für Graphen. *Math. Ann.*, 187:95–103, 1970.
581. Y. Kaneko and S.C. Locke. The minimum degree approach for Paul Seymour's distance 2 conjecture. *Congr. Numer.*, 148:201–206, 2001.
582. S. Kapoor and H. Ramesh. An Algorithm for Enumerating All Spanning Trees of a Directed Graph. *Algorithmica*, 27(2):120–130, 2000.
583. J. Kari. Synchronizing finite automata on Eulerian digraphs. *Theor. Comput. Sci.*, 295:223–232, 2003.
584. R.M. Karp. A simple derivation of Edmonds' algorithm for optimum branching. *Networks*, 1:265–272, 1971/72.
585. R.M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Symp., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, 1972.
586. A.V. Karzanov. The problem of finding the maximal flow in a network by the method of preflows. *Dokl. Akad. Nauk SSSR*, 215:49–52, 1974.
587. P. Keevash, D. Kühn, and D. Osthus. An exact minimum degree condition for Hamilton cycles in oriented graphs. Submitted 2007.
588. L. Kelly, D. Kühn, and D. Osthus. A Dirac type result on Hamilton cycles in oriented graphs. http://arxiv.org/abs/0709.1047, 2007.
589. J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Springer-Verlag, New York, 1976.
590. A. Kemnitz and B. Greger. A forbidden subdigraph condition implying an oriented graph to be Hamiltonian. *Congr. Numer.*, 130:127–131, 1998.
591. C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proc. STOC'07: 39th Annual ACM Symposium on Theory of Computing*, pages 95–103. ACM Press, 2007.
592. S. Khuller, B. Raghavachari, and N. Young. Approximating the minimum equivalent digraph. *SIAM J. Comput.*, 24(4):859–872, 1995.
593. S. Khuller, B. Raghavachari, and N. Young. On strongly connected digraphs with bounded cycle length. *Discrete Appl. Math.*, 69(3):281–289, 1996.
594. S. Khuller, B. Raghavachari, and A. Zhu. A uniform framework for approximating weighted connectivity problems, In *SODA'99: Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, Maryland)*, pages 937–938, Philadelphia, 1999, SIAM.
595. Z. Király and Z. Szigeti. Simultaneous well-balanced orientations of graphs. *J. Combin. Theory Ser. B*, 96(5):684–692, 2006.
596. G. Kirchoff. Über die Ausflösung der Gleichungen auf welsche man bei der Untersuchungen der Linearen Verteilung Galvanisher Ströme geführt wird. *Poggendorf Ann. Phys.*, 72:497–508, 1847.
597. L.M. Kirousis and C.H. Papadimitriou. Interval graphs and searching. *Discrete Math.*, 55:181–184, 1985.
598. M. Klein. A primal method for minimum cost flows with applications to the assignment and transportation problems. *Manage. Sci.*, 14:205–220, 1967.
599. J. Kleinberg and É. Tardos. *Algorithm Design*. Pearson Education, Inc., Addison Wesley, Boston, 2006.
600. P. Kleinschmidt and H. Schannath. A strongly polynomial algorithm for the transportation problem. *Math. Program.*, 68:1–13, 1995.
601. T. Kloks. *Treewidth: computations and approximations*, volume 842 of *Lect. Notes Comput. Sci.* Springer-Verlag, 1994.

602. D.E. Knuth. *The art of computer programming. Vol. 1: Fundamental algorithms*. Addison-Wesley, Reading, 1968. Second printing.
603. A.V. Knyazev. Diameters of Pseudosymmetric Graphs. *Math. Notes*, 41(5-6):473–482, 1987.
604. W. Kocay and D. Stone. An algorithm for balanced flows. *J. Combin. Math. Combin. Comput.*, 19:3–31, 1995.
605. K.M. Koh. Even circuits in directed graphs and Lovasz's conjecture. *Bull. Malays. Math. Soc.*, 7(3):47–52, 1976.
606. K.M. Koh and B.P. Tan. The diameters of a graph and its orientations, Technical report, Department of Mathematics, National University of Singapore, 1992.
607. K.M. Koh and B.P. Tan. Kings in multipartite tournaments. *Discrete Math.*, 147:171–183, 1995.
608. K.M. Koh and B.P. Tan. Number of 4-kings in bipartite tournaments with no 3-kings. *Discrete Math.*, 154(1-3):281–287, 1996.
609. K.M. Koh and B.P. Tan. The number of kings in a multipartite tournament. *Discrete Math.*, 167/168:411–418, 1997.
610. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of even cycles and paths. *Networks*, 30(1):1–7, 1997.
611. K.M. Koh and E.G. Tay. Optimal orientations of products of paths and cycles. *Discrete Appl. Math.*, 78(1-3):163–174, 1997.
612. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of graphs (I). *Discrete Math.*, 190:115–136, 1998.
613. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products with a bipartite graph. *Discrete Appl. Math.*, 98:103–120, 1999.
614. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of graphs (II): complete graphs and even cycles. *Discrete Math.*, 211:75–102, 2000.
615. K.M. Koh and E.G. Tay. On optimal orientations of G vertex-multiplications. *Discrete Math.*, 219:153–171, 2000.
616. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of trees. *Graphs Combin.*, 17:79–97, 2001.
617. A.E. Koller and S.D. Noble. Domination Analysis of Greedy Heuristics for the Frequency Assignment Problem. *Discrete Math.*, 275(1-3):331–338, 2004.
618. D. König. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Math. Ann.*, 77:454–465, 1916.
619. D. König. Graphs and matrices (in Hungarian). *Mat. Fiz. Lapok*, 38:116–119, 1931.
620. D. König. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, 1936.
621. J.-C. König, D.W. Krumme, and E. Lazard. Diameter preserving orientation of the torus. *Networks*, 32:1–11, 1998.
622. B. Korte and J. Vygen. *Combinatorial Optimization*. Springer, Berlin, 2000.
623. A. Kotzig. On the theory of finite graphs with a linear factor II. *Math. Fyz. Čazopis*, 9:73–91, 1959.
624. A. Kotzig. Moves without forbidden transitions in a graph. *Math. Fyz. Čazopis*, 18:76–80, 1968.
625. A. Kotzig. The decomposition of a directed graph into quadratic factors consisting of cycles. *Acta Fac. Rerum Natur. Univ. Comenian. Math. Publ.*, 22:27–29, 1969.
626. S. Kreutzer and S. Ordyniak. Digraph decomposition and monotonicity in digraph searching. Manuscript, March 2008.

627. M. Kriesell. Disjoint A-paths in digraphs. *J. Combin. Theory Ser. B*, 95(1):168–172, 2005.
628. D. Kühn and D. Osthus. Linkedness and ordered cycles in digraphs. *Combin. Probab. Comput.*, 17:411–422, 2008.
629. D. Kühn, D. Osthus, and A. Young. A note on complete subdivisions in digraphs of large out-degree. *J. Graph Theory*, 57:1–6, 2008.
630. D. Kühn, D. Osthus, and A. Young. k-Ordered Hamilton cycles in digraphs. *J. Combin. Theory Ser. B*, to appear, 2008.
631. A. Kunzmann and H.J. Wunderlich. An analytical approach to the partial scan problem. *J. Electron. Testing Theory Appl.*, 1:163–174, 1990.
632. C. Kuratowski. Sur le probléme des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.
633. J.M. Laborde, C. Payan, and N.H. Xuong. Independent sets and longest directed paths in digraphs. *Teubner-Texte Math.*, 59:173–177, 1983.
634. H.G. Landau. On dominance relations and the structure of animal societies III. The condition for a score structure. *Bull. Math. Biophys.*, 15:143–148, 1953.
635. M. Las Vergnas. Sur les arborescences dans un graphe orienté. *Discrete Math.*, 15(1):27–39, 1976.
636. E.L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart & Winston, New York, 1976.
637. E.L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Discrete Math.*, 2:75–90, 1978. Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).
638. O. Lee and Y. Wakabayashi. Note on a min-max conjecture of Woodall. *J. Graph Theory*, 38(1):36–41, 2001.
639. C.E. Leiserson and J.B. Saxe. Retiming synchronous circuitry. *Algoritmica*, 6:5–35, 1991.
640. L.A. Levin. Universal sorting problems. *Probl. Inf. Transm.*, 9:265–266, 1973.
641. M. Lewin. On maximal circuits in directed graphs. *J. Combin. Theory Ser. B*, 18:175–179, 1975.
642. H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation*. Prentice Hall, 1998.
643. H. Li and J. Shu. The partition of a strong tournament. *Discrete Math.*, 290:211–220, 2005.
644. N. Lichiardopol, A. Pór, and J.-S. Sereni. A step towards the Bermond-Thomassen conjecture about disjoint cycles in digraphs. Manuscript 2007.
645. N. Linial, L. Lovász, and A. Wigderson. Rubber bands, convex embeddings and graph connectivity. *Combinatorica*, 8(1):91–102, 1988.
646. C. Little, K. Teo, and H. Wang. On a conjecture on directed cycles in a directed bipartite graph. *Graphs Combin.*, 13(3):267–273, 1997.
647. J. Liu and H.S. Zhou. Graphs and digraphs with given girth and connectivity. *Discrete Math.*, 132(1-3):387–390, 1994.
648. X. Liu and D.B. West. Line digraphs and coreflexive vertex sets. *Discrete Math.*, 188(1-3):269–277, 1998.
649. Y. Liu, S. Lu, J. Chen, and S.-H. Sze. Greedy localization and color-coding: Improved matching and packing algorithms. In *Proc. IWPEC 2006*, volume 4169 of *Lect. Notes Comput. Sci.*, pages 84–95. Springer, 2006.
650. S.C. Locke and D. Witte. On non-Hamiltonian circulant digraphs of out-degree three. *J. Graph Theory*, 30:319–331, 1999.
651. L. Lovász. On decompositions of graphs. *Stud. Sci. Math. Hung.*, 1:237–238, 1966.

652. L. Lovász. Connectivity in digraphs. *J. Combin. Theory Ser. B*, 15:174–177, 1973.
653. L. Lovász. Coverings and coloring of hypergraphs. *Congr. Numer.*, 8:3–12, 1973.
654. L. Lovász. On two min–max theorems in graph theory. *J. Combin. Theory Ser. B*, 21:96–103, 1976.
655. L. Lovász. *Combinatorial problems and exercises*. North-Holland, Amsterdam, 1979.
656. L. Lovász. Connectivity algorithms using rubber bands. In *Foundations of software technology and theoretical computer science (New Delhi, 1986)*, pages 394–411. Springer, 1986.
657. L. Lovász and M.D. Plummer. *Matching theory*, volume 29. North-Holland, 1986. Ann. Discrete Math.
658. X. Lu. On avoidable and unavoidable trees. *J. Graph Theory*, 22:335–346, 1996.
659. C.L. Lucchesi. *A minimax equality for directed graphs*, PhD thesis, University of Waterloo, Ontario, Canada, 1976.
660. C.L. Lucchesi and M.C.M.T. Giglio. On the connection between the undirected and the acyclic directed two disjoint paths problem. *Ars Combin.*, 47:191–200, 1997.
661. C.L. Lucchesi and D.H. Younger. A minimax theorem for directed graphs. *J. London Math. Soc. (2)*, 17(3):369–374, 1978.
662. J.F. Lynch. The equivalence of theorem proving and the interconnection problem. *(ACM) SIGDA Newsl.*, 5(3):31–36, 1975.
663. W. Mader. Homomorphieeigenschaften und mittlere Kantendichte von Graphen. *Math. Ann.*, 174:265–268, 1967.
664. W. Mader. Minimale n-fach kantenzusammenhängende Graphen. *Math. Ann.*, 191:21–28, 1971.
665. W. Mader. Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen. *Arch. Math. (Basel)*, 23:219–224, 1972.
666. W. Mader. 1-factoren von Graphen. *Math. Ann.*, 201:269–282, 1973.
667. W. Mader. Ecken vom Innen- und Aussengrad n in minimal n-fach kantenzusammenhängenden Digraphen. *Arch. Math. (Basel)*, 25:107–112, 1974.
668. W. Mader. A reduction method for edge-connectivity in graphs. *Ann. Discrete Math.*, 3:145–164, 1978. Advances in graph theory (Cambridge Combinatorial Conf., Trinity College, Cambridge, 1977).
669. W. Mader. Konstruktion aller n-fach kantenzusammenhängenden Digraphen. *Eur. J. Combin.*, 3(1):63–67, 1982.
670. W. Mader. Degree and local connectivity in digraphs. *Combinatorica*, 5(2):161–165, 1985.
671. W. Mader. Minimal n-fach zusammenhängende Digraphen. *J. Combin. Theory Ser. B*, 38(2):102–117, 1985.
672. W. Mader. Ecken von kleinem Grad in kritisch n-fach zusammenhängenden Digraphen. *J. Combin. Theory Ser. B*, 53(2):260–272, 1991.
673. W. Mader. Existence of vertices of local connectivity k in digraphs of large outdegree. *Combinatorica*, 15(4):533–539, 1995.
674. W. Mader. On topological tournaments of order 4 in digraphs of outdegree 3. *J. Graph Theory*, 21(4):371–376, 1996.
675. W. Mader. On vertices of degree n in minimally n-connected graphs and digraphs. In *Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993)*, pages 423–449. János Bolyai Math. Soc., Budapest, 1996.
676. W. Mader. On vertices of out-degree n in minimally n-connected digraphs. *J. Graph Theory*, 39:129–144, 2002.

677. W. Mader. High connectivity keeping sets in graphs and digraphs. *Discrete Math.*, 302:173–187, 2005.
678. F. Maffray. Kernels in perfect line graphs. *J. Combin. Theory Ser. B*, 55:1–8, 1992.
679. V.M. Malhotra, M.P. Kumar, and S.N. Maheshwari. An $O(n^3)$ algorithm for finding maximum flows in networks. *Inf. Processing Lett.*, 7:277–278, 1978.
680. Y. Manoussakis. k-linked and k-cyclic digraphs. *J. Combin. Theory Ser. B*, 48(2):216–226, 1990.
681. Y. Manoussakis. A linear-time algorithm for finding Hamiltonian cycles in tournaments. *Discrete Appl. Math.*, 36(2):199–201, 1992.
682. Y. Manoussakis. Directed Hamiltonian graphs. *J. Graph Theory*, 16(1):51–59, 1992.
683. Y. Manoussakis. Alternating paths in edge-coloured complete graphs. *Discrete Appl. Math.*, 56:297–309, 1995.
684. Y. Manoussakis, M. Spyratos, Zs. Tuza, and M. Voigt. Minimal colorings for properly colored subgraphs. *Graphs Combin.*, 12:345–360, 1996.
685. S. Martello and P. Toth. Finding a minimum equivalent graph of a digraph. *Networks*, 12:89–100, 1982.
686. T. Masuzawa, K. Hagihara, and N. Tokura. An optimal time algorithm for the k-vertex-connectivity unweighted augmentation problem for rooted directed trees. *Discrete Appl. Math.*, 17(1-2):67–105, 1987.
687. D. May. The next generation transputers and beyond. In *Distributed Memory Computing*, volume 487 of *Lect. Notes Comput. Sci.*, pages 7–22. Springer-Verlag, 1991.
688. J.E. McCanna. Orientations of the n-cube with minimum diameter. *Discrete Math.*, 68(2-3):309–310, 1988.
689. W. McCuaig. Intercyclic digraphs. In *Graph structure theory (Seattle, WA, 1991)*, volume 147 of *Contemp. Math.*, pages 203–245. American Mathematical Society, 1993.
690. C. McDiarmid. Probability. In L.W. Beineke et al., editor, *Graph connections. Relationships between graph theory and other areas of mathematics*, pages 194–207. Oxford University Press, Oxford, 1997.
691. K. Mehlhorn. *Data Structures and Algorithms 2: Graph Algorithms and NP-completeness*. Springer-Verlag, Berlin, 1984.
692. D. Meierling and L. Volkmann. All 2-connected in-tournaments that are cycle complementary. *Discrete Math.*, 308:2115–2133, 2008.
693. D. Meierling and L. Volkmann. On arc-traceable local tournaments. *Discrete Math.*, to appear, 2008.
694. V. Melkonian and É. Tardos. Algorithms for a network design problem with crossing supermodular demands. *Networks*, 43(4):256–265, 2004.
695. V. Melkonian and É. Tardos. Primal-dual-based algorithms for a directed network design problem. *INFORMS J. Comput.*, 17(2):159–174, 2005.
696. K. Menger. Zur allgemeinen Kurventheorie. *Fund. Math.*, 10:96–115, 1927.
697. A. Metzlar. Disjoint paths in acyclic digraphs. *J. Combin. Theory Ser. B*, 57(2):228–238, 1993.
698. H. Meyniel. Une condition suffisante d'existence d'un circuit hamiltonien dans un graphe orienté. *J. Combin. Theory Ser. B*, 14:137–147, 1973.
699. M. Miller and I. Fris. Maximum order digraphs for diameter 2 or degree 2. In *Pullman Volume of Graphs and Matrices*, volume 139 of *Lect. Notes Pure Appl. Math.*, pages 269–278. Pullman, New York, 1992.
700. G.J. Minty. A theorem on n-colouring the points of a linear graph. *Amer. Math. Mon.*, 69:623–624, 1962.

701. C.L. Monma and J.B. Sidney. A general algorithm for optimal job sequencing with series-parallel constraints. *Math. Oper. Res.*, 4:215–224, 1977.
702. J.W. Moon. Solution to problem 463. *Math. Mag.*, 35:189, 1962.
703. J.W. Moon. On subtournaments of a tournament. *Can. Math. Bull.*, 9:297–301, 1966.
704. J.W. Moon. *Topics on tournaments*. Holt, Rinehart & Winston, New York, 1968.
705. E.F. Moore. The shortest path through a maze. In *Proc. Int. Symp. on the Theory of Switching*, pages 285–292. Harward University Press, 1959.
706. M. Morvan and L. Viennot. Parallel comparability graph recognition and modular decomposition. In *STACS 96*, pages 169–180. Springer, 1996.
707. H.M. Mulder. Julius Petersen's theory of regular graphs. *Discrete Math.*, 100:157–175, 1992.
708. H. Müller. Recognizing interval digraphs and interval bigraphs in polynomial time. *Discrete Appl. Math.*, 78:189–205, 1997.
709. J.H. Muller and J. Spinrad. Incremental modular decomposition. *J. Assoc. Comput. Mach.*, 36(1):1–19, 1989.
710. K.G. Murty. *Network programming*. Prentice Hall, Englewood Cliffs, NJ, 1992.
711. D. Naddef. Polyhedral Theory and Branch-and-Cut Algorithms for the Symmetric TSP. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 29–116. Kluwer, Dordrecht, 2002.
712. H. Nagamochi and T. Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5-6):583–596, 1992.
713. H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math.*, 5(1):54–66, 1992.
714. H. Nagamochi and T. Ibaraki. Deterministic $\widetilde{O}(nm)$ time edge-splitting in undirected graphs. *J. Combin. Optim.*, 1(1):5–46, 1997.
715. W. Narkiewicz. *Number Theory*. World Scientific, Singapore, 1983.
716. C.St.J.A. Nash-Williams. On orientations, connectivity and odd-vertex-pairings in finite graphs. *Can. J. Math.*, 12:555–567, 1960.
717. C.St.J.A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. London Math. Soc.*, 36:445–450, 1961.
718. C.St.J.A. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964.
719. C.St.J.A. Nash-Williams. Problem 47. In *Proc. Colloq. Tihany 1966*, page 366. Academic Press, 1968.
720. C.St.J.A. Nash-Williams. Hamilton circuits in graphs and digraphs. In *The many facets of graph theory*, volume 110 of *Springer-Verlag Lect. Notes*, pages 237–243. Springer-Verlag, 1969.
721. C.St.J.A. Nash-Williams. Hamilton circuits. In *Studies in Graph Theory Part II*, *Studies in Mathematics* 12, pages 301–360. Mathematical Association of America, Washington, DC, 1975.
722. M. Natu and S.-C. Fang. The point-to-point connection problem–analysis and algorithms. *Discrete Appl. Math.*, 78:207–226, 1997.
723. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.
724. V. Neumann-Lara. The acyclic disconnection of a digraph. Talk at the 16th British Combin. Conference 1997.
725. L.L. Ng. Hamiltonian decomposition of complete regular multipartite digraphs. *Discrete Math.*, 177(1-3):279–285, 1997.
726. L.L. Ng. Hamiltonian decomposition of lexicographic products of digraphs. *J. Combin. Theory Ser. B*, 73(2):119–129, 1998.

727. R. Niedermeier. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, Oxford, 2006.
728. M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.
729. T. Nishimura. Short cycles in digraphs. *Discrete Math.*, 72(1-3):295–298, 1988.
730. Y. Nobert and J.-C. Picard. An optimal algorithm for the mixed Chinese postman problem. *Networks*, 27(2):95–108, 1996.
731. J. Obdržálek. DAG-width: connectivity measure for directed graphs, In *SODA'06: Proc. 17th annual ACM-SIAM symposium on Discrete algorithm*, pages 814–821, New York, NY, USA, 2006, ACM.
732. C. Olsen. Heuristics for combinatorial optimization problems. Course project 1998.
733. O. Ore. *Theory of graphs*. American Mathematical Society, Providence, RI, 1962. *American Mathematical Society Colloquium Publications*, Vol. XXXVIII.
734. Z. Ouyang, N. Memon, T. Suel, and D. Trendafilov. Cluster-based delta compression of a collection of files, In *Proc. 3rd International Conf. on Web Information Systems Engineering (WISE 2002)*, pages 257–268, 2002.
735. M. Overbeck-Larisch. Hamiltonian paths in oriented graphs. *J. Combin. Theory Ser. B*, 21(1):76–80, 1976.
736. M. Overbeck-Larisch. A theorem on pancyclic-oriented graphs. *J. Combin. Theory Ser. B*, 23(2-3):168–173, 1977.
737. C. Özturan. Network flow models for electronic barter exchange. *J. Org. Comput. Electron. Commer.*, 14(3):175–194, 2004.
738. C. Özturan. Resource bartering in data grids. *Sci. Program.*, 12:155–168, 2004.
739. C. Özturan. Used car salesman problem: a differential auction-barter market. *Ann. Math. Artif. Intell.*, 44(3):255–267, 2005.
740. A. Palbom. Complexity of the directed spanning cactus problem. *Discrete Appl. Math.*, 146(1):81–91, 2005.
741. C.H. Papadimitriou. On the complexity of edge-traversing. *J. Assoc. Comput. Mach.*, 23:544–554, 1976.
742. C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
743. C.H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension, In *Proc. 8th Annual Symp. Structure in Complexity Theory*, pages 12–18, San Diego, 1993.
744. R. Pastor and A. Corominas. Branch and win: OR tree search algorithm for solving combinatorial optimization problems. *Top*, 1:169–192, 2004.
745. L.E. Penn and D. Witte. When the Cartesian product of two directed cycles is hypohamiltonian. *J. Graph Theory*, 7:441–443, 1983.
746. Y. Perl and Y. Shiloach. Finding two disjoint paths between two pairs of vertices in a graph. *J. Assoc. Comput. Mach.*, 25:1–9, 1978.
747. J. Petersen. Die Theorie der regulären graphs. *Acta Math.*, 15:193–220, 1891.
748. V. Petrović. Kings in bipartite tournaments. *Discrete Math.*, 173(1-3):187–196, 1997.
749. V. Petrović and C. Thomassen. Kings in k-partite tournaments. *Discrete Math.*, 98(3):237–238, 1991.
750. P.A. Pevzner. DNA physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13, 1995.
751. J. Plesník. Remarks on diameters of orientations of graphs. *Acta Math. Univ. Comenian.*, 46/47:225–236 (1986), 1985.

752. J. Plesník and S. Znám. Strongly geodetic directed graphs. *Acta Fac. Rerum Nat. Univ. Comenianae Math.*, 29:29–34, 1975.
753. L. Pósa. A theorem concerning Hamiltonian lines. *Publ. Math. Inst. Hung. Acad. Sci.*, 7:225–226, 1962.
754. E. Prisner. *Graph dynamics*. Longman, Harlow, 1995.
755. E. Prisner. Line graphs and generalizations—a survey. *Congr. Numer.*, 116:193–229, 1996. Surveys in graph theory (San Francisco, 1995).
756. A.P. Punnen and S. Kabadi. Domination analysis of some heuristics for the asymmetric traveling salesman problem. *Discrete Appl. Math.*, 119(1):117–128, 2001.
757. A.P. Punnen, F. Margot, and S.N. Kabadi. TSP heuristics: domination analysis and complexity. *Algorithmica*, 35:111–127, 2003.
758. J. Radhakrishnan and A. Srinivasan. Improved bounds and algorithms for hypergraph 2-colouring. *Random Struct. Algor.*, 16:4–32, 2000.
759. R. Rado. Note on independence functions. *Proc. London Math. Soc.*, 7:300–320, 1957.
760. V. Ramachandran. A minimax arc theorem for reducible flow graphs. *SIAM J. Discrete Math.*, 3:554–560, 1990.
761. F.P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.
762. B. Randerath, I. Schiermeyer, M. Tewes, and L. Volkmann. Vertex pancyclic graphs. *Discrete Appl. Math.*, 120(1-3):219–237, 2002.
763. R.A. Rankin. A campanological problem in group theory. *Proc. Cambridge Philos. Soc.*, 44:17–25, 1948.
764. J. Rattner. The new age of supercomputing. In *Distributed Memory Computing*, volume 487 of *Lect. Notes Comput. Sci.*, pages 1–6. Springer Verlag, 1991.
765. I. Razgon. Parameterized DFVS and multicut problems on DAGs. In H. Broersma, S. Dantchev, M. Johnson, and S. Szeider, editors, *Proc. ACiD'07, Algorithms and Complexity in Durham 2007*, Texts in Algorithmics. College Publications, 2007.
766. A. Recski. *Matroid theory and its applications in electric network theory and in statics*. Springer-Verlag, Berlin, 1989.
767. S. M. Reddy, D.K. Pradhan, and J. G. Kuhl. Directed graphs with minimal diameter and maximal connectivity, Tech. Rep., School of Engineering, Oakland University, 1980.
768. L. Rédei. Ein kombinatorischer Satz. *Acta Litt. Szeged*, 7:39–43, 1934.
769. B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
770. B.A. Reed and F.B. Shepherd. The Gallai-Younger conjecture for planar graphs. *Combinatorica*, 16(4):555–566, 1996.
771. C.R. Reeves (editor). *Modern heuristic techniques for combinatorial problems*. McGraw-Hill, 1995.
772. C. Rego and F. Glover. Local Search and Metaheuristics. In G. Gutin and A.P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, pages 309–368. Kluwer, Dordrecht, 2002.
773. K.B. Reid. Two complementary circuits in two-connected tournaments. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 321–334. North-Holland, 1985.
774. K.B. Reid. Tournaments: scores, kings, generalizations and special topics. *Congr. Numer.*, 115:171–211, 1996. Surveys in graph theory (San Francisco, 1995).

775. G. Reinelt. *The linear ordering problem: algorithms and applications*. Heldermann Verlag, Berlin, 1985.
776. F. Rendl. Quadratic assignment problems on series-parallel digraphs. *Z. Oper. Res. Ser. A-B*, 30(3):161–173, 1986.
777. P.I. Richards. Precedence constraints and arrow diagrams. *SIAM Rev.*, 9:548–553, 1967.
778. M. Richardson. Solution of irreflective relations. *Ann. Math.*, 58:573–580, 1953.
779. M.B. Richey, R.G. Parker, and R.L. Rardin. An efficiently solvable case of the minimum weight equivalent subgraph problem. *Networks*, 15(2):217–228, 1985.
780. H.E. Robbins. A theorem on graphs with an application to a problem on traffic control. *Amer. Math. Mon.*, 46:281–283, 1939.
781. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs I: Large grids. *SIAM J. Discrete Math.*, 1:199–222, 1988.
782. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs II: Two east-west avenues or north-south streets. *Networks*, 19:221–233, 1989.
783. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs III: Three east-west avenues or north-south streets. *Networks*, 22:109–143, 1992.
784. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs IV: Four east-west avenues or north-south streets. *Discrete Appl. Math.*, 49:331–356, 1994.
785. N. Robertson and P.D. Seymour. Graph minors. XIII: The disjoint paths problem. *J. Combin. Theory Ser. B*, 63:65–110, 1995.
786. N. Robertson, P.D. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Ann. Math.*, 150:929–975, 1999.
787. M. Rosenfeld. Antidirected Hamiltonian paths in tournaments. *J. Combin. Theory Ser. B*, 12:93–99, 1971.
788. M. Rosenfeld. Antidirected Hamiltonian cycles in tournaments. *J. Combin. Theory Ser. B*, 16:234–242, 1974.
789. B. Roy. Nombre chromatique et plus longs chemins d'un graphe. *Rev. Fr. Inf. Rech. Opér.*, 1(5):129–132, 1967.
790. V.I. Rublineckii. Estimates of the Accuracy of Procedures in the Traveling Salesman Problem. *Numer. Math. Comput. Technol.*, no. 4:18–23, 1973. (in Russian).
791. R. Saad. Finding a longest alternating cycle in a 2-edge-coloured complete graph is in RP. *Combin. Probab. Comput.*, 5:297–306, 1996.
792. M.R. Samathan and D.K. Pradhan. The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI. *IEEE Trans. Comput.*, C-38:567–581, 1989.
793. B.K. Sanyal and M.K. Sen. New characterization of digraphs represented by intervals. *J. Graph Theory*, 22:297–303, 1996.
794. V.I. Sarvanov. The mean value of the functional of the assignment problem. *Vestsi Akad. Navuk BSSR Ser. Fiz. Mat. Navuk*, no. 2:111–114, 1976. (in Russian).
795. J.P. Schmidt and A. Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM J. Comput.*, 19(5):775–786, 1990.
796. C.-P. Schnorr. Bottlenecks and edge connectivity in unsymmetrical networks. *SIAM J. Comput.*, 8(2):265–274, 1979.

797. A. Schrijver. Min-max relations for directed graphs. In *Bonn Workshop on Combinatorial Optimization (Bonn, 1980)*, volume 16 of *Ann. Discrete Math.*, pages 261–280. North-Holland, 1982.
798. A. Schrijver. Total dual integrality from directed graphs, crossing families, and sub- and supermodular functions. In *Progress in combinatorial optimization (Waterloo, Ont., 1982)*, pages 315–361. Academic Press, 1984.
799. A. Schrijver. A group-theoretical approach to disjoint paths in directed graphs. *CWI Quarterly*, 6(3):257–266, 1993.
800. A. Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994.
801. A. Schrijver. Paths in graphs and curves on surfaces., In *First European congress of mathematics (ECM), Paris, France, July 6-10, 1992. Volume II: Invited lectures (Part 2). Basel: Birkhauser.[ISBN 3-7643-2799-5/hbk]*, pages 381–406, 1994.
802. A. Schrijver. A combinatorial algorithm for minimizing submodular functions in strongly polynomial time. *J. Combin. Theory Ser. B*, 80:346–355, 2000.
803. A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.
804. A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. B*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Matroids, trees, stable sets, Chapters 39–69.
805. A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. C*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Disjoint paths, hypergraphs, Chapters 70–83.
806. M. Sen, S. Das, A.B. Roy, and D.B. West. Interval digraphs: an analogue of interval graphs. *J. Graph Theory*, 13(2):189–202, 1989.
807. S. Severini. On the Digraph of a Unitary Matrix. *SIAM J. Matrix Anal. Appl.*, 25(1):295–300, 2003.
808. P.D. Seymour. Disjoint paths in graphs. *Discrete Math.*, 29:293–309, 1980.
809. P.D. Seymour. Nowhere-zero 6-flows. *J. Combin. Theory Ser. B*, 30(2):130–135, 1981.
810. P.D. Seymour. Nowhere-zero flows. In *Handbook of combinatorics, Vol. 1, 2*, pages 289–299. Elsevier, Amsterdam, 1995. Appendix: Colouring, stable sets and perfect graphs.
811. P.D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.
812. P.D. Seymour. Packing circuits in Eulerian digraphs. *Combinatorica*, 16(2):223–231, 1996.
813. P.D. Seymour and C. Thomassen. Characterization of even directed graphs. *J. Combin. Theory Ser. B*, 42(1):36–45, 1987.
814. J. Shearer. A property of the colored complete graph. *Discrete Math.*, 25:175–178, 1979.
815. J. Shen. On the girth of digraphs. *Discrete Math.*, 211(1-3):167–181, 2000.
816. J. Shen. On the Caccetta-Häggkvist conjecture. *Graphs Combin.*, 18(3):645–654, 2002.
817. F. B. Shepherd and A. Vetta. Visualizing, finding and packing dijoins. In *Graph theory and combinatorial optimization*, volume 8 of *GERAD 25th Anniv. Ser.*, pages 219–254. Springer, 2005.
818. Y. Shiloach. Edge-disjoint branching in directed multigraphs. *Inf. Process. Lett.*, 8(1):24–27, 1979.
819. Y. Shiloach. A polynomial solution to the undirected two paths problem. *J. Assoc. Comput. Mach.*, 27:445–456, 1980.

820. K. Simon. An improved algorithm for transitive closure on acyclic digraphs. *Theoretical Computer Science*, 58:325–346, 1988.
821. K. Simon. Finding a minimal transitive reduction in a strongly connected digraph within linear time. In *Graph-theoretic concepts in Computer Science (Kerkrade, 1989)*, Lect. Notes Comput. Sci., pages 245–259. Springer-Verlag, 1990.
822. S.S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, New York, 1997.
823. D.J. Skrien. A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. *J. Graph Theory*, 6(3):309–316, 1982.
824. A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs, In *Proc. 11th Annual Eur. Symp. Algorithms, EAS'03*, volume 2832 of *Lect. Notes Comput. Sci.*, pages 482–493, 2003.
825. B. Smetaniuk. A new construction on Latin squares. *Ars. Combin.*, 11:155–172, 1981.
826. J. Soares. Maximum diameter of regular digraphs. *J. Graph Theory*, 16(5):437–450, 1992.
827. L. Šoltés. Orientations of graphs minimizing the radius or the diameter. *Math. Slovaca*, 36(3):289–296, 1986.
828. Z.M. Song. Complementary cycles in bipartite tournaments. *J. Nanjing Inst. Tech.*, 18:32–38, 1988.
829. Z.M. Song. Complementary cycles of all lengths in tournaments. *J. Combin. Theory Ser. B*, 57(1):18–25, 1993.
830. Z.M. Song. Pancyclic oriented graphs. *J. Graph Theory*, 18(5):461–468, 1994.
831. E. Speckenmeyer. On feedback problems in digraphs. In *Proc. 15th WG* , volume 411 of *Lect. Notes Comput. Sci.*, pages 218–231. Springer, 1989.
832. G. Steiner. A compact labeling scheme for series-parallel graphs. *Discrete Appl. Math.*, 11(3):281–297, 1985.
833. M. Stiebitz. Decomposition of graphs and digraphs. *KAM Series in Discrete Mathematics-Combinatorics-Operations Research-Optimization*, 95-309:56–59, 1995.
834. M. Stiebitz. Decomposing graphs under degree constraints. *J. Graph Theory*, 23:321–324, 1996.
835. H.J. Straight. The existence of certain type of semi-walks in tournaments. *Congr. Numer.*, 29:901–908, 1980.
836. X.Y. Su. Paths, cycles, and arc-connectivity in digraphs. *J. Graph Theory*, 19(3):339–351, 1995.
837. B. Sullivan. A summary of results and problems related to the Caccetta-Häggkvist conjecture. Preprint arXiv:math.CO/0605646v1, May 2006.
838. S. Szeider. Finding paths in graphs avoiding forbidden transitions. *Discrete Appl. Math.*, 126(2-3):261–273, 2003.
839. J. Szigeti and Z. Tuza. Generalized colorings and avoidable orientations. *Discuss. Math. Graph Theory*, 17(1):137–145, 1997.
840. J.L. Szwarcfiter. On minimum cuts of cycles and maximum disjoint cycles. In *Graphs and algorithms (Boulder, CO, 1987)*, volume 89 of *Contemp. Math.*, pages 153–166. Amer. Math. Soc., 1989.
841. A.S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 2nd edition, 2001.
842. É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.
843. R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.
844. R.E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.

845. R.E. Tarjan. *Data structures and network algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1983.
846. E.G. Tay. *Optimal orientations of graphs*, PhD thesis, National University of Singapore, Department of Mathematics, 1999.
847. M. Tewes. *In-tournaments and Semicomplete Multipartite Digraphs*, PhD thesis, Lehrstuhl II für Mathematik, RWTH Aachen, 1998.
848. M. Tewes. Pancyclic in-tournaments. *Discrete Math.*, 233(1-3):193–204, 2001.
849. M. Tewes and L. Volkmann. On the cycle structure of in-tournaments. *Australas. J. Combin.*, 18:293–301, 1998.
850. M. Tewes and L. Volkmann. Vertex pancyclic in-tournaments. *J. Graph Theory*, 36(2):84–104, 2001.
851. A. Thomason. Paths and cycles in tournaments. *Trans. Amer. Math. Soc.*, 296(1):167–180, 1986.
852. S. Thomassé. Covering a strong digraph by $\alpha - 1$ disjoint paths: a proof of Las Vergnas' conjecture. *J. Combin. Theory Ser. B*, 83(2):331–333, 2001.
853. C. Thomassen. An Ore-type condition implying a digraph to be pancyclic. *Discrete Math.*, 19(1):85–92, 1977.
854. C. Thomassen. Long cycles in digraphs with constraints on the degrees. In *Surveys in combinatorics (Proc. Seventh British Combinatorial Conf., Cambridge, 1979)*, volume 38 of *London Math. Soc. Lect. Note Ser.*, pages 211–228. Cambridge University Press, 1979.
855. C. Thomassen. 2-linked graphs. *Eur. J. Combin.*, 1:371–378, 1980.
856. C. Thomassen. Hamiltonian-connected tournaments. *J. Combin. Theory Ser. B*, 28(2):142–163, 1980.
857. C. Thomassen. Edge-disjoint Hamiltonian paths and cycles in tournaments. *Proc. London Math. Soc. (3)*, 45(1):151–168, 1982.
858. C. Thomassen. Disjoint cycles in digraphs. *Combinatorica*, 3(3-4):393–396, 1983.
859. C. Thomassen. Connectivity in tournaments. In *Graph theory and combinatorics (Cambridge, 1983)*, pages 305–313. Academic Press, 1984.
860. C. Thomassen. Even cycles in directed graphs. *Eur. J. Combin.*, 6(1):85–89, 1985.
861. C. Thomassen. Hamilton circuits in regular tournaments. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 159–162. North-Holland, 1985.
862. C. Thomassen. The 2-linkage problem for acyclic digraphs. *Discrete Math.*, 55(1):73–87, 1985.
863. C. Thomassen. Sign-nonsingular matrices and even cycles in directed graphs. *Linear Algebra Appl.*, 75:27–41, 1986.
864. C. Thomassen. Counterexamples to Adám's conjecture on arc reversals in directed graphs. *J. Combin. Theory Ser. B*, 42(1):128–130, 1987.
865. C. Thomassen. Paths, circuits and subdivisions. In *Selected topics in graph theory Vol. 3*, pages 97–131. Academic Press, 1988.
866. C. Thomassen. Configurations in graphs of large minimum degree, connectivity, or chromatic number. *Annals of the New York Academy of Sciences*, 555:402–412, 1989.
867. C. Thomassen. Highly connected non-2-linked digraphs. *Combinatorica*, 11(4):393–395, 1991.
868. C. Thomassen. The even cycle problem for directed graphs. *J. Amer. Math. Soc.*, 5(2):217–229, 1992.
869. C. Thomassen. The even cycle problem for planar digraphs. *J. Algor.*, 15:61–75, 1993.

870. M. Thorup. On RAM priority queues, In *Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 59–67, New York, 1996, ACM Press.
871. M. Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. Assoc. Comput. Mach.*, 46:362–394, 1999.
872. K. Thulasiraman and M.N.S. Swamy. *Graphs: theory and algorithms*. John Wiley & Sons, New York, 1992.
873. F. Tian, Z.S. Wu, and C.Q. Zhang. Cycles of each length in tournaments. *J. Combin. Theory Ser. B*, 33(3):245–255, 1982.
874. T.W. Tillson. A Hamiltonian decomposition of K^*_{2m}, $2m \geq 8$. *J. Combin. Theory Ser. B*, 29(1):68–74, 1980.
875. A. Trahtman. The road coloring problem. *Israel J. Math.*, to appear, 2008.
876. W.T. Tutte. The dissection of equilateral triangles into equilateral triangles. *Proc. Cambridge Philos. Soc.*, 44:463–482, 1948.
877. W.T. Tutte. A contribution to the theory of chromatic polynomials. *Can. J. Math.*, 6:80–91, 1954.
878. W.T. Tutte. A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82:99–116, 1956.
879. W.T. Tutte. On the problem of decomposing a graph into n connected factors. *J. London Math. Soc.*, 36:221–230, 1961.
880. Z. Tuza. Graph coloring in linear time. *J. Combin. Theory Ser. B*, 55(2):236–243, 1992.
881. Z. Tuza. Characterization of $(m, 1)$-transitive and $(3, 2)$-transitive semicomplete directed graphs. *Discrete Math.*, 135(1-3):335–347, 1994.
882. J. Urrutia and F. Gavril. An algorithm for fraternal orientation of graphs. *Inf. Process. Lett.*, 41(5):271–274, 1992.
883. J. Valdes, R.E. Tarjan, and E.L. Lawler. The recognition of series parallel digraphs. *SIAM J. Comput.*, 11(2):298–313, 1982.
884. V.V. Vazirani. *Approximation Algorithms*. Springer, Berlin, 2001.
885. L.A. Végh and A.A. Benczúr. Primal-dual approach for directed vertex connectivity augmentation and generalizations. *ACM Trans. Algor.*, 4:1–21, 2008.
886. A. Vetta. Approximating the minimum strongly connected subgraph via a matching lower bound. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 417–426. SIAM, 2001.
887. L.M. Vitaver. Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix. *Dokl. Akad. Nauk SSSR*, 147:758–759, 1962.
888. L. Volkmann. Longest paths in semicomplete multipartite digraphs. *Discrete Math.*, 199:279–284, 1999.
889. L. Volkmann. Spanning multipartite tournaments of semicomplete multipartite digraphs. *Ars Combin.*, 58:271–278, 2001.
890. L. Volkmann. Cycles in multipartite tournaments: results and problems. *Discrete Math.*, 245(1):19–53, 2002.
891. L. Volkmann. Complementary cycles in regular multipartite tournaments, where one cycle has length four. *Kyungpook Math. J.*, 44:219–247, 2004.
892. L. Volkmann. Complementary cycles in regular multipartite tournaments. *Australas. J. Combin.*, 31:119–134, 2005.
893. L. Volkmann. On cycles in regular 3-partite tournaments. *Discrete Math.*, 306(12):1198–1206, 2006.
894. L. Volkmann. Multipartite tournaments: a survey. *Discrete Math.*, 307(24):3097–3129, 2007.
895. L. Volkmann and S. Winzen. Almost regular c-partite tournaments with $c \geq 8$ contain an n-cycle through a given arc for $4 \leq n \leq c$. *Australas. J. Combin.*, 31:61–84, 2005.

896. L. Volkmann and S. Winzen. Every cycle-connected multipartite tournament has a universal arc. *Discrete Math.*, to appear, 2008.
897. L. Volkmann and A. Yeo. Hamiltonian paths, containing a given path or collection of arcs, in close to regular multipartite tournaments. *Discrete Math.*, 281(1-3):267–276, 2004.
898. C. Wang, E.L. Floyd, and M.L. Soffa. Feedback vertex sets and cyclically reducible graphs. *J. Assoc. Comput. Mach.*, 32:296–313, 1985.
899. H. Wang, C. Little, and K. Teo. Partition of a directed bipartite graph into two directed cycles. *Discrete Math.*, 160(1-3):283–289, 1996.
900. S. Warshall. A theorem on boolean matrices. *J. Assoc. Comput. Mach.*, 9:11–12, 1962.
901. R.F. Werner and M.M. Wolf. Bell Inequalities and Entanglement. *Quantum Inf. Comput.*, 1:1–25, 2001.
902. D.B. West. *Introduction to graph theory.* Prentice-Hall, 1996.
903. D.B. West. Short proofs for interval digraphs. *Discrete Math.*, 178(1-3):287–292, 1998.
904. R.J. Wilson. An Eulerian trail through Königsberg. *J. Graph Theory*, 8:265–275, 1986.
905. S. Winzen. *Close to Regular Multipartite Tournaments*, PhD thesis, Aachen University, 2004.
906. G. Woeginger. Exact algorithms for NP-hard problems: A survey. In *Combinatorial Optimization - Eureka! You shrink!*, Lect. Notes Comput. Science, pages 185–207. Springer-Verlag, 2003.
907. D.R. Wood. Bounded degree acyclic decompositions of digraphs. *J. Combin. Theory*, 90:309–313, 2004.
908. D. Woodall. Improper colourings of graphs. In *Graph Colourings*, volume 218 of *Pitman Research Notes in Math. Series*, pages 45–63. Longman, 1990.
909. D. R. Woodall. Menger and König systems. In *Theory and applications of graphs (Proc. Int. Conf., Western Mich. Univ., Kalamazoo, 1976)*, volume 642 of *Lect. Notes Math.*, pages 620–635. Springer, 1978.
910. D. R. Woodall. Minimax theorems in graph theory. In L.W. Beineke and R. Wilson, editors, *Selected topics in graph theory*, pages 237–270. Academic Press, 1978.
911. D.R. Woodall. Sufficient conditions for cycles in digraphs. *Proc. London Math. Soc.*, 24:739–755, 1972.
912. N.C. Wormald. Subtrees of large tournaments. *Lect. Notes Math.*, 1036:417–419, 1983.
913. Z.S. Wu, K.M. Zhang, and Y. Zou. A necessary and sufficient condition for arc-pancyclicity of tournaments. *Sci. Sinica Ser. A*, 25:249–254, 1982.
914. A. Yeo. A note on alternating cycles in edge-coloured graphs. *J. Combin. Theory Ser. B*, 69:222–225, 1997.
915. A. Yeo. One-diregular subgraphs in semicomplete multipartite digraphs. *J. Graph Theory*, 24(2):175–185, 1997.
916. A. Yeo. *Semicomplete Multipartite Digraphs*, PhD thesis, Department of Mathematics and Computer Science, Odense University, Denmark, 1998.
917. A. Yeo. A polynomial time algorithm for finding a cycle covering a given set of vertices in a semicomplete multipartite digraph. *J. Algor.*, 33(1):124–139, 1999.
918. A. Yeo. Diregular c-partite tournaments are vertex-pancyclic when $c \geq 5$. *J. Graph Theory*, 32:137–152, 1999.
919. A. Yeo. How close to regular must a semicomplete multipartite digraph be to secure Hamiltonicity? *Graphs Combin.*, 15:481–493, 1999.

920. A. Yeo. Paths and cycles containing given arcs, in close to regular multipartite tournaments. *J. Combin. Theory Ser. B*, 97(6):949–963, 2007.
921. A. Young. Extremal problems for dense graphs and digraphs, Master's thesis, School of Mathematics, University of Birmingham, 2005.
922. D.H. Younger. Graphs with interlinked directed circuits, In *Proc. Midwest symposium on circuit theory 2*, pages 2–1–XVI 2.7, 1973.
923. D.H. Younger. Integer flows. *J. Graph Theory*, 7(3):349–357, 1983.
924. D.A. Youngs. Minimal orientations of colour critical graphs. *Combinatorica*, 15(2):289–295, 1995.
925. X.-D. Yuan and M.-C. Cai. Vertices of degree k in a minimally k-edge-connected digraph. *Discrete Math.*, 218:293–298, 2000.
926. N. Zadeh. Theoretical efficiency of the Edmonds-Karp algorithm for computing maximal flows. *J. Assoc. Comput. Mach.*, 19:184–192, 1972.
927. E. Zemel. Measuring the quality of approximate solutions to zero-one programming problems. *Math. Oper. Res.*, 6:319–332, 1981.
928. C.Q. Zhang. Every regular tournament has two arc-disjoint hamiltonian cycles. *J. Qufu Normal College*, Special Issue Oper. Res.:70–81, 1980.
929. K.-M. Zhang. Vertex even-pancyclicity in bipartite tournaments. J. Nanjing Univ. Math. Biquarterly, 1:85–88, 1981.
930. K.M. Zhang, Y. Manoussakis, and Z.M. Song. Complementary cycles containing a fixed arc in diregular bipartite tournaments. *Discrete Math.*, 133(1-3):325–328, 1994.
931. K.M. Zhang and J.-Z. Wang. Complementary cycles containing a fixed arc and a fixed vertex in bipartite tournaments. *Ars Combin.*, 35:265–269, 1993.
932. L. Zhao, H. Nagamochi, and T. Ibaraki. A linear time 5/3-approximation for the minimum strongly-connected spanning subgraph problem. *Inf. Process. Lett.*, 86:63–70, 2003.
933. L.-C. Zhao and J.-H. Meng. A sufficient condition for hamiltonian cycles in digraphs. *Ars Combin.*, 32:335–338, 1991.
934. G. Zhou and K. Zhang. A sufficient condition for a semicomplete multipartite digraph to be hamiltonian. *Australas. J. Combin.*, 19:231–234, 1999.

Symbol Index

To shorten and unify notation, in this index we use the following convention:
B denotes a bipartite (di)graph.
C, C_i denote cycles (directed, undirected, edge-coloured, oriented).
D, D_i denote digraphs, directed multigraphs and directed pseudographs.
G, G_i denote undirected graphs and undirected multigraphs.
H denotes a hypergraph.
M denotes a mixed graph or a matroid.
P, P_i denote path (directed, undirected, edge-coloured, oriented).
S denotes a matrix or a multiset.
X, X_i denote abstract sets or sets of vertices.
Y, Y_i denote sets of arcs.

$(D_1, D_2)_D$: set of arcs with tails in $V(D_1)$ and heads in $V(D_2)$, 6
$(E, \mathcal{F})$: specification of independence system, 656
$(X, \prec)$: partial order on X, 521
$(X_1, X_2)_D$: set of arcs with tail in X_1 and head in X_2, 3
$(\Gamma, +)$: an additive group, 436
$(\mathcal{F}, b)$: pair of a family $\mathcal{F}$ and a submodular function b on $\mathcal{F}$, 454
$*P$: P minus the first vertex on P, 298
$>_u$: ordering of neighbours of u, 681
$A(D)$: arc set of D, 2
$A(x)$: arc set of residual network w.r.t. x, 130
$B = (X_1, X_2; E)$: specification of a bipartite graph with bipartition X_1, X_2, 19
$BG(D)$: bipartite representation of D, 19
B^+, B^-: out- and in-branching with no root specified, 339
B_s^+, B_s^-: out- and in-branching rooted at s, 22
$B_{\mathcal{L}}$: bad vertices with respect to the locally optimal ordering $\mathcal{L}$, 602
$CM(D)$: the 2-edge-coloured bipartite multigraph corresponding to the bipartite digraph D, 625
$CM^{-1}(B)$: the bipartite digraph corresponding to the 2-edge-coloured bipartite multigraph B, 625
$C[x_i, x_j]$: subpath of C from x_i to x_j, 13
$C_1 \simeq\!> C_2$: C_1 contains singular vertices with respect to C_2 and they all are out-singular, and C_2 has singular vertices with respect to C_1 and they all are in-singular, 249
$D(G)$: digraph obtained from G via BD-correspondence, 626
$D(d, n, q, r)$: consecutive-d digraph, 47
$D - X$: deleting the vertices of $X \subseteq V(D)$ from D, 7
$D - Y$: deleting the arcs of $Y \subseteq A(D)$ from D, 7
$D//P$: path-contraction, 8
D/D_1: contracting the subdigraph D_1 in D, 7
$D = (V + s, A)$, $D = (V + s, E \cup F)$: specification of D with special vertex s, 553
$D = (V, A)$: specification of D, 2
$D = (V, A, c)$: specification of weighted D, 6
$D[D_1, D_2, \ldots, D_n]$: composing D with $D_1, D_2, \ldots, D_n$, 9
D^p: pth power of D, 10
$D_1 \Rightarrow D_2$: no arc from $V(D_2)$ to $V(D_1)$, 6

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1,

$D_1 \cong D_2$: D_1 is isomorphic to D_2, 7
$D_1 \cup D_2$: union of D_1 and D_2, 11
$D_1 \mapsto D_2$: $V(D_1)$ dominates $V(D_2)$ and no arc from $V(D_2)$ to $V(D_1)$, 6
$D_1 \to D_2$: $V(D_1)$ dominates $V(D_2)$, 6
$D_1 \times D_2 \times \ldots \times D_n$, $\prod_{i=1}^{n} D_i$: Cartesian product of digraphs, 10
$D_B(d,t)$: the de Bruijn digraph, 44
$D_G(d,n)$: generalized de Bruijn digraph, 47
$D_K(d,t)$: the Kautz digraph, 46
D_{ST}: digraph obtained from D by the vertex splitting procedure, 134
$D_{\mathcal{F}}$: digraph associated with a 2-SAT expression, 667
$D\langle X\rangle$: subdigraph of D induced by X, 5
$E(G)$: edge set of the graph G, 18
$G = (V + s, E)$: specification of undirected graph with special vertex s, 442
$G_1 \times G_2 \times \ldots \times G_n$: Cartesian products of graphs, 113
$G_{\mathcal{L}}$: good vertices with respect to the locally optimal ordering $\mathcal{L}$, 602
G_{itd}: graph corresponding to orientability as a locally in-tournament digraph, 427
G_{ltd}: graph corresponding to orientability as a locally tournament digraph, 422
G_{qtd}: graph corresponding to orientability as a quasi-transitive digraph, 418
$H = (V, \mathcal{E})$: specification of the hypergraph H, 26
$K(D)$: the Kirchoff matrix of D, 340
K_n^c: c-edge-coloured complete graph of order n, 635
K_n: complete graph of order n, 19
$K_{n_1,n_2,\ldots,n_p}$: complete multipartite graph, 19
$L(B)$: set of leaves of B, 358
$L(D)$: line digraph of D, 39
$L^k(D)$: iterated line digraph of D, 43
$M = (S, \mathcal{I})$: specification of matroid, 711
$M = (V, A, E)$: specification of the mixed graph M, 24
M^*: dual of the matroid M, 713
$M_1 \vee M_2$: union of matroids M_1 and M_2, 715
$N_D^{+p}(X)$, $N_D^{-p}(X)$: pth out- and in-neighbourhood of X, 88
$N_D^{+p}[X]$, $N_D^{-p}[X]$: closed pth out- and in-neighbourhood of X, 88
$N_D(v)$: neighbourhood of v, 4
$N_D^+(X)$, $N_D^-(X)$: out-neighbourhood, in-neighbourhood of X, 4
$N_D^+(v)$, $N_D^-(v)$: out-neighbourhood and in-neighbourhood of v, 4
$N_G(x)$: neighbourhood of x in G, 20
$O(f(k))$: O-notation, 696
$P(k,l)$: path with two blocks of length k, l, respectively, 434
$P[x_i, x_j]$: subpath of P from x_i to x_j, $i \leq j$, 13
$R^+(X)$: vertices that can be reached from X, 299
$R^-(X)$: vertices that can reach X, 299
$R_l(r,q)$: Ramsey number for l-uniform hypergraphs, 598
$S = [s_{ij}]$: matrix, 2
$SC(D)$: strong component digraph of D, 17
S^T: transpose of matrix S, 2
$TC(D)$: transitive closure of D, 37
TT_s: transitive tournament on s vertices, 418
T_s^+, T_s^- : out- and in-tree rooted at s, 339
T^{rev}: reverse of T, 609
$UG(D)$: underlying graph of D, 20
$UMG(D)$: underlying multigraph of D, 20
$U_{n,k}$: uniform matroid, 713
$V(D)$: vertex set of D, 2
$V(G)$: vertex set of the graph G, 18
X^+, X^-: successors and predecessors of vertices in X, 13
$X_1 \Rightarrow X_2$: no arc from X_2 to X_1, 3
$X_1 \mapsto X_2$: $X_1 \to X_2$ and $X_1 \Rightarrow X_2$, 3
$X_1 \to X_2$: X_1 dominates X_2, 3
$X_1 \times X_2 \times \ldots \times X_p$: Cartesian product of sets, 2
$[n]$: the set $\{1, 2, \ldots, n\}$, 1
$\Delta(G)$: maximum degree of G, 20

$\Delta^+(D)$, $\Delta^-(D)$: maximum out- and in-degree of D, 5
$\Delta^0(D)$: maximum semi-degree of D, 5
$\Delta_{mon}(G)$: maximum monochromatic degree of G, 609
$\Omega(f(k))$: Ω-notation, 696
$\Omega(\mathcal{P})$: intersection graph of the family $\mathcal{P}$ of subgraphs, 624
$\Omega(D)$: maximum number of arc-disjoint dicuts in D, 505
Φ^{ext}: set of extended Φ-digraphs, 10
Φ_0: union of semicomplete multipartite, connected extended locally semicomplete digraphs and acyclic digraphs, 69
Φ_1: union of semicomplete bipartite, connected extended locally semicomplete and acyclic digraphs, 69
Φ_2: union of connected extended locally semicomplete and acyclic digraphs, 69
Ψ: union of transitive and extended semicomplete digraphs, 52
$\Theta(f(k))$: Θ-notation, 696
$\alpha(D)$: independence number of D, 21
$\beta(T)$: the maximum number of arcs in an acyclic subdigraph of T, 585
$\chi(X_1X_2)$: colour of edges between X_1 and X_2, 609
$\chi(e)$: colour of edge e, 609
$\chi(v)$: colour set of v, 609
$\chi_{end}(P)$: colour of last edge of P, 609
$\chi_{start}(P)$: colour of first edge of P, 609
$\chi(H)$: chromatic number of D, 21
$\delta(G)$: minimum degree of G, 20
$\delta^+(D)$, $\delta^-(D)$: minimum out- and in-degree of D, 5
$\delta^0(D)$: minimum semi-degree of D, 5
δ_{ij}^m: length of a shortest (i,j)-path using only internal vertices from $[m-1]$, 100
$\delta_{\geq k}(D)$: the minimum number of arcs in a spanning subdigraph D' of D which has $\delta^0(D') \geq k$, 479
$\delta_{mon}(G)$: minimum monochromatic degree of G, 609
$\delta(P)$: capacity of augmenting path P, 141
$\delta_x(s,t)$: length of a shortest (s,t)-path in $\mathcal{N}(x)$, 147
$\ell^t_{\max}(D)$: maximum number of leaves in an out-tree of D, 361
$\ell_{\max}(D)$: maximum number of leaves in an out-branching of D, 358
$\ell_{\min}(D)$: minimum number of leaves in an out-branching of D, 358
$\epsilon(D)$: size of a minimum dijoin of D, 505
$\eta_k(\mathcal{F})$: deficiency of the family $\mathcal{F}$ of one-way pairs, 564
$\eta_k(X,Y)$: deficiency of the one-way pair (X,Y), 563
$\gamma(S,\overline{S})$: flow demand of the (s,t)-cut $(S,\overline{S})$, 159
$\gamma^*_k(D)$: subpartition lower bound for augmenting the vertex-strong connectivity of D to k, 562
$\gamma_k(D)$: subpartition lower bound for augmenting the arc-strong connectivity of D to k, 557
$\kappa(D)$: vertex-strong connectivity of D, 17
$\kappa(x,y)$: local vertex-strong connectivity from x to y, 192
$\lambda(D)$: arc-strong connectivity of D, 17
$\lambda(x,y)$: local arc-strong connectivity from x to y, 192
$\mathcal{CC}(D)$: the family of cc-sets in D, 652
$\mathcal{CO}(D)$: the family of convex sets in D, 652
$\overleftrightarrow{G}$: complete biorientation of G, 20
$\overleftrightarrow{K}_n$: complete digraph of order n, 35
$\mu(n)$: the nth mediation number, 643
$\mu_D(x,y)$: number of arcs with tail x and head y, 4
$\mu_G(u,v)$: number of edges between u and v in G, 18
$\nu_0(D)$: maximum number of vertex-disjoint cycles in D, 512
$\nu_1(D)$: maximum number of arc-disjoint cycles in D, 512
$\overline{G}$: complement of G, 18
$\overline{K}_n$: graph of order n with no edges, 19

$\overline{x}$: negation of boolean variable x, 667, 702
$\phi(u)$: forefather of u, 196
$\rho(G)$: $\mathrm{diam}_{\min}(G) - \mathrm{diam}(G)$, 108
$\tau_0(D)$: size of a minimum feedback vertex set of D, 583
$\tau_1(D)$: size of a minimum feedback arc set of D, 583
$\theta(D)$: minimum number of arcs whose contraction in D leads to a strong directed multigraph, 506
$\vec{C}_n$: directed cycle on n vertices, 12
$\vec{P}_n$: directed path on n vertices, 12
$a_k(D)$: k-strong augmentation number of D, 563
$a_{\mathcal{F}}$: the number of edges, oriented or not, which enter some $X \in \mathcal{F}$, 366
$b(v)$: balance prescription for the vertex v, 128
b_x: balance vector of the flow x, 128
$bd(F)$: boundary of face F, 72
$c(G)$: the number of connected components of G, 449
$c(Y)$: sum of costs/weights of arcs in Y, 6
$c(a)$: cost/weight of the arc a, 6
$d(X,Y)$: $d^+(X,Y) + d^+(Y,X)$, 191, 192
$d(x)$: degree of x, 20
$d^+(X,Y)$: number of arcs with tail in $X - Y$ and head in $Y - X$, 192
$d_D(X)$: degree of X, 4
$d_D^+(X)$, $d_D^-(X)$: out- and in-degree of X, 4
$d_j(v)$: jth degree of of v, 609
$deor_k^{arc}(D)$: the minimum number of arcs to deorient in D to get a digraph with $\lambda \geq k$, 574
$deor_k^{deg}(D)$: the minimum number of arcs to deorient in D to get minimum degree at least k, 574
$e(X_1, X_2)$: number of edges between X_1 and X_2, 349
$e_G(X)$: number of edges of G with at least one end in X, 449
$e_{\mathcal{F}}$: number of edges connecting different sets of partition $\mathcal{F}$, 453
$f(X_1, X_2)$: sum of f-values over arcs with tail in X_1 and head in X_2, 128
$g(D)$: girth of D, 12
$g_v(D)$: length of a shortest cycle through v in D, 312
$h(X,Y)$: number of vertices not in the one-way pair (X,Y), 563
$h(p)$: height of vertex p, 150
$i_G(X)$: number of edges of G with both ends in X, 449
$i_g(D)$: global irregularity of D, 255
$i_l(D)$: local irregularity of D, 255
$l(\overline{S}, S)$: lower bound of the cut $(\overline{S}, S)$, 157
l_{ij}: lower bound of the arc ij, 127
$p(D)$: period of D, 677
$r(X)$: rank of X, 712
$r^*(X)$: dual rank of X, 714
$r^+(U)$: sum of function values of r on arcs in $(U, \overline{U})$, 453
$r^-(U)$: sum of function values of r on arcs in $(\overline{U}, U)$, 453
$r_k(D)$: minimum number of arcs to reverse in D to obtain a k-strong digraph, 568
$r_k^{arc}(D)$: minimum number of arcs one needs to reverse in D in order to obtain a k-arc-strong directed multigraph, 570
$r_k^{deg}(D)$:minimum number of arcs one needs to reverse in D in order to obtain a directed multigraph D' with $\delta^0(D') \geq k$, 570
r_{ij}: residual capacity of the arc ij, 130
$s(D)$: minimum number of steps for gossiping in D, 691
$sgn(P)$: $-$ if P is an in-path and $+$ if P is an out-path, 298
$u(S, \overline{S})$: capacity of the (s,t)-cut $(S, \overline{S})$, 140
u_{ij}: capacity of the arc ij, 127
$x(S, \overline{S})$: flow across the (s,t)-cut $(S, \overline{S})$, 141
$x(uv)$: value of integer flow x on the arc uv, 436
$x + x'$: arc-sum of flows x and x', 136
$x \to y$: x dominates y, 3

$x \succ y$: x is a descendant of y in a DFS tree, 27
$x^* = x \oplus \tilde{x}$: adding the residual flow $\tilde{x}$ to x, 137
x_i^+, x_i^-: successor and predecessor of x_i, 13
x_{ij}: flow value on the arc ij, 128
$\mathcal{D}_6, \mathcal{D}_8$: classes of non-arc-pancyclic arc-3-cyclic tournaments, 318
$\mathcal{F} = P_1 \cup \ldots \cup P_q \cup C_1 \cup \ldots \cup C_t$: q-path-cycle subdigraph, 14
$\mathcal{N}(D)$: network representation of D, 194
$\mathcal{N}(x)$: residual network w.r.t. x, 130
$\mathcal{N} = (V, A, l, u, b, c)$: specification of the flow network $\mathcal{N}$, 128
$\mathcal{N}_B$: network corresponding to the bipartite graph B, 171
$\mathcal{N}_S = (V, A, f, g, (\mathcal{B}, b), c)$: submodular flow network, 458
$\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$: $\mathcal{S}$ polynomially reducible to $\mathcal{T}$, 701
$\mathcal{T}^*$: set of second powers of even cycles of length at least 4, 283
$\mathcal{T}_4$, $\mathcal{T}_6$: classes of semicomplete digraphs, 283
$\mathbb{Q}_+$: set of positive rational numbers, 1
$\mathbb{Q}_0$: set of non-negative rational numbers, 1
$\mathbb{Q}$: set of rational numbers, 1
$\mathbb{R}_+$: set of positive reals, 1
$\mathbb{R}_0$: set of non-negative reals, 1
$\mathbb{R}$: set of reals, 1
$\mathbb{Z}_+$: set of positive integers, 1
$\mathbb{Z}_0$: set of non-negative integers, 1
$\mathbb{Z}$: set of integers, 1
$\mathrm{Prob}(E)$: probability of the event E, 321
$\mathrm{cc}(D)$: the number of connected convex sets in D, 650
$\mathrm{co}(D)$: the number of convex sets in D, 652
$\mathrm{dagw}(D)$: the DAG-width of a digraph D, 78
$\mathrm{diam}(D)$: diameter of D, 89
$\mathrm{diam}_{\min}(G)$: minimum diameter of an orientation of G, 104
$\mathrm{dist}(X_1, X_2)$: distance from X_1 to X_2, 89
$\mathrm{dist}(x, y)$: distance from x to y, 88
$\mathrm{domn}(\mathcal{H}, n)$: domination number of heuristic H, 661
$\mathrm{domr}(\mathcal{H}, n)$: domination ratio of heuristic H, 661
$\mathrm{dpw}(D)$: the directed path-width of a digraph D, 78
$\mathrm{dtw}(D)$: the directed tree-width of a digraph D, 79
$\mathrm{ext}(X)$: set of elements each of which can extend X to an independent set, 657
$\mathrm{in}(D)$: intersection number of D, 82
$\mathrm{lp}(D)$: length of a longest path in D, 431
$\mathrm{lp}_k(D)$: maximum number of vertices in a k-path subdigraph of D, 543
$\mathrm{pcc}(D)$: path-cycle covering number D, 15
$\mathrm{pcc}^*(D)$: 0 if D has a cycle factor and $\mathrm{pcc}(D)$ otherwise, 481
$\mathrm{pc}(D)$: path covering number of D, 15
$\mathrm{pc}_x(D)$: minimum number of paths in a path factor which starts at x, 277
$\mathrm{pc}^*(D)$: 0 if D is hamiltonian and $\mathrm{pc}(D)$ otherwise, 485
$\mathrm{pred}(x)$: predecessor of x w.r.t. a DFS search, 26
$\mathrm{rad}(D)$: radius of D, 89
$\mathrm{rad}^+(D)$: out-radius of D, 89
$\mathrm{rad}^-(D)$: in-radius of D, 89
$\mathrm{srad}(D)$: strong radius of D, 105
$\mathrm{texpl}(x)$: time when x is explored by a DFS search, 26
$\mathrm{tvisit}(x)$: time when x is visited in a DFS search, 26
$\mathrm{tw}(D)$: the tree-width of a digraph D, 74
$\mathrm{tw}(G)$: the tree-width of a graph G, 74
$\mathrm{vs}(D)$: the vertex separation of a digraph D, 78
$|D|$: the order of the digraph D, 2
$|S|$: cardinality of the multiset S, 2
$|x|$: value of flow x, 132

co-$\mathcal{NP}$: class of co-$\mathcal{NP}$ decision problems, 700

$\mathcal{NP}$: class of $\mathcal{NP}$ decision problems, 700

Author Index

Abouelaoualim, A., 615, 621, 622
Ádám, A., 412, 603
Addario-Berry, L., 434
Adler, R.L., 688
Aharoni, R., 121, 348
Aho, A.V., 38, 699
Ahuja, R.K., 127, 153, 161
Aigner, M., 679
Ailon, N., 587
Ainouche, A., 233
Aldous, J., 127
Alegre, I., 41, 45, 100
Alon, N., 74, 177, 308, 321–323, 327, 328, 332, 337, 361, 372, 467, 513, 514, 528, 547, 550, 585, 587, 598, 637, 662, 704, 705
Alspach, B., 60, 81, 298, 318, 335, 516
Alt, H., 172
Amar, D., 527
Andersson, A., 97
Apartsin, A., 121
Appel, K., 432
Applegate D., 705
Aragon, C.R., 710, 711
Arata, K., 364
Ariyoshi, H., 81
Arkin, E.M., 91, 325, 578
Arora, S., 592
Arts, E.H.L., 710
Assad, A.A., 414
Ausiello, G., 702, 707
Ayel, J., 253
Ayoub, J.N., 225, 332

Baffi, L., 48
Bagchi, A., 369
Baker, R.C., 646
Balas, E., 659
Balcer, Y., 678
Balinski, M., 682
Balister, P., 652
Bampis, E., 302
Bang-Jensen, J., 52–55, 57–63, 65–69, 71, 81, 118, 126, 188, 190, 220, 221, 228, 230–233, 235, 237, 239, 240, 244–250, 252, 256–260, 265, 269, 271, 276, 277, 279–295, 304, 309–315, 347, 353–358, 385–394, 399, 406–410, 415, 422, 426, 428, 468, 481, 482, 484–487, 492–494, 502, 518, 522–524, 526, 528, 536, 542–546, 549, 557, 567, 569, 572–576, 587, 617, 621, 623, 624, 627–631, 641, 656, 657, 694, 705
Bankfalvi, M., 629
Bankfalvi, Z., 629
Bárász, M., 365
Barthélémy, J.-P., 598
Baskoro, E. T., 101
Battista, G., 48
Baum, E.B., 709
Beck, J., 329
Becker, J., 365
Beigel, R., 705
Beineke, L.W., 39, 41, 82, 628
Ben-Arieh, D., 661
Benczúr, A.A., 565
Benkouar, A., 611, 635, 636
Bennett, M.D., 670
Berend, D., 662
Berg, A.R., 208, 447, 578
Berge, C., 119–121, 514
Berman, K.A., 236
Bermond, J.-C., 44, 271, 289, 307, 308, 316, 331, 397, 513, 516
Bertolazzi, P., 48
Berwanger, D., 73, 80
Bessy, S., 519, 527, 530
Bhargava, A., 369
Bienia, W., 441
Biggs, N.L., 23
Bixby, R., 705
Blum, N., 172
Bodlaender, H.L., 75, 77
Boesch, F., 24, 201
Böhme, T., 397
Bollobás, B., 517, 527, 547, 637
Bondy, J.A., 103, 228, 236, 240, 334, 354, 431, 434, 530, 543, 610
Bonsma, P., 361, 363, 704
Boppana, R., 680
Boros, E., 121
Boyd, A., 690
Brandstädt, A., 417
Brandt, S., 269

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1,

Brassard, G., 205, 699
Bratley, P., 205, 699
Bridges, W.G., 101
Brown, D.E., 84
Bruck, R., 647
Burr, S.A., 433
Busacker, R.G., 166
Busch, A.H., 84, 296

Caccetta, L., 329
Cai, M., 212, 368
Cameron, K., 535
Camion, P., 16
Cao, F., 44, 46, 47
Cartwright, D., 35
Cayley, A., 339
Chang, G.J., 268
Charbit, P., 535, 587
Charikar, M., 499, 587
Chartrand, G., 597
Chekuri, C., 499
Chen, C.C., 529, 536, 637
Chen, G., 527, 601
Chen, J., 321, 593, 704, 705
Cheng, E., 561
Cheriyan, J., 153, 205, 489, 490, 492
Cherkassky, B.V., 92
Chetwynd, A.G., 626, 627
Cheung, T., 499
Chow, W.S., 609
Christofides, N., 707
Chudnovsky, M., 121, 584
Chung, F.R.K., 24, 107, 225, 430
Chvátal, V., 103, 105, 119, 122, 238, 330, 705
Climer, S., 706
Cohen, R. F., 48
Cohn, Z., 601
Coleman, T., 709
Conitzer, V., 587
Cook, S.A., 701
Cook, W.J., 145, 705, 711
Coppersmith, D., 37, 322
Cormen, T.H., 195, 699
Corominas, A., 706
Crescenzi, P., 702, 707
Csaba, B., 243
Cunningham, W.H., 145, 459, 711
Curran, S.J., 269
Czumaj, A., 351

Dahl, G., 499
Dahllöf, V., 705
Dai, Z., 499
Dalmazoo, M., 207
Dankelmann, P., 74, 103, 361
Darbinyan, S.K., 234, 236, 289, 308, 309
Darrah, M., 320
Das, K.Ch., 615, 621, 622
Das, P., 628, 630, 631
Das, S., 82, 86
Dawar, A., 73, 80
Daykin, D.E., 637
de Bruijn, N. G., 44
de Fluiter, B., 75
de Werra, D., 426
Dean, N., 601
Demers, A., 358, 360
Deng, X., 368, 423, 472
Di Battista, G., 48
Diestel, R., 441
Dijkstra, E.W., 94
Dilworth, R.P., 521
Ding, G., 396
Dinic, E.A., 148, 406
Dinitz, J., 679, 680
Dinur, I., 593
Dolan, A., 127
van Doorn, E.A., 81
Dorn, F., 361, 363, 704
Dorninger, D., 609, 670, 671
Douglas, R.J., 335
Downey, R.G., 704
Downing, A., 358, 360
Dowsland, K., 710
Du, D.-Z., 41, 44, 46, 47, 268
Duchet, P., 120, 121
Duff, I. S., 674, 675, 677
Duffin, R. J., 48, 51

Eades, P., 48
Edmonds, J., 147, 165, 171, 175, 186, 342, 345–350, 453, 454, 509, 535, 696, 716
Ehrenfeucht, A., 424
Eiben, A.E., 711
El-Sahili, A., 368
Enni, S., 462, 557
Eppstein, D., 705
Erdős, P., 268, 328, 332, 432, 518, 519, 585, 629, 637
Erisman, A. M., 674, 675, 677
Esfahanian, A.H., 205
Euler, L., 23
Even, G., 593
Even, S., 154, 205, 348, 413, 633, 670

Favaron, O., 536
Feder, T., 666
Feldman, J., 499
Feller, W., 677
Fellows, M.R., 704
Feng, J., 635
Feofiloff, P., 511
Ferapontova, E., 121
Fernandez de la Vega, W., 586, 615, 621, 622
Fink, J.F., 268
Fiol, M.A., 41, 45, 100
Fisher, D.C., 601
Fisher, M.J., 37
Fleischer, L., 457, 458
Fleischner, H., 23, 440, 441, 476, 611, 616
Floyd, E.L., 597
Floyd, R.W., 100
Flum, J., 704
Fomin, F.V., 74, 77, 115, 361, 372, 704, 705
Forcade, R., 298
Ford, L.R., Jr., 127, 130, 141, 143, 205, 451
Fortune, S., 375, 382, 398, 400
Fraigniaud, P., 691
Fraisse, P., 292
Frank, A., 202, 209, 347, 350, 365–367, 371, 399, 407, 443, 447, 450–466, 477, 511, 554, 555, 557–567, 577, 580, 597
Fredman, M.L., 96, 97, 323
Frieze, A., 592
Fris, I., 101
Frisch, I.T., 225, 332
Frobenius, G., 174
Fujishige, S., 364, 455, 457–461
Fulkerson, D.R., 127, 130, 141, 143, 205, 342, 451
Funke, M., 592
Fűredi, Z., 103
Fürer, M., 351
Furman, M.E., 37

Gabow, H.N., 205, 424, 494–496, 557, 576
Gale, D., 158, 682
Galeana-Sánchez, H., 82, 119, 120, 426
Galil, Z., 205
Gallai, T., 432, 519, 530, 597
Galluccio, A., 324–327, 338
Galvin, F., 682
Gambosi, G., 702, 707
Garey, M.R., 24, 38, 107, 225, 228, 551, 587, 670, 700, 703
Gavril, F., 426, 587
Geller, D., 597
Gerards, A.M.H., 430, 473
Gerke, S., 652
Germa, A., 331
Ghosh, D., 659, 706
Ghouila-Houri, A., 101, 234, 418
Giesen, H.-E., 635
Giglio, M.C.M.T., 384, 415
Giles, R., 453, 454, 509
Glover, F., 661, 663, 711
Godbole, A., 601
Goddard, W.D., 117, 335, 430
Goddyn, L., 441
Goel, A., 499
Goemans, M.X., 496, 598
Goldberg, A.V., 92, 150, 153, 165
Goldberg, M.K., 101
Goldengorin, B., 659, 662, 706
Golumbic, M.C., 83, 417, 420
Gondran, M., 414
Goodwyn, L.W., 688
Goralcikova, A., 37
Gould, R., 527
Gowen, P.J., 166
Grötzsch, H., 473
Grünbaum, B., 298
Greger, B., 272
Grinberg, È.Y., 603
Grohe, M., 704
Grossman, J.W., 613, 640
Grötschel, M., 266, 457, 565, 592
Guha, S., 499
Guo, Y., 59, 62, 63, 65–68, 237, 245, 255, 256, 283–289, 309, 312–315, 319, 333, 335, 468, 526, 549, 635
Gurvich, V., 121
Gutin, G., 58, 59, 63, 65–69, 71, 74, 75, 86, 103, 108, 111, 112, 116, 117, 123, 126, 178, 190, 221, 228, 231, 233, 235, 237, 239, 240, 244–250, 252, 256–260, 265, 270–273, 276, 277, 280, 281, 288, 290–295, 304, 308, 309, 312–315, 334, 335, 337, 357, 361, 372, 528, 593, 615, 617–621, 623, 624, 627–631, 635, 637, 641,

644–664, 691, 694, 700, 704, 705
Gvozdjak, P., 441
Gyárfás, A., 435, 450, 452, 547, 597

Habib, M., 39
Häggkvist, R., 243, 246, 308, 329, 516, 517, 608, 609, 613, 626, 630, 640
Hagihara, K., 566
Hajnal, A., 690
Haken, W., 432
Hakimi, S.L., 205
Hall, P, 173
Hamidoune, Y.O., 219, 330
Hansen, S., 710
Harant, J., 397
Harary, F., 35, 39, 41, 51, 83, 266, 477
Harman, G., 646
Hassin, R., 578, 707
Havet, F., 298–302, 331, 434, 542, 601
Hedetniemi, S.M., 691
Hedetniemi, S.T., 597, 691
Hell, P., 44, 232, 301, 302, 420–423, 472
Hemminger, R.L., 39, 41
Hendry, G.R.T., 317, 336
Heslop-Harrison, J.S., 670
Hetyei, G., 412
Heuchenne, C., 39
van den Heuvel, J., 363–365
Heydemann, M.-C., 236, 303, 331, 380, 381, 415
Hillier, F.S., 685, 686
Hilton, A.J.W., 626, 627
Hoang, C.T., 330, 331
Hoffman, A.J., 157
Holzman, R., 121
Honami, S., 364
Hoos, H.H., 711
Hopcroft, J.E., 73, 375, 382, 398, 400, 699
Horak, P., 103
Hromkovič, J., 691
Hsu, D.F., 41, 44, 46, 47, 268
Huang, J., 52–54, 58, 60, 61, 115, 118, 126, 232, 233, 240, 246–250, 252, 256, 257, 280, 281, 288, 310–312, 355, 357, 420–426, 468, 472, 482, 485, 492–494, 659, 662
Hubenko, A., 412
Hudry, O., 598
Hunter, P., 73, 80
Hurkens, C., 381
Hwang, F.K., 47, 268

Ibaraki, T., 206, 403, 404, 407
Imase, M., 44, 45, 47
Imori, M., 316
Isaak, G., 597, 598
Itai, A., 670
Itatsu, Y., 364
Ito, H., 364
Itoh, M., 47
Iwata, S., 457, 458

Jackson, B., 244, 271, 272, 347, 399, 446, 459, 461, 467, 516, 528, 557, 578
Jacob, H., 122
Jacobson, M.S., 296
Jaeger, F., 439–441
Jäger, G., 659
Janssen, J., 680
Jensen, T.R., 432, 441, 635, 662
Jewell, W.S., 166
Jirásek, J., 269, 603, 604
Jocke, S.C., 269
Johnson, D.B., 125
Johnson, D.S., 228, 551, 587, 670, 700, 703, 709–711
Johnson, E.L., 175
Johnson, M., 363–365
Johnson, T., 73, 80, 230, 286, 379
Johnstone, A., 652
Jones, N., 270, 643, 644, 648
Jonsson, P., 705
Jordán, T., 208, 217, 447, 557, 561–567, 569, 576, 578
Jung, H.A., 379
Jünger, M., 592

Kabadi, S., 663, 664
Kabell, J. A., 83
Kaneko, Y., 601
Kanj, I.A., 704, 705
Kann, V., 702, 707
Kannan, R., 592
Kaplan, H., 592
Kapoor, S., 342
Kari, J., 688
Karp, R.M., 147, 165, 186, 344, 432, 587, 701
Karpinski, M., 615, 621, 622
Karzanov, A.V., 150, 406
Keevash, P., 243, 517

Kelly, L., 244
Kelly, P.J., 516
Kemeny, J.G., 677
Kemnitz, A., 272
Kenyon-Mathieu, C., 593
Khuller, S., 483, 497–498, 707
Kim, E.J., 74, 361, 617–621, 705
Király, Z., 443–446
Kirchoff, G., 341
Kirousis, L.M., 78
Klasing, R., 691
Klein, M., 163
Kleinberg, J., 699
Kleinschmidt, P., 170
Kleitman, D., 430
Kloks, T., 75, 77, 704, 705
Knuth, D.E., 33
Knyazev, A.V., 103
Kocay, W., 172
Koh, K.-M., 111, 112, 114, 116, 123, 327
Koller, A.E., 662, 664
Komlós, J., 323
König, D., 34, 172, 188
König, J-C., 114
Korst, J.H.M., 710
Kostochka, A, 585
Kotzig, A., 515, 610
Koubek, V., 37
Kouider, M., 368
Krarup, J., 51
Kreutzer, S., 73, 74, 80
Kriesell, M., 223
Krivelevich, M., 74, 361, 372, 662, 704, 705
Krumme, D.W., 114
Kubicki, G., 117
Kuhl, J. G., 47
Kühn, D., 243, 244, 381, 412, 517
Kumar, M.P., 182
Kunth, D.E., 679
Kunzmann, A., 583
Kuratowski, C., 72
Kwak, J.H., 335

Laborde, J.M., 542
Laguna, M., 711
Landau, H.G., 115, 449
Las Vergnas, M., 359
Latka, B.J., 601
Lawler, E.L., 47, 48, 51, 52, 100
Lazard, E., 114, 691
Lee, C.M., 75, 704, 705
Lee, O., 512
Lehel, J., 547
Leiserson, C.E., 195, 583, 699
Lesniak-Foster, L., 268
Levin, L.A., 701
Lewin, M., 271, 289
Li, H., 233, 235, 237, 239, 271, 525, 527
Li, M., 499
Li, X., 119
Lichiardopol, N., 513
Lieberman, G.J., 685, 686
Liestman, A., 691
Lifshitz, E.M., 663
Linden, N., 643, 648
Linial, N., 205, 328, 337
Little, C., 527, 628
Liu, J., 332
Liu, X., 44, 236
Liu, Y., 320, 321, 593, 704, 705
Lloyd, E.K., 23
Locke, S.C., 601
Loebl, M., 324–327, 338
Lovász, L., 104, 122, 201, 205, 221, 346, 442, 457, 460, 506, 565
Lu, S., 321, 593, 704, 705
Lu, X., 353
Lucchesi, C.L., 384, 415, 506, 590
Lundgren, J.R., 84
Lynch, J.F., 379, 395
Lyuu, Y.-D., 41

Mader, W., 207, 208, 212, 213, 215–219, 410, 411, 442, 443, 555, 557
Maffray, F., 683
Magnanti, T.L., 127, 153, 161
Maheshwari, S.N., 153, 182
Makino, K., 364
Malhotra, V.M., 182
Manalastas, P., Jr., 529, 536
Manoussakis, Y., 236, 246, 271, 282, 287, 290, 302, 380, 525, 609, 611, 615, 621, 622, 630, 635, 636
Marchetti-Spaccamela, A., 44, 702, 707
Margot, F., 664
Martinhon, C.A., 615, 621, 622
Massar, S., 643, 648
Masuzawa, T., 566
Matamala, M., 115
Matsumoto, M., 316
May, D., 691

McCanna, J.E., 114
McConnell, R.M., 424
McCuaig, W., 324, 598
McDiarmid, C., 327, 513
McGeoch, L., 711
McGeoch, L.A., 709–711
McMorris, F. R., 83
Megalakaki, O., 609
Mehlhorn, K., 37, 172
Meierling, D., 297, 526
Melkonian, V., 500
Memon, N., 369
Meng, J.-H., 236, 271
Menger, K., 201
Metzlar, A., 384
Meyer, A.R., 37
Meyniel, H., 122, 235
Milgram, A.N., 519
Miller, M., 101
Milner, E. C., 690
Minoux, M., 414
Minty, G.J., 435
Molloy, M., 513
Monien, B., 691
Monma, C.L., 47
Moon, J.W., 16, 115, 126, 318, 585
Morgenstern, O., 120, 126
Morvan, M., 39, 420
Müller, H., 83
Muller, J.H., 420
Munos, X., 44
Murty, K.G., 127
Murty, U.S.R., 610

Naddef, D., 705, 706
Nagamochi, H., 206, 407
Naor, J., 593
Nash-Williams, C.St.J.A., 237–239, 348, 349, 442–444
Neumann-Lara, V., 73, 120, 548
Newman, A., 587
Ng, L.L., 516
Niedermeier, R., 704
Nielsen, M.H., 522–524, 542–546
Nishimura, T., 330
Noble, S.D., 662
Norman, R.Z., 35, 39, 41

Obdržálek, J., 73, 80
Oellermann, O.R., 117, 335
Okada, K., 44, 45, 47
Olsen, C., 709, 710
Ordaz, O., 528
Ordyniak, S., 74
Ore, O., 177
Orlin, J.B., 127, 153, 161
Osthus, D., 243, 244, 381, 412, 517
O'Sullivan, B., 593, 704, 705
Ouyang, Z., 369
Overbeck-Larisch, M., 236, 289, 336
Özturan, C., 683

Palbom, A., 501
Papadimitriou, C.H., 78, 91, 146, 321, 325, 667, 670, 706
Pareek, C.M., 103
Paschos, V., 611, 635, 636
Pastor, R., 706
Paul, M., 172
Payan, C., 542
Peine, R., 691
Pekéc, A., 470
Penn, L.E., 603
Penn, M., 661
Perl, Y., 382
Petersen, J., 431, 609
Petreschi, R., 48
Petrović, V., 116, 119
Pevzner, P.A., 609, 611
Peyrat, C., 44
Pinkernell, A., 335
Pintz, J., 646
Plesník, J., 101, 107, 111
Plummer, M., 201
Poljak, S., 403, 404, 406
Pór, A., 513
Pradhan, D.K., 44, 47
Prisner, E., 41, 58, 115, 232, 417, 426
Protasi, M., 702, 707
Pulleyblank, W.R., 145, 711
Punnen, A.P., 661, 663, 664, 700
Pósa, L., 238

Radhakrishnan, J., 329
Rado, R., 714
Radzik, T., 92
Rafiey, A., 270, 335, 635, 644
Raghavachari, B., 351, 483, 497–498
Ramachandran, V., 597
Raman, V., 704, 705
Ramesh, H., 342
Rampon, J.-X., 39
Ramsey, F.P., 598
Randerath, B., 308
Rankin, R.A., 268
Rapaport, I., 115
Raspaud, A., 527

Ratier, G., 682
Rattner, J., 691
Razgon, I., 74, 593, 705
Recski, A., 711, 716
Reddington, J., 652
Reddy, S. M., 47
Rédei, L., 14
Reed, B., 330, 331, 597, 598, 600
Reeves, C.R., 711
Rego, C., 711
Reid, J. K., 674, 675, 677
Reid, K.B., 116, 296, 449, 525
Reif, J.H., 205
Reinelt, G., 592
Rendl, F., 48
Richards, P.I., 39
Richardson, M., 120
Rivest, R.L., 195, 699
Robbins, H.E., 20, 200
Roberts, F.S., 113, 598
Robertson, N., 73, 80, 121, 230, 286, 324, 379, 597, 600
Rosenfeld, M., 298, 301, 302
Roy, A.B., 82, 86
Roy, B., 432
Ruhl, M., 499
Ryser, H., 647

Saad, R., 611, 615, 621, 622, 629, 635, 636
Sabidussi, G., 611
Safra, S., 593
Saltzman, M.J., 659
Samathan, M. R., 44
Sanyal, B.K., 83
Sarvanov, V.I., 662
Saurabh, S., 74, 361, 372, 704, 705
Saxe, J.B., 583
Schannath, H., 170
Schevon, C., 710, 711
Schieber, B., 593
Schiermeyer, I., 308
Schmidt, J.P., 323
Schnorr, C.-P., 204
Schrijver, A., 145, 395, 396, 401, 455, 457, 458, 461, 511, 565, 711
Schudy, W., 593
Schwenk, A., 51
Scott, A., 547
Scott, E., 652
Sebő, A., 441, 535
Sen, M.K., 82, 83, 86
Sereni, J.-S., 513
Severini, S., 270, 644
Seymour, P.D., 73, 80, 121, 230, 286, 324, 326, 379, 396, 406, 439, 441, 584, 593, 597, 598, 600, 601
Shahar, S., 578
Shamir, A., 670
Shapley, L.S., 682
Shearer, J., 637
Shen, J., 103, 330, 601
Shepherd, F.B., 430, 473, 598
Shiloach, Y., 379, 382, 399
Shu, J., 525
Sidney, J.B., 47
Siegel, A., 323, 680
Sierksma, G., 706
Simon, K., 37–39
Skiena, S.S., 100, 662
Škrekovski, R., 548
Skrien, D.J., 422, 428
Smith, J.E., 711
Snell, J.L., 677
Soares, J., 102
Soffa, M.L., 597
Soleimanfallah, A., 652
Šoltés, L., 111
Soneoka, I., 44, 45, 47
Song, Z.M., 308, 525, 527
Sotteau, D., 303, 331, 380, 381, 415, 516
Spencer, J.H., 327, 332, 637
Spinrad, J., 420
Spyratos, M., 609, 636
Srinivasan, A., 329
Steiglitz, K., 146, 667, 670, 706
Stein, C., 195, 699
Steiner, G., 48
Stiebitz, M., 549, 585
Stockmeyer, L., 670, 703
Stone, D., 172
Straight, H.J., 298
Strothmann, W.B., 351
Stützle, T., 711
Su, X., 557
Sudan, M., 593
Suel, T., 369
Sullivan, B., 584
Sullivan, S.J., 424
Sze, S.-H., 321
Szeider, S., 616, 617
Szekeres, G., 519
Szemerédi, E., 323, 330, 690
Szigeti, J., 435

Szigeti, Z., 443–446
Szwarcfiter, J.L., 597

Tabib, C., 60
Tamassia, R., 48
Tan, B.P., 116
Tardos, É., 165, 461, 496, 500, 699
Tarjan, R.E., 24, 48, 51, 52, 73, 96, 97, 107, 150, 153, 154, 165, 172, 195, 225, 228, 342
Tarsi, M., 441
Tay, E.G., 111, 112, 114, 123
Teo, K., 527
Tesman, B., 598
Tewes, M., 245, 255, 256, 308, 315
Thilikos, D.M., 77
Thomas, R., 73, 80, 121, 230, 286, 324, 379, 597, 600
Thomason, A., 298, 302
Thomassé, S., 122, 244, 298–301, 331, 353, 357, 360, 370, 434, 530, 548, 587, 601
Thomassen, C., 103, 105, 116, 119, 218, 236, 240, 243, 266, 268, 271, 277, 286, 287, 289, 290, 292, 295, 302, 303, 307, 308, 316, 319, 324, 326, 327, 331, 337, 348, 354, 357, 379, 382, 384–388, 390, 397, 406, 412, 435, 446, 467, 468, 511, 513, 514, 516–518, 548, 550, 587, 603, 604
Thorup, M., 97
Thurimella, R., 489, 490, 492
Tian, F., 318
Tian, S.L., 117
Tillson, T.W., 516
Timischl, W., 609
Tindell, R., 24, 201
Toft, B., 432, 441
Tokura, N., 566
Toledano Laredo, V., 426
Tollis, I.G., 48
Tong, L.D., 268
Toueg, S., 101
Trahtman, A., 688
Trendafilov, D., 369
Trotter, T., 268
Tutte, W.T., 339, 349, 397, 437, 439
Tuza, Z., 295, 432, 435, 609, 636
Twitto, Y., 662

Ullman, J.D., 38, 699
Urrutia, J., 426

Valdes, J., 48, 51, 52
Vazirani, V.V., 702
Veblen, O., 515
Végh, L.A., 565
Veinott, A.F., 678
Viennot, L., 420
Vince, A., 431
Vitaver, L.M., 432
Voigt, M., 636
Volkmann, L., 24, 30, 59, 62, 63, 65–68, 228, 245, 255, 256, 272, 283–285, 297, 308, 312–315, 332, 333, 335, 412, 526
von Neumann, J., 119, 126

Wahlström, M., 705
Wakabayashi, Y., 512
Wang, C., 597
Wang, H., 527
Wang, J.Z., 525
Wang, L., 368
Warshall, S., 100
Wegner, E., 611
Weiss, B., 688
West, D.B., 44, 82, 83, 86, 126, 236
Wigderson, A., 205
Williamson, D.P., 496, 598
Wilson, R.J., 23
Winograd, S., 37, 322
Winzen, S., 245, 412
Wirth, A., 709
Witte, D., 269, 603
Woeginger, G., 705
Woodall, D.R., 235, 511, 609
Wright, E., 601
Wu, Z.S., 318
Wunderlich H.J., 583
Wyllie, J., 375, 382, 398, 400

Xia, G., 704, 705
Xu, Y., 113
Xuong, N.H., 542

Yamada, H., 316
Yannakakis, M., 177, 321, 325
Ye, D., 115
Yebra, J.L.A., 41, 45, 100
Yeo, A., 75, 108, 111, 112, 117, 123, 177, 178, 221, 237, 240, 244–247, 253–256, 260, 265, 270, 273, 293–295, 304, 315, 331, 335, 353, 356, 357, 481,

482, 484–487, 492–494, 502, 518, 525, 528, 536, 542–546, 549, 551, 572–576, 587, 593, 613, 624, 641, 644–659, 661, 662, 664, 671, 672, 691, 694, 704, 705
Young, A., 243, 381, 412
Young, N., 483, 497
Younger, D.H., 441, 506, 511, 597
Youngs, D.A., 432
Yuan, X.-D., 212
Yuster, R., 321–323, 601

Zadeh, N., 148
Zamfirescu, C.M., 82
Zemel, E., 707
Zhang, C.Q., 318, 320, 516
Zhang, K.-M., 272, 337, 525
Zhang, W., 706
Zhao, L.-C., 236, 271
Zhou, G., 272
Zhou, H.S., 332
Zhu, A., 497–498
Zhu, X., 103
Ziegler, G., 467, 679
Znám, S., 101
Zverovich, A., 663
Zverovitch, A., 661
Zwick, U., 321–323

Subject Index

abelian group, 436
activity-on-node (AON) project network, 685
acyclic
  arc-decomposition, 548
  orientation, 473
  spanning subdigraph, 322
  subdigraph problem, 606, 718
acyclic digraph, 32–34, 36, 38, 48, 69, 93, 120, 123, 126, 154, 253, 277, 287, 304, 353, 382–385, 400, 430, 480, 486, 520, 521, 550, 566, 567, 580
acyclic ordering, 33, 195, 278, 284
  of strong components, 17
  unique, 84
Ádám's conjecture, 603
adjacency list representation, 84, 697
adjacency matrix, 697
adjacent vertices, 2, 18
admissible pair, 554
all trail (AT) problem, 402, 406
almost
  regular digraph, 516
  transitive tournament, 517
alternating
  cycle subgraph, 632
  Hamilton cycle, 608, 626–631, 670–674
  Hamilton path, 628
  trail in a 2-edge-coloured multigraph, 608
alternating-pancyclic, 610, 628, 630
  2-edge-coloured multigraph, 610
anti-directed
  cycle, 215
  path, 297
  trail, 215, 490
antichain
  of a family of sets, 126, 547
  of a partial order, 521
application of flows, 170–179, 194, 204–206, 224, 279, 303, 304, 448, 451, 476, 478, 480, 504, 520, 534, 561, 580
approximation algorithm, 497, 592
  $f(n)$-approximation algorithm, 592
  feedback arc set problem, 593
  for MSSS problem, 488
  via iterative rounding, 500
  via LP-rounding, 496
arboreal decomposition, 79
arc, 2
  k-critical, 216
  backward with respect to an ordering, 592
  cost, 6
  forward with respect to an ordering, 592
  head, 2
  leaving a set, 192
  ordinary, 295
  tail, 2
  tight, 456
  weight, 6
arc reversal, 6
  effect on vertex-strong connectivity, 568–570
  increasing arc-strong connectivity, 460, 570
  versus augmentation, 569
arc series-parallel (ASP) directed multigraph, 48
arc-k-cyclic, 318, 517
arc-coloured digraph, 607
k-arc-cyclic, 374, 390
arc-disjoint
  (s,t)-paths, 201
  (x,y)-,(y,z)-paths, 409
  2-path problem, 354
  branchings, 205, 345–350, 354–358, 371, 477, 580
  cycles, 13, 513
  dicuts, 505, 507, 510
  dijoins, 511
  feedback arc sets, 511
  hamiltonian cycles, 515, 536
  hamiltonian path and hamiltonian cycle, 517
  in- and out-branchings, 354–358
  paths, 13, 194
  strong subdigraphs, 536
arc-in-locally semicomplete digraph, 82
arc-induced subdigraph, 5

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1,

arc-linkages, 398–410
arc-locally semicomplete digraph, 81–82, 269
arc-out-locally semicomplete digraph, 82
arc-pancyclic digraph, 318–320
k-arc-strong, 17
arc-strong connectivity, 17, 191–226
  k-arc-strong in V, 554
  algorithms, 204
  augmentation, 559, 560
  certificate, 491
k-arc-strong orientation, 476
  of a mixed graph, 477
arc-traceable digraph, 296
arms of chromosome, 670
augmenting, 217
  (s,t)-flow along a path, 141
  arc-strong connectivity, 557–562
  connectivity of a graph, 217
  cycle, 162
  path, 141, 185
    with respect to a matching, 623
  rooted arc-strong connectivity, 370, 580
  set of arcs, 557, 569
  successive arc-connectivity augmentation property, 561
  vertex-strong connectivity, 562, 565–567, 580
automorphism of a digraph, 662
average cost of a hamiltonian cycle, 662

backward arc, 708
  for a path, 361
  on an augmenting path, 141
  with respect to an ordering, 14, 530, 583, 592, 708
bad vertex with respect to a locally optimal ordering, 602
bags, 74
balance vector
  of a flow, 128, 437
  of a network, 128
balanced edge, 424
Balcer-Veinott algorithm, 679, 694
bartering, 683
base
  of a matroid, 712
  of an independence system, 657
BB-correspondence, 625
BD-correspondence, 625
Bellman-Ford-Moore algorithm, 97–98, 124
best-balanced orientation, 443
BFS, *see* breadth-first search
BFS tree, 692
  from a root s, 93
bi-submodular function, 581
biorientation of a mixed graph, 24
bipartite digraph, *see also* semicomplete bipartite digraph, 34
  versus bipartite 2-edge-coloured graph, 625
bipartite graph, 19, 419, 423
  matching, 170
  maximum matching, 170
  perfect matching, 173, 177
  regular, 188
  vertex cover in, 172
bipartite representation, 19, 177, 490, 504
bipartite tournament, *see also* semicomplete bipartite digraph, 35, 104, 246, 282, 336, 431, 525, 628
bivalent digraph, 600
block, 645
$(n_1,\ldots,n_p)$-block-triangular structure, 675
boolean
  matrix multiplication, 321
  multiplication, 667, 703
  variable, 666, 702
branch-and-bound, 706
branchings, 208, 345–372, 495, 566
  application in approximation algorithm, 497
  application of, 205, 358, 497, 519, 538
  arc-disjoint, 205, 345–350
  arc-disjoint in- and out-branchings, 354–358
  minimum cost branchings, 342–345
breadth-first search, 92–93
bridge of a graph, 20, 201, 266
bridgeless graph, 615, 640
buildup algorithm, 166, 167
C-bypass, 230

Caccetta-Häggkvist conjecture, 330
Camion's theorem, 16, 232
canonical minimum cycle factor, 523
capacity

of an (s,t)-cut, 140
of an arc, 127
of an augmenting path, 141
Cartesian product
of digraphs, 10, 268
of sets, 2
certificate
for k-(arc)-strong connectivity, 479–492
for an instance of a decision problem, 700
for strong connectivity via contraction, 488
for vertex-strong connectivity, 490
chain of a partial order, 521
Chinese postman problem, 174
chordal graph, 83, 114, 115, 418
chromatic index, 680
chromatic number of a (di)graph, 21, 121, 431
chromosome arrangement, 670
Chvátal-Erdős condition, 519
circuit matroid, 713
circuit of a matroid, 712
circulant digraph, 80–81, 268
circular arc graph, 418
circulation, 133, 435, 453
decomposition into cycle flows, 137
feasible, 156
Hoffman's circulation theorem, 157
reducing (s,t)-flow to, 133
clause, 667, 702
closed
pth in-neighbourhood, 88
pth out-neighbourhood, 88
walk, 11
closeness among polygonal paths, 396
co-$\mathcal{NP}$, 718
co-disjoint sets, 458
cocircuit of a matroid, 714
coherent cyclic order, 531
colour-coding, 322
colour-connectivity, 622, 623, 627, 629
colour-isomorphic, 609
colourful
path, 322
set, 322
colouring, 21
k-colouring, 431
and orientations, 431
proper, 431
comparability graph, 418, 419
complement of an undirected graph, 18
complementary cycles, 525, 526, 549
complete
p-partite graph, 19
biorientation, 20, 225, 326, 467
of a mixed graph, 24
bipartite graph, 680
digraph, 16, 35
graph, 18
multipartite graph, 19
composition
of digraphs, 9
of graphs, 21, 424
Conjecture, 73, 103, 111, 113, 212, 213, 216, 219, 237–239, 244, 269, 270, 288, 290, 291, 295, 296, 315, 316, 330, 331, 351, 357, 389, 390, 405, 411, 412, 433, 439, 446, 447, 467, 494, 499, 500, 503, 511, 513, 516, 518, 520, 524, 526, 527, 536, 537, 541–543, 548, 549, 566, 567, 569, 570, 601, 603, 616, 618, 621, 622, 635, 637, 646
connected
(g,f)-factor, 273
component, 20
convex (cc) set, 649–652
digraph, 20
graph, 20
k-connected graph, 20
consecutive-d digraph, 47
consistent cycles, 529
spanning pair of, 530
contraction, 7, 405
of a subdigraph, 7
of an arc, 383, 403, 405, 506
of an edge, 206
of cycles, 488
convenient multigraph, 623
converse
of a digraph, 58, 210
of a directed multigraph, 7
convex set, 649–655
cost
of a branching, 342
of a path/cycle, 162
of a vertex, 6
of an arc, 6, 127

cover of a family of sets, 2
covering
  all vertices by cycles, 529–536
  by disjoint paths, 519
  by out-trees, 350
  path covering number, 519
critical
  k-critical arc, 213
  k-critical set, 208
  activities, 686
  colour with respect to a PC trail, 611
  kernel-imperfect, 120
  path, 686
  vertices, 218, 686
critically k-strong digraph, 218–219
cross-free family, 192
crossing
  G-supermodular function, 452, 462
  dicuts, 507
  family, 192
  family of pairs of sets, 581
  pair, 454
  paths, 396
  submodular function, 454, 459
cubic (multi)graph, 438, 473
cut, 17
  (s,t)-cut
    in a graph, 192
    in a network, 140
(s,t)-cut
  in a network
    minimum, 141
cutset, 17
  of a matroid, 714
cycle, *see also* hamiltonian cycle, 12
  k-cycle, 12, 321
  1-maximal, 318, 719
  alternating, 608
  augmenting, 162
  avoiding/containing prescribed arcs, 295
  even, 12
  even cycle problem, 324
  extendable, 317
  finding a cycle of prescribed length, 322
  flow, 136
  length, 12
  longest, 12
  mean cost of a cycle, 165
  modulo k, 337
  negative, 88, 160, 163, 165
  odd, 12
  odd through a fixed arc, 337
  of length $\Theta(\log n)$, 321
  of length k modulo p, 324–329
  of length 0 (mod q), 328
  of minimum mean cost, 165
  ordinary, 60
  oriented, 20
  shortest, 12, 125
  simple with respect to cyclic order, 531
  through a vertex, 12
  with two blocks, 435
cycle canceling algorithm, 163, 188
cycle extendable digraph, 317, 336
cycle factor, 15, 237, 244–259, 481
  k-cycle factor, 527
    with prescribed cycle lengths, 527
  canonical minimum cycle factor, 523
  complexity of finding, 178
  existence of, 177
  good, 249–253
  in regular directed multigraph, 551
  irreducible, 522
  minimum, 521–524
  sufficient condition in terms of independence number, 528
2-cycle factor, *see* complementary cycles
cycle subdigraph, *see also* cycle factor, 15
  t-cycle subdigraph, 15
  covering a prescribed vertex set, 528, 529
  covering specified arcs, 225
  of maximum cardinality, 528
cyclic connectivity, 624
cyclic digraph, 536
  k-cyclic digraph, 255, 374
cyclic independence number, 532
cyclic independent set, 532
cyclic order, 531
  coherent, 531
  simple cycle, 531
cyclically connected digraph, 30
cyclomatic number, 536

DAG, *see* acyclic digraph
DAG-decomposition, 78
DAG-width, 78

k-dangerous set, 475
data compression, 369
data dependency graph (DDG), 649
de Bruijn digraph, 44–47, 316
de-randomizing, 323
decision problem, 700
decomposable digraph, *see also* quasi-transitive digraph, 9, 220, 277
decomposition
  Φ-decomposition of a digraph, 10
  into acyclic digraphs, 548
  into arc-disjoint hamiltonian cycles, 515, 516
  into strong spanning subdigraphs, 536–542
  into strong subdigraphs, 548
  of a graph into cliques, 424
  of the arc set of regular tournaments, 516
decreasing subsequence, 519
deficiency
  of a one-way pair, 563, 564
k-degenerate graph, 433
degree of a vertex
  jth degree, 609
  in a digraph, 4
  in a graph, 20
degree-constrained digraphs, 504
  hamiltonian cycles, 233–243
deletion
  of a subdigraph, 7
  of arcs from a digraph, 7
  of multiple arcs, 6
  of vertices from a digraph, 7
demand arc, 402
demand directed multigraph, 403
density of a digraph, 331
deorienting an arc, 574
dependent set
  of a matroid, 712
  of an independence system, 657
depth-first search, 26–29
descendant in a DFS tree, 27
design, 645
deterministic finite automaton, 687
DFS, *see also* depth-first search, 195
DFS forest, 27
DFS tree, 27
  backward arc, 27
  cross arc, 27
  descendant of a vertex in, 27
  forward arc, 27
  root of, 27
DHM-construction, 630
diameter, 89, 100–115
  minimizing, 101
  minimum in orientation, 103
  Moore bound on number of vertices, 101
  versus degree, 44
dicut, 505
  arc-disjoint, 506
  crossing dicuts, 507
  Woodall's conjecture, 511
difference between two sets, 2
digraph, 2
  corresponding to instance of 2-SAT, 667
Dijkstra's algorithm, 94–97, 123
dijoin, 505
  disjoint dijoins, 511
Dilworth's theorem, 521
3-dimensional matching problem, 551
Dinic's algorithm, 148
  for simple networks, 156
  on unit capacity networks, 154
Dinitz conjecture, proof using kernels, 679–683
directed
  cactus, 501
  cut, *see* dicut
  dual of a planar digraph, 590, 605
  graph, *see also* digraph, 2
  multigraph, 4
  pseudograph, 4, 513
    associated with a Markov chain, 677
directed path decomposition, 78
directed path-width, 78
directed Steiner problem with connectivity constraints, 499
directed tree-width, 79
disjoint cycles, 415, 513, 550, 596–600
  versus feedback sets, 596
disjoint paths, 374
disjoint sets, 2
distance
  from a set to another, 89
  from a vertex to another, 88
distance classes from a vertex, 93
distances
  acyclic digraphs, 94
  algorithms for finding, 91–100

Bellman-Ford-Moore algorithm, 97–98
Dijkstra's algorithm, 94–97
in complete biorientations, 97
dominated, 3
dominated pair of vertices, 233
dominates, 3
dominating pair of vertices, 233
drop algorithm, 497
dual of a matroid, 713
dynamic programming, 322

ear decomposition, 198, 481, 678
application of, 200, 487
ear of, 198
linear algorithm for, 200
edge of an undirected graph, 18
edge-coloured multigraph, 608, 625–628
2-edge-coloured complete multigraph, 628–634
edge-colouring, 438
k-edge-connected, 20, 442
edge-connectivity, 206
algorithm to determine, 206
maximum adjacency ordering, 206
edge-cover, 483
edge-disjoint
2-linkage problem, 406
mixed branchings, 365
paths, 406
spanning trees, 348
trees, 478
Edmonds' branching theorem, 205, 208, 346–350, 370, 372, 399
generalization of, 347
Edmonds-Giles theorem, 454
electronic circuit design, 583
element of a directed pseudograph, 6
elementary operation, 696
ellipsoid method, 457, 565
embedding of a planar (di)graph in the plane, 72
end-vertex
of a walk, 12
of an arc, 2
entering arc, 2
Euler trail, *see* eulerian trail
properly coloured, 610
Euler's formula, 73
Euler's theorem, 23
eulerian
(multi)graph, 441, 443
digraph, 102
directed multigraph, 13, 23, 401–407
orientation of a mixed graph, 452
oriented graph, 103
subgraph, 438
trail, 13, 718
weak linkage problem, 403
eulerian directed multigraph, *see also* regular digraph, 166, 174, 175, 180, 254, 416, 513, 557
decomposition into cycles, 180
Evans Conjecture, 537
even cycle, 12, 35, 428
in a k-regular digraph, 328
oriented graphs with many arcs, 430
even cycle problem, 324
even digraph, 326
even pancyclic, 336
even vertex-pancyclic digraph, 336
exponential-time algorithm, 705
extended Φ-digraph, 10
extended locally in-semicomplete digraph, 35, 86, 407, 409
extended locally out-semicomplete digraph, 276
extended locally semicomplete digraph, 35, 69, 276, 277
extended semicomplete digraph, 35, 52, 69, 246, 251, 252, 276, 280, 288, 309, 310, 336, 393, 481, 484–487, 501, 503, 521
hamiltonian cycle, 246
longest cycle, 246
MSSS problem, 484, 485
extended tournament, 275, 279–281, 288, 470
hamiltonian $[x, y]$-path, 280
proof using the structure of, 275
weakly hamiltonian-connected, 281
extension
of a digraph, 10, 111, 393
of a graph, 21
extension-closed class of digraphs, 10, 36

face of a plane (di)graph, 72
facial cycle, 384, 590
factor of a digraph, 5
family

2-covering, 645
cross-free, 192, 209
crossing, 192
has an SDR, 645
intersecting, 192
laminar, 192, 209
mediated, 645
of sets, 2
symmetric, 645
fan-in, fan-out in eulerian directed multigraphs, 416
feasibility theorem
for circulations, 157
for crossing submodular flows, 458
for flows, 158
for fully submodular flows, 455
feasible
k-commodity flow, 413
flow, 129
with balance vectors within intervals, 185
pairing, 444
submodular flow, 454–458
feedback arc set, 511, 583, 593, 606, 708
feedback arc set problem, 587, 707–711
approximation algorithm, 593, 606
planar digraph, 590
feedback sets, 583–600
versus (arc)-disjoint cycles, 596
feedback vertex set, 583, 705
feedback vertex set problem, 587
Fibonacci heap, 96
finite automaton, 687
fixed-parameter algorithmics, 703
fixed-parameter tractable (FPT), 704
flow, 128
across a cut, 140
adding a residual flow, 138
application, *see* application of flows
arc sum of two flows, 136
augmenting path, 141
balance vector of, 128
circulation, 133
cost of, 129
cycle flow, 136
decomposition, 136, 140, 180
demand of a cut, 159
difference between two flows, 139
feasibility theorem, 158
feasible, 129, 156, 185, 449
integer, 128
maximal, 144, 148
maximum, *see* maximum flow problem
maximum capacity augmenting path method, 185
netto flow, 129
optimal, 162
path flow, 136
residual network with respect to, 130
(s,t)-flow, 132
(s,t)-cut, 140
minimum value, 158
reducing general flows to, 132
relation to arc-strong connectivity in directed multigraphs, 194
value of, 132
Floyd-Warshall algorithm, 37, 99
Ford-Fulkerson algorithm, 142, 205
on real valued instances, 181
forefather, 196
forest, 21
forward arc
for a path, 361
on an augmenting path, 141
with respect to an ordering, 530, 583, 592
fragment, 218
Frank's orientation theorem, 452, 465
Frank-Jordán vertex-connectivity augmentation theorem, 565
free matroid, 713
fully G-supermodular function, 452
fully submodular function, 454

gadget for $\mathcal{NP}$-completeness proof, 228, 375, 551
Gallai-Milgram theorem, 519
Gallai-Roy-Vitaver theorem, 432
game theory, 119
gap of a C-bypass, 241
Gaussian elimination, 674
generalized de Bruijn digraph, 47
generalized matching, 188
generating pair, 82
genetics, 670

geometric random variable, 321
girth, 12, 125, 312, 329–332
global irregularity, 255
good cycle factor, 249–253
  theorem, 250
good vertex with respect to a locally optimal ordering, 602
gossip problem, 690
Grötzsch graph, 473
graph, *see also* undirected graph, 18
graph Steiner problem, 366
greedy algorithm, 371
  for independence systems, 657, 714
  for matroids, 719
group flow, 436

half-duplex gossip problem, 691
Hall's theorem, 173
Hamilton cycle, *see* hamiltonian cycle
Hamilton Cycle Problem, 39, 700
Hamilton path, *see* hamiltonian path
Hamilton walk, *see* hamiltonian walk
hamiltonian (x, y)-path, 286, 288
hamiltonian $[x, y]$-path, 277, 280, 283, 285
hamiltonian connected, 286–289
hamiltonian cycle, 13, 37, 60, 207, 228, 230, 231, 257, 258, 268, 289–296, 303–305, 308–311, 315, 318, 319, 337, 480, 483, 485, 515–518, 529, 552, 662
  alternating in 2-edge-coloured multigraph, 608
  arc-disjoint hamiltonian cycles, 515, 536
  avoiding prescribed arcs, 292–296
  containing prescribed arcs, 290, 292
  in almost acyclic digraph, 304
  in almost semicomplete digraph, 290
  in undirected graph, 238
  multipartite tournament, 293
  power of a hamiltonian cycle, 517
  properly coloured, 621, 635–640
  quasi-transitive digraph, 256–259, 485
  semicomplete multipartite digraph, 244–256
  sufficient conditions in terms of degrees, 233–243
hamiltonian digraph, 13
hamiltonian path, 13, 62, 91, 228, 231, 232, 245, 257, 275–289, 304, 313, 515–518, 530
  alternating in 2-edge-coloured multigraph, 608
  between two prescribed vertices, 277
  in a tournament, 14, 697
  in semicomplete bipartite digraph, 86
  one end vertex prescribed, 275–277
  oriented, 297
  properly coloured, 635
  Rédei's theorem, 14
hamiltonian walk, 13
Havet-Thomassé theorem, 298
k-HCA problem, 290–292
head
  of a one-way pair, 563
  of an arc, 2
height function with respect to a preflow, 150, 182
hereditary set of digraphs, 69
heuristics, 660, 707
  domination number, 661
  domination ratio, 661
  for $\mathcal{NP}$-hard problems, 660, 707–711
Hoffman's circulation theorem, 157, 457
hypergraph, 26, 104, 328
  2-colourable, 26
  2-colouring of, 26
  edge of, 26
  order of, 26
  rank of, 26
  transversal of edges, 364
  uniform, 26
  vertex of, 26
hypertournament, 607

implication class, 420, 423, 424
in-branching, *see also* out-branching, 22, 232
  minimum cost, 497
k-in-critical set, 208, 209
in-degree of a vertex, 4
in-generator of a digraph, 299
in-neighbour, 4
in-neighbourhood, 4

*p*th in-neighbourhood, 88
in-path, 298
in-path-mergeable digraph, 58, 86
in-pseudodegree of a vertex, 4
in-radius, 89
in-singular vertex with respect to a cycle, 247
in-tree, 22
incident to an arc, 3
incomparable elements with respect to a partial order, 521
increasing subsequence, 519
independence number, 21, 85, 254, 519
  effect on cycle factors, 530
independence oracle for a matroid, 343, 715
independence system, 657
  base, 657
  dependent set, 657
  independent set, 657
  uniform, 657
independent arcs (edges), 21
independent set, 427, 519, 718
  of a matroid, 712
  of an independence system, 657
independent set problem, 718
independent vertices, 21
index of a pair of alternating trails, 612
index-bounded weighting, 535
induced subdigraph, 5
initial strong component, 17
initial vertex of a walk, 12
inserting one path into another, 239
instance of a problem, 700
integer multicommodity flow problem, 413
integrality theorem for maximum flows, 144
Intel Δ-prototype, 691
intercyclic digraph, 598
intermediate strong component, 17
internally disjoint paths, 13, 201, 224, 374
intersecting
  G-supermodular function, 452
  family, 192
  pair, 454
  submodular function, 454
intersection
  digraph, 82
  graph, 624
  number of a digraph, 82, 86
  of digraphs, 38
interval digraph, 83
interval graph, 83, 115
interval of an oriented path, 298
  length, 298
2-irreducible instance of k-ST problem, 405
irreducible alternating cycle subgraph, 632
irreducible cycle factor, 522
isomorphic
  directed pseudographs, 7
  graphs, 20
isomorphism, 7
iterated line digraph, 43, 44
iterative compression, 594
iterative rounding of an LP solution, 500

Jordan curve theorem, 395

Kautz digraph, 46
Kelly's conjecture, 516
kernel, 119–122
  (k, l)-kernel, 119
  kernel-imperfect digraph, 120
  kernel-perfect digraph, 119, 680
  kernel-solvable graph, 121
king, 115–119, 126
Kirchoff matrix, 340
König's theorem, 172
Kruskal's algorithm, 342
Kuratowski's theorem, 72

labelled digraph, 7
labelling algorithm for maximum flow, 143
laminar family, 192, 209
Landau's theorem, 449, 476
large packet radio network, 44
Las Vegas algorithm, 205
Latin square, 679
layered network, 146
leaving arc, 2
length
  of a cycle, 12
  of a path, 12
  of a walk, 12
lexicographic 2-colouring, 421, 423
lexicographically smaller vertex, 421
line digraph, 39–44, 119, 316
  iterated, 43
  obstructions for, 42

recognition, 41
linear ordering problem, 592
linear programming, 123, 145, 454, 457, 496, 500, 509, 535
k-linkage, 373, 598
k-linkage problem, 304, 374–395, 398, 414
k-linked digraph, 373, 375, 379–398, 415
linking principle, 450, 461
list chromatic index, 682
list colouring, 680
list edge-colouring, 120, 679–683
literal, 667, 702
local arc-strong connectivity, 192
local edge-connectivity, 443
local in-tournament, *see* locally in-tournament digraph
local irregularity, 255
local tournament, *see* locally tournament digraph
local vertex-strong connectivity, 192
locally in-semicomplete digraph, *see also* locally out-semicomplete digraph, 57–59, 85, 86, 231–233, 271, 273, 315, 426, 427
strong decomposition, 59
structure of non-strong, 59
locally in-tournament digraph, 57
locally optimal ordering, 601
locally optimal solution of an optimization problem, 708
locally out-semicomplete digraph, *see also* locally in-semicomplete digraph
locally semicomplete digraph, 57, 59–68, 111, 221, 225, 233, 283–285, 288, 303, 312–315, 336, 356, 357, 374, 387–389, 415, 421, 422, 425, 526
classification theorem, 68
complementary cycles, 526
extended, 276
generalization, 235
hamiltonian (x, y)-path, 288
hamiltonian $[x, y]$-path, 283, 285
hamiltonian connected, 288
independence number, 85
minimal separating set in, 225
non-round decomposable, 67
orientation of, 468
round decomposable, 62
semicomplete decomposition, 63
structure of non-strong, 61
weakly hamiltonian-connected, 285
locally tournament digraph, *see also* locally semicomplete digraph, 57, 289, 312, 313, 315, 319, 390, 422, 423, 425, 426, 468, 472, 473
characterization through orientations, 425
round, 61
longest
(x, y)-path problem, 287
$[x, y]$-path problem, 288
alternating cycle, 627, 629
cycle
extended semicomplete digraph, 484, 486
relation to chromatic number, 434
cycle problem, 53
path, 232
relation to chromatic number, 432
path problem, 53
acyclic digraph, 94, 123
loop, 4
Lovász's local lemma, 327
Lovász's splitting theorem, 442, 475
lower bound on an arc, 127
removing from a network, 131
Lucchesi-Younger theorem, 506, 509

Mader's directed splitting theorem, 555
main $(n_1, \ldots, n_p)$-blocks, 674
Markov chain, 677
marriage theorem, 174, 451, 476
matching, 21, 223, 443
perfect, 21, 427
matching diagram digraph, 83
matrix multiplication, 37
Matrix-tree theorem, 339
matroid, 343, 711–717, 719
matroid intersection problem, 343, 367, 461, 716, 717, 720
matroid partition problem, 715, 720
MAX-2-SAT, 670, 703, 718
Max-Flow Min-Cut theorem, 141, 172
relation to Menger's theorem, 202
1-maximal cycle, 719

maximal flow, 144, 148
maximal with respect to property $\mathcal{P}$, 2
maximum
  k-path subdigraph, 543
  acyclic subdigraph problem, 592
  adjacency ordering, 206
  in-degree of a digraph, 5
  matching in bipartite graphs, 170, 171
  monochromatic degree, 609
  out-degree of a digraph, 5
  semi-degree of a digraph, 5
  with respect to property $\mathcal{P}$, 2
maximum flow algorithms, 142–156
  capacity scaling algorithm, 183
  Dinic's algorithm, 148
  for unit capacity networks, 154
  Ford-Fulkerson algorithm, 142
  maximum capacity augmenting path method, 186
  MKM algorithm, 182
  on simple networks, 156
  push-relabel algorithm, 150
  shortest augmenting paths, 147
maximum flow problem, 140–156
  and arc-strong connectivity, 204
  in unit capacity networks, 154
  integrality theorem, 144
  re-optimizing after small perturbation, 182
mean cost of a cycle, 165
mediated digraph, 644
mediation number, 644
member
  of a family of digraphs, 7
  of a family of sets, 192
Menger's theorem, 201–206, 212, 224, 225, 287, 345, 347, 348, 370, 409, 410, 415, 443, 459, 469, 529, 557, 562, 580, 606
  applied to sets of vertices, 225
  refinement of, 224
  relation to the Max-Flow Min-Cut theorem, 224
Mergesort, 699
merging paths in a digraph, *see* path-mergeable digraph
meta-heuristics, 707, 709
Meyniel set, 236
Min-Flow Max-Demand theorem, 159
minimal
  (x, y)-path, 53
  vertex series-parallel digraphs, 48
minimally
  k-arc-strong directed multigraph, 207–213
  k-edge-connected multigraph, 442, 475
  k-strong digraph, 213–217
minimizing a submodular function, 457, 459, 478
minimum
  covering out-tree problem, 366
  cycle factor problem, 521
  diameter orientation, 103–115
  diameter versus degree, 44
  dijoin, 510
  equivalent subdigraph, 39, 480
  flow, 158
  in-degree of a digraph, 5
  monochromatic degree, 609
  out-degree of a digraph, 5
  path factor problem, 520
  semi-degree of a digraph, 5
  spanning tree, 342, 371
minimum cost
  branching problem, 342
  cover of directed cuts, 510
  submodular flows, 458, 478, 509
minimum cost flows, 160–170
  application to Chinese postman problem, 174
  applied to a branching problem, 224
  assignment problem, 169
  buildup algorithm, 167
  buildup theorem, 166
  characterization, 163
  cycle canceling algorithm, 163
  integrality theorem, 165
  strongly polynomial algorithm, 165
  transportation problem, 169
minimum equivalent subdigraph, *see* MSSS problem
minimum spanning strong subgraph problem, *see* MSSS problem
mixed (multi)graph, 24, 225, 365, 451
mixed branchings, 365
mixed Chinese postman problem, 175
mixed graph
  arc of, 24

biorientation of, 24
branchings, 365
bridge of, 24
complete biorientation of, 24
connected, 24
edge of, 24
orientation of, 24, 461–466
strong, 24
modular function, 448
monochromatic complete subgraph, 598
Monte Carlo algorithm, 205
Moon's theorem, 16
Moore bound, 101
MSSS problem, 480–489
multi-insertion technique, 239
multicommodity flow, 413
multigraph, 18
multipartite digraph, 34–36
multipartite tournament, *see also* semicomplete multipartite digraph, 35, 112, 116, 117, 126, 247, 267, 272, 282, 293, 315, 316, 333, 526
multiple arcs, 3
multiset, 2

Nash-Williams' orientation theorem, 442, 443, 459, 461, 475
negation, 667, 702
negative cycle, 88
detection, 98
effect on shortest path problems, 91
in residual network, 163
neighbour, 4
neighbourhood, 4, 20
neighbourhood of a solution, 708
neighbouring solutions, 708
nested interval graph, 428
network, 127
augmenting path in, 141
balance vector of, 128
balanced vertex in, 129
capacity of arcs, 127
circulation in, 133
cost of arcs, 127
flow in, 128
layered, 146
lower bound on arcs, 127
maximum flow in, 140
residual with respect to a flow, 130
simple, 155
sink vertex in, 129
source vertex in, 129
unit capacity, 153
with bounds/costs on vertices, 134
network design, 44
network representation, 194, 202
nice tree decomposition, 75
non-deterministic finite automaton, 688
non-trivial λ-cut, 537
normal biorientation, 121
nowhere-zero flow
Γ-flow, 436
k-flow, 435–441
$\mathcal{Z}_k$-flow, 437

2-objective optimization problem, 44
odd
K_4, 429, 473
chain, 284
cycle, 12, 35
cycle through a fixed arc, 337
digraph
(k, p)-odd digraph, 326
necklace, 473
orientation, 428, 430
one-way
communication, 691
cut, 546
pairs, 214, 563–566, 581
set of arcs, 82
O, Ω, Θ-notation, 696
open pth in-neighbourhood, 88
open pth out-neighbourhood, 88
open problem, 121, 146, 206, 231, 268, 280, 292–295, 302, 303, 320, 325, 327, 329, 354, 356, 370, 379, 381, 399, 412, 429, 431, 435, 470–472, 491, 497, 500–502, 515, 518, 523, 524, 527, 547–549, 568, 575, 576, 597, 605, 613, 635, 647, 680, 694
opposite vertices, 283
1-OPT, 708, 720
optimal
augmentation, 557
base of a matroid, 714
flow, 162
submodular flow, 459
optimization problem, 701
order

of a digraph, 2
of functions, 696
order exchange, 612
order reflection, 612
ordered partitioning, 657
ordinary
arc, 60, 295, 574
cycle, 60
path, 60
orientation, 473
as a local tournament, 422
as a quasi-transitive digraph, 418
as a transitive digraph, 418
as an in-tournament digraph, 428
best-balanced, 443
nowhere-zero flows, 435–441
odd, 428
of a graph, 417–453
respecting degree constraints, 448–453
smooth, 443, 475
strong, 20
of mixed graph, 201
well-balanced, 443
orientation of a digraph, 25, 85, 467, 468, 691
orientation of a graph, *see also* orientation, 20, 121, 201, 266, 268, 293, 349, 350, 365, 366, 417–453, 459, 460, 472, 473, 475–478, 631, 681, 682
minimum diameter, 103, 104, 691
with high arc-strong connectivity, 476
with small strong radius, 105
orientation of a mixed graph, 24, 452, 461–466
with small diameter, 107
orientation of a mixed multigraph, 225
orientation of a multigraph, 441
oriented
cycle, 20
forest, 21
graph, 14
hamiltonian cycle in a tournament, 301–303
hamiltonian path, 297–301
path, 20, 298
tree, 21
origin of an oriented path, 298
orthogonal rows in a matrix, 39
out-branching, *see also* in-branching, 22, 58, 339, 519, 705
arc-disjoint, 580
BFS tree, 93
minimum cost, 497
of shortest paths, 90
out-branchings
min/maxleaf problems, 358–363
with bandwidth constraints, 367
k-out-critical set, 208
out-degree of a vertex, 4
out-forest, 368
out-generator of a digraph, 299
out-neighbour, 4
out-neighbourhood, 4
pth out-neighbourhood, 88
out-path, 298
out-path-mergeable digraph, 58
out-pseudodegree of a vertex, 4
out-radius, 89
finite in a weighted digraph, 89
out-singular vertex with respect to a cycle, 247
out-tree, 22, 346, 705
outer face of a plane (di)graph, 72

$\mathcal{P}$, 700
P-gadget, 617–621
P-gadget graph, 618–621
packing cuts, 505
pancircular digraph, 316
pancyclic digraph, 307–316, 336
parallel
architectures, 44
arcs, 3
composition of digraphs, 48
reduction, 50
partial order, 521
comparable elements, 521
p-partite digraph, 34
p-partite graph, 19
partite sets, 19
partition, 2
path, 12
(X, Y)-path, 12
(x, y)-path, 12
$[x, y]$-path, 12
xy-path, 21
anti-directed, 297
arc-disjoint, 201, 374
colourful, 322
crossing, 396

edge-disjoint, 406
even, 12
finding a colourful path of prescribed length, 322
finding a path of prescribed length, 322
good reversal, 476
internally disjoint paths, 374
length, 12
longest, 12
odd, 12
of length $\Theta(\log n)$, 321
ordinary, 60
oriented, 20
vertex-disjoint, 201, 374
path covering number, 15, 245, 258, 519, 521
path factor, 15, 277, 486, 520
path flow, 136
path partition conjecture, 542–546
q-path subdigraph, 15
path-contraction, 8, 310, 664
versus set-contraction, 8
path-cycle covering number, 15, 245, 481, 519
path-cycle factor, 178, 303, 485
q-path-cycle factor, 15
q-path-cycle subdigraph, 14
path-mergeable digraph, 55–57, 85, 230, 231, 271, 287
hamiltonian (x, y)-path, 287
(s, t)-paths
arc-disjoint, 201
internally disjoint, 201
PC, *see* properly coloured
k-perfect family of hash functions, 323
perfect graph, 121
perfect matching, 21, 188, 451, 610, 615, 617–621, 626, 639, 640, 672
in a bipartite graph, 174
of minimum weight in a bipartite graph, 169
period of a directed pseudograph, 677
permutation graph, 83
PERT/CPM , 685–686
Petersen graph, 439, 473, 474
PFx problem, 277
planar digraph, 71–73, 224, 228, 384, 394–401, 404, 406, 407, 416, 511
feedback arc set problem, 590
linkage problem, 395
recognition, 72
vertex-strong connectivity of, 224
planar graph, 71
plane (di)graph, 72
point, 645
polygonal curve, 71
polymatroid, 368, 561
polynomial algorithm, 696
polynomial reduction, 701
polynomial-time approximation scheme (PTAS), 592, 593
power
of a cycle, 224, 283, 303, 526
of a digraph, 10
of a hamilton cycle, 517
of a matrix, 337
of a path, 284, 303, 505
predecessor of a vertex on a path/cycle, 13
preflow, 150, 186, 347
preflow directed multigraph, 347
problem, 700
project scheduling, 685
projective plane, 645
proof technique
BB-correspondence, 627
BD-correspondence, 641
colour-coding, 322
contraction, 506
DHM-construction, 630
divide and conquer, 699
gadgets for $\mathcal{NP}$-completeness proofs, 228, 376, 551
insertion method, 697
iterative rounding, 500
matroid intersection, 461, 720
matroid partition, 720
multi-insertion, 239–243
one-way pairs, 214, 566, 567
probabilistic method, 328, 514, 548, 585, 637–640
random acyclic subdigraph method, 321
reduction to a flow problem, 202, 449, 579, 580
reversing arcs, 469
splitting off arcs, 559
splitting off edges, 441
submodular flows, 366, 459–466, 509
transversals in hypergraphs, 364

uncrossing, 209, 507
using orientations of undirected graphs, 350
using recursive formulas, 45
using submodularity, 202, 346, 554, 555, 559
using the bipartite representation of a directed multigraph , 19
vertex splitting procedure, 201
proper
circular arc graph, 418, 424
orientation as a round local tournament, 422
recognition in linear time, 423
colouring, 21, 431
edge-colouring, 680
interval graph, 115, 472
subset, 2
properly coloured
m-path-cycle subgraph, 608
1-path-cycle subgraph, 620, 621
cycle, 613, 620
cycle subgraph, 608, 618–620
Euler trail, 610
hamiltonian cycle, 621, 635–640
hamiltonian path, 635
path, 620, 621
trail, 608
pseudograph, 18
pseudoregular directed pseudograph, 44, 47
push-relabel algorithm, 150–152, 181

quasi-kernel, 119, 122
quasi-transitive digraph, 52–55, 108–111, 118, 257, 258, 265, 272, 273, 309, 311, 336, 357, 392, 394, 416, 418, 420, 421, 469, 485, 486, 503
hamiltonian cycle, 256–259
highly connected orientation of, 469
minimum cycle factor, 521–524
MSSS problem, 485
path-partition problem, 544–546
recursive characterization, 54
vertex-heaviest paths and cycles, 260–265
quasi-transitive orientation, 418
queue, 92

radius, 89, 100–101
Ramsey's theorem, 598
random acyclic subdigraph method, 321
rank function of a matroid, 712
rank of a matroid, 712
re-weighting the arcs of a digraph, 124, 125, 344
reachable from a vertex, 15
recognition
interval digraphs, 83
line digraph, 41
path-mergeable digraph, 56
planar digraph, 72
round decomposable locally semicomplete digraph, 65
round local tournament digraphs, 472
totally Φ-decomposable digraph, 70
vertex series-parallel digraph, 52
red/blue subgraph of a 2-edge-coloured multigraph, 608
Rédei's theorem, 14, 298
reducible graph, 424
reduction among flow models, 131
redundant arc of a digraph, 37
reference orientation, 448, 459
regular digraph, 5, 46, 102, 225, 228, 254, 525, 536
arc-disjoint cycles in, 513
regular graph, 20
regular oriented graph, 103
reorienting arcs, 459
representation
of a digraph, 82
of a graph, 418
residual network, 130, 137
reversal of a path, 476
reverse of a trail, 609
reversing an arc, 6
reversing arcs, *see* arc reversal
reversing arcs to obtain arc-disjoint branchings, 478
Road Colouring Conjecture, 688
Robbins' theorem, 20, 201, 441, 452
root
of a branching, 22
of a DFS tree, 27
round
decomposition, 62
digraph, 60–61, 332
labelling, 60

round decomposable locally semicomplete digraph, 62–65
routing problems, 414

SAT, 703
2-SAT, 666–670, 702–703
application to orientability as in-tournaments, 426
3-SAT, 667, 703
Satisfiability, *see also* SAT, 667
satisfiable boolean expression, 667, 703
saturated arc, 144
scaling algorithm for maximum flow, 183
scan register, 583
scheduling problems, 47, 186, 520
score of a vertex, 449
score sequence, 449
semi-degree of a vertex, 4
semi-partitioncomplete digraph, 256
semicomplete p-partite digraph, *see* semicomplete multipartite digraph
semicomplete bipartite digraph, 35, 86, 108–111, 246, 252, 272, 276, 277, 337, 525, 528, 627, 628
even pancyclic, 336
hamiltonian cycle, 246
hamiltonian path with one end vertex specified, 276
longest cycle, 246
semicomplete decomposition of a locally semicomplete digraph, 63, 67
semicomplete digraph, 35, 53, 54, 115, 121, 126, 188, 225, 271, 283, 286, 287, 290, 291, 295, 296, 304, 314, 320, 385, 386, 388–391, 393, 394, 407, 410, 414, 468, 469, 516, 528, 548, 551, 567–569
2-linkage problem, 389
hamiltonian (x, y)-path, 286
hamiltonian connected, 287
hamiltonian-connected, 287
highly connected orientation of, 468
vertex-heaviest paths and cycles, 260–265
semicomplete multipartite digraph, 35, 71, 111, 116–118, 244, 245, 247, 249, 251, 253–256, 272, 273, 295, 315, 316, 481, 525, 526
'short' cycles, 332–335
hamiltonian cycle, 244–256
hamiltonian path, 245
longest path, 245
path covering number, 245
regular, 254
separator, 16
(s, t)-separator, 17
minimum, 192
trivial, 287, 391
sequencing problems, 47
2-serf, 116
series composition of digraphs, 48
series reduction, 50
series-parallel digraph, 47–52
k-set, 2
set covering problem, 582
set-contraction, *see* contraction
Seymour's second neighbourhood conjecture, 600
ship loading problem, 161
short cycle in a digraph, 321
shortest
cycle, *see also* girth, 125
path tree form s, 90
paths, 90, 130
k-similar arms of chromosomes, 671
similar size arms of chromosomes, 670
similar vertices, 10, 407
simple network, 155
simulated annealing, 709–711
singular vertex, 247
sink
of a network, 129
of an anti-directed trail, 215
vertex with respect to a flow, 129
sink in a digraph, 4
size of a digraph, 2
smooth orientation, 443, 475
solution of an optimization problem, 708
sorting versus distances in digraphs, 97
source
in a digraph, 4
of a network, 129
of an anti-directed trail, 215
vertex with respect to a flow, 129

source location problem, 363
source-sink connected digraph, 511
spanning out-forest, 368
spanning tree, 21
specific trail (ST) problem, 402, 405
Sperner's lemma, 126, 547
splitting, 554
  vertices, *see* vertex splitting procedure
splitting a vertex, 10
splitting off, 475, 553–562, 580
  admissible, 554
  complete, 556
  in eulerian directed multigraphs, 557
  in mixed graphs, 557
  in undirected graphs, 441
  in undirected multigraphs, 442
  preserving local arc-strong connectivity, 557
  splitting off arcs, 213
stable matching, 681
star hypergraph, 577, 607
state diagram, 687
Steiner tree problem, 366
straight
  digraph, 472
  ordering, 472
strictly alternating cycle, 636
strong
  k-strong, 16
  digraph, 16, 195–198
  orientation, 20, 201, 452
k-strong augmentation number of a digraph, 563, 565
strong component digraph, 17
strong components, 17, 232, 668
  algorithm for finding, 195
  application to finding block-triangular structure in matrices, 676
strong decomposition of a digraph, 17, 59
Strong Perfect Graph theorem, 122
strong radius, 105
strongly connected, *see* strong
strongly polynomial algorithm, 165
subdigraph, 5
  spanning, *see also* factor of a digraph, 5
  with prescribed degrees, 176
    minimum cost, 176
subdivision
  of a digraph, 10, 411
  of an arc, 10, 326, 483, 507
submodular flows, 453–466, 477, 478, 509, 570, 582
submodular function, 193
  minimizing, 457
submodularity
  of (s,t)-cuts, 185
  of matroid rank functions, 713
subpartition, 2
subpartition lower bound, 558, 563, 565
subpath, 13
subtree intersection digraph, 83
successive arc-connectivity augmentation property, 561
successor of a vertex on a path/cycle, 13
sum of boolean variables, 667, 702
superdigraph, 5
supermodular function, 448, 455
  G-supermodular function, 452
switch, 376
(e,f)-switch, 672
symmetric
  design, 645
  digraph, 20
  function, 452
synchronizing string, 688
system of distinct representatives (SDR), 645
Szemerédi regularity lemma
  directed version, 512

tail
  of a one-way pair, 563
  of an arc, 2
telecommunications, 414, 499
terminal strong component, 17
terminal vertex of a walk, 12
terminals of a trail in eulerian directed multigraph, 402
terminus of an oriented path, 298
Thomassen's even cycle theorem, 326
tight
  arc, 456
  set, 202, 456
Tillson's decomposition theorem, 516
time complexity of an algorithm, 696
topological obstruction for linkages, 394
topological sorting, *see* acyclic ordering
total unimodularity, 535

totally Φ-decomposable digraph, 10, 52, 69–71
  hamiltonian cycle, 259
  hamiltonian path, 259
  recognition, 70
  total Φ-decomposition, 10
totally unimodular matrix, 145
tournament, *see also* semicomplete digraph, 14, 84, 104, 115, 117, 122, 126, 188, 218, 224, 225, 278, 279, 286–288, 290, 292, 293, 297–299, 301, 302, 304, 305, 317, 318, 336, 356, 357, 385, 390–392, 407, 415, 449, 467–469, 473, 492, 516–518, 527, 548, 549, 552, 567, 569, 580, 581, 592, 601, 602, 697, 709
  arc-3-cyclic, 318, 517
  complementary cycles, 525, 549
  decomposition into strong subdigraphs, 536
  feedback vertex set problem, 587
  hamiltonian $[x, y]$-path, 277
  weakly hamiltonian-connected, 277
traceable, *see also* hamiltonian path, 13, 14, 236, 530, 697
  arc-traceable digraph, 296
trail, 12
  alternating, 608
M-trail, 610
transitive
  closure, 37, 38
    versus transitive reduction, 38
  digraph, 36, 54, 85, 521, 533
  reduction, 37, 48
  tournament, 84, 597, 719
  triple, 30
transputer-based machine, 691
transversal of a hypergraph, 364
travelling salesman problem, *see* TSP
tree, 21
tree decomposition, 73–77
tree solution to a flow problem, 188
tree-width, 73–77
triangular digraph, 309, 311
trivial (s, t)-separator, 287
trivial separator, 391
truth assignment, 667, 703
TSP, 655, 658–666, 694, 700, 705, 707
k-tuple, 382
Tutte's 5-flow conjecture, 439, 475
two-terminal
  parallel composition, 48
  series composition, 48

unbalanced edge, 424
uncrossing technique, 209, 507
underlying graph, 418–428, 467
underlying graph of a digraph, *see also* underlying graph, 20, 27, 53–55, 70, 85, 201, 231, 294, 343, 349, 384, 396, 407, 417–428, 431, 432, 446, 467, 472, 509, 536, 681, 717
underlying multigraph of a digraph, 20
undirected graph, 18
  non-critical edge of, 217
uniform
  independence system, 657
  matroid, 713
unilateral digraph, 17
union
  of digraphs, 11, 38
  of matroids, 715
unique trail (UT) problem, 402, 404
unit capacity network, 153
universal
  arc, 412
  set, 82
k-universal digraph, 432
upward embedding, 85

value
  of a flow, 132
  of a solution, 708
vertex, 2
  cost, 6
  weight, 6
vertex cover of a bipartite graph, 172
vertex cover problem (VC), 172, 704
vertex even pancyclic digraph, 336
vertex separation, 78
vertex series-parallel (VSP) digraph, 48
vertex series-parallel digraph
  recognition algorithm, 52
vertex splitting, 202
vertex splitting procedure, 134, 202, 398, 512, 685
vertex-arc incidence matrix, 145
vertex-cheapest
  k-path subdigraph, 261
  cycle, 261

vertex-pancyclic digraph, 307, 309, 311, 313, 314
vertex-strong connectivity, 17, 191–226
  algorithms, 204
  certificate, 490
  of complete biorientations, 225
  of extensions of digraphs, 220
  of special classes of digraphs, 220
vertex-weighted directed pseudograph, 6
Volkmann's meta-conjecture, 526

W[1] problems, 704
  W[1]-complete, 704
  W[1]-hard, 704
walk, 11–13
  (x, y)-walk, 11
  Chinese postman walk, 174
weak k-linkage problem, 398–414
  semicomplete digraphs, 409
weak linkages, 373, 374, 398–414
  acyclic digraphs, 400
  eulerian directed multigraphs, 401–407
  generalizations of tournaments, 407–414
k-weak-double-cycle, 326, 337
weakly k-linked
  digraph, 398, 399
  directed multigraph, 373
weakly cycle extendable, 336
weakly hamiltonian-connected, 277–285, 289
weight
  of a subdigraph, 6, 260
  of a vertex, 6
  of an arc, 6
weighted arc-strong connectivity augmentation problem, 561
weighted directed pseudograph, 6
well-balanced orientation, 443
Woodall's conjecture, 511

Yeo's irreducible cycle subdigraph theorem, 253, 293
Younger's conjecture, 597–600

Zemel measure, 707