Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Sciences 176 (2006) 2791–2816

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

The edge-orientation problem and some of its variants on weighted graphs ☆,☆☆

William Chung-Kung Yen *

Department of Information Management, Shih Hsin University, #1, Lane 17, Mu-Cha Rd., Sec. 1, Taipei 116, Taiwan

Received 20 April 2004; received in revised form 25 August 2005; accepted 7 September 2005

Abstract

Let $G(V,E)$ be a connected undirected graph with n vertices and m edges, where each vertex v is associated with a cost $C(v)$ and each edge $e=(u,v)$ is associated with two weights, $W(u\rightarrow v)$ and $W(v\rightarrow u)$. The issue of assigning an orientation to each edge so that G becomes a directed graph is resolved in this paper. Determining a scheme to assign orientations of all edges such that $\max_{x\in V}\left\{C(x)+\sum_{x\rightarrow z}W(x\rightarrow z)\right\}$ is minimized is the objective. This issue is called the edge-orientation problem (the EOP). Two variants of the EOP, the Out-Degree-EOP and the Vertex-Weighted EOP, are first proposed and then efficient algorithms for solving them on general graphs are designed. Ascertaining that the EOP is NP-hard on bipartite graphs and chordal graphs is the second result. Finally, an $O(n\log n)$-time algorithm for the EOP on trees is designed. In general, the algorithmic results in this paper facilitate the implementation of the weighted fair queuing (WFQ) on real networks. The objective of the WFQ is to assign an effective weight for each flow to enhance link utilization. Our findings consequently

☆ This research was supported by the National Science Council, Taiwan, ROC, under the contract number NSC-90-2213-E-128-003.

☆☆ The preliminary version of this paper has been published in the Proceedings of International Computer Symposium 2002, National Dong Hwa University, Hualien, Taiwan, ROC, pp. 910–917.

* Tel.: +886 2 22368225x3365; fax: +886 2 22367114.
E-mail address: ckyen001@ms7.hinet.net

doi:10.1016/j.ins.2005.09.001

can be easily extended to other classes of graphs, such as cactus graphs, block graphs, and interval graphs.

Keywords: Undirected graph; Directed graph; Edge-orientation problem; Trees; NP-hard; Special graphs

1. Introduction

The input graphs of most traditional graph problems, such as minimum domination and minimum spanning trees, are either undirected or directed. The issue of transforming an undirected graph $G = (V, E)$ into a directed graph (digraph) via assigning each edge $e = (u, v)$ an orientation, either from u to v or from v to u, is studied in this paper. Throughout the paper, all input graphs are simple, undirected, and connected. Meanwhile, n and m denote the number of vertices and the number of edges of the input graph, respectively.

Before formally defining the problem, a simple network system of four nodes a, b, c, and d as shown in Fig. 1 will be considered. Assume that the link $e = (u, v)$ connecting any pair of nodes, u and v, can be maintained by the technical staff of either u or v. The different maintenance costs are denoted as $W(u \to v)$ and $W(v \to u)$, respectively. For example, the maintenance costs of the link (c, d) are 46 (by the node c) and 15 (by the node d). Furthermore, each vertex (node) v is associated with a cost $C(v)$ to indicate the fixed cost for maintaining its incident links.

The maintenance cost for each link (edge) $e = (u, v)$ will constitute the total cost of u (or v) if the edge is maintained by u (or v). The two different link-maintenance schemes of Fig. 1 are illustrated in Fig. 2. The italic boldface numbers represent the total cost of each vertex in each scheme. Keeping the maximum total cost of all vertices, denoted as μ *values*, as small as possible is the central

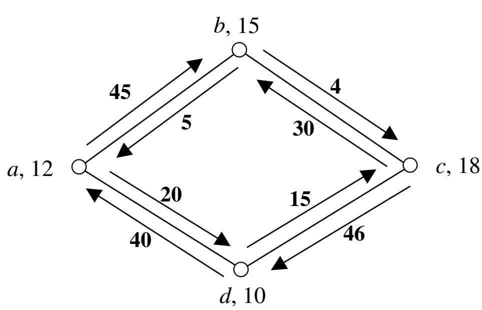

Fig. 1. A simple network system with maintenance costs.

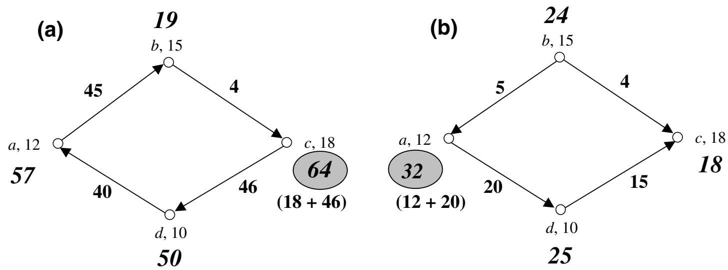

Fig. 2. Two link-maintenance schemes.

issue of this paper. The μ values of scheme (a) and scheme (b) in Fig. 2 are 64 and 32, respectively, and scheme (b) is a link-orientation scheme such that its μ value is minimized.

The above discussions imply the equivalence property of the following two tasks: (1) To determine a link-maintenance scheme such that its μ value is minimized. (2) To assign each edge $e = (u, v)$ an orientation, either from u to v or from v to u, such that $\max_{x \in V}\left\{C(x) + \sum_{x \to z} W(x \to z)\right\}$ is minimized. Theoretically, there are 2^m ways to assign the orientations of all edges and each way is called an *edge-orientation scheme* hereafter. The problem explored in this paper can thus be precisely defined as follows.

The edge-orientation problem (the EOP) [37]: Given a graph $G(V, E)$ in which each vertex v is associated with a cost $C(v)$ and each edge $e = (u, v)$ is associated with two weights, $W(u \to v)$ and $W(v \to u)$. Denote $\mu(A)$ as $\max_{x \in V}\{C(x) + \sum_{x \to z} W(x \to z)\}$, for each edge-orientation scheme A. The value $\sum_{x \to z} W(x \to z)$ is defined as zero if $\text{outdeg}(x) = 0$ within A, where $\text{outdeg}(x)$ is the out-degree of the vertex x. Identifying an edge-orientation scheme A^* such that $\mu(A^*)$ is minimized is the objective. Let $\mu(G) = \min\{\mu(A) \mid A$ is an edge-orientation scheme of $G\}$ hereafter.

We now describe the motivation of the EOP from the perspective of practical applications. Modern information networks, including Internet, private Intranets, and private Extranets, provide people with easy and efficient environments for information access. Achieving high efficiency, quality, and throughput, are the major concerns of information services over various networks. Extensive research has dealt with many fundamental network problems, such as resource allocations (e.g., replication of data objects on a distributed database), quality of service (QoS) and routing, load balancing, and flow controls [1,7,22,24,26–29,35]. In [37,38], we proposed that the link-orientation problem (the LOP) is highly related to the efficient assignment of flow orientations in links of a computer network. In the applications of flow controls, each link is assigned an effective weight corresponding to the switch router for fairness of buffers. A switch router often uses the peer-link to determine the effective weight for each queue (flow). Many approaches for flow controls and related issues in QoS monitoring have been investigated [6,25,26,34]. Among them, a powerful variant of the fair queuing (FQ), called the *weighted fair queuing* (WFQ), has been proposed [31]. The WFQ was designed to allocate resources in a fairer manner and to enhance the utilization of links. In general, a router performing the WFQ must determine the best weight for each flow. Previous literature often assumed that it is possible to split the flow according to the actual weight of each link. The direction of message flows within a link $e = (u, v)$ can be oriented, either from u to v or from v to u, in an asynchronous network. The basic issue is to determine which orientation will be the most helpful for the WFQ and other network applications such as routing and load balancing.

The following two special versions of the EOP have been established to demonstrate the inner spirit and the significance of our research.

The vertex-weighted edge-orientation problem (the Vertex-Weighted EOP) [37]: Given a vertex-weighted graph $G(V, E, C)$, denote $\pi(A)$ as $\max_{x\in V}\{C(x) + \text{outdeg}(x)\}$, for any edge-orientation scheme A. Identifying an edge-orientation scheme A^* such that $\pi(A^*)$ is minimized is the aim. Let $\pi(G) = \min\{\pi(A) \mid A$ is an edge-orientation scheme of $G\}$ hereafter.

The out-degree edge-orientation problem (the Out-Degree-EOP) [37]: Given a graph $G(V, E)$, let $\theta(A) = \max_{v\in V}\{\text{outdeg}(v)\}$, for each edge-orientation scheme A. Obtaining an edge-orientation scheme A^* such that $\theta(A^*)$ is minimized is the objective. Let $\theta(G) = \min\{\theta(A) \mid A$ is an edge-orientation scheme of $G\}$ hereafter.

The Vertex-Weighted EOP is in fact the edge-direction assignment problem (the EDA problem) originally proposed and studied in [39,40]. The EDA Problem has been applied to successfully design linear-time optimal algorithms for solving the searchlight guarding problem on cographs and interval graphs [39,40]. The Vertex-Weighted EOP is also the LOP addressed in [38] and linear-time algorithms were designed to deal with the problem on weighted complete networks and weighted trees. In another, the Out-Degree-EOP plays the key role in solving the bottleneck searchlight guarding problem [37]. A linear-time optimal algorithm for the EOP on complete-split graphs has also been designed by the recursive greedy approach [4,38].

The rest of this paper is organized as follows. A survey of related works on graph orientation problems will be addressed in Section 2. Then, an $O(mn\delta\log\chi)$-time algorithm for the Out-Degree-EOP on general graphs will be proposed in Section 3, where $\delta = \max_{v\in V}\{\deg(v)\}$, in which $\deg(v)$ denotes the degree of any vertex v and $\chi = \max\left\{1, \delta - \left\lceil \frac{m}{n} \right\rceil\right\}$. Section 4 will generalize the algorithmic result of Section 3 to the Vertex-Weighted EOP. The time-complexity of the new algorithm is $O(mn\delta\log\delta)$. Section 5 will prove that the EOP is NP-hard on bipartite graphs and chordal graphs. Then, an $O(n\log n)$-time algorithm for the EOP on trees will be designed in Section 6. The conclusions and future research directions finally will be discussed in Section 7.

2. Related works and literature review

A directed graph D is called an *orientation* of an undirected graph G if G is the underlying undirected graph of D [33]. The EOP can be viewed as one variant of graph orientation problems by this definition. Chapter 61 of Schrijver's book provided an elegant survey on the topics related to graph orientation. The following variants of graph orientation were discussed [33].

2.1. Orientations with bounds on in- and out-degrees

This type of orientation is to derive orientations of an undirected graph G satisfying the bounds conditions on the in-degrees and/or out-degrees. The results can be proved quite directly from bipartite matching or flow theory. The major results are shown by the following theorems.

Theorem 1 [21]. *Let $G(V,E)$ be an undirected graph and let $l: V \to Z_+$. Then G has an orientation $D(V,A)$ with $\deg_A^{\text{in}}(v) \geqslant l(v)$, for each $v \in V$, iff each $U \subseteq V$ is incident with at least $l(U)$ edges.*

Theorem 2 [12]. *Let $G(V,E)$ be an undirected graph and let $l,u : V \to Z_+$ with $l \leqslant u$. Then G has an orientation $D(V,A)$ with $l(v) \leqslant \deg_A^{\text{in}}(v) \leqslant u(v)$, for each $v \in V$, iff each $U \subseteq V$ is incident with at least $l(U)$ edges and spans at most $u(U)$ edges.*

2.2. Two-edge-connectivity and strongly connected orientations

This type of orientation is to derive an orientation D of an undirected graph G such that D is strongly connected. The following is the well-known Robbins' Theorem.

Theorem 3 (Robbins' Theorem [32]). *An undirected graph G has a strongly connected orientation iff G is two-edge-connected.*

The following corollary can be obtained directly.

Corollary 1 ([32,33]). *Given a two-edge-connected graph G, a strongly connected orientation of G can be found in linear-time.*

Robbins' Theorem extends to the following results of Frank [11] and Bosech and Tindell [2] for mixed graphs.

Theorem 4 ([2,11,33]). *Let $G(V,E)$ be a graph in which part of the edges are oriented. Then the remainder of the edges can be oriented so as to obtain a* *strongly connected digraph iff G is two-edge-connected and there is no non-empty proper subset U of V such that all edges in $\delta(U)$ are oriented from U to $V \setminus U$, where $\delta(U)$ is the set of edges of G connected U and $V \setminus U$.*

Corollary 2 ([3,33]). *An orientation as described in Theorem* 4 *can be found in linear-time.*

The following theorem extends the Robbins' Theorem to the case where upper and lower bounds are prescribed on the in-degrees of the orientation, where $\kappa(G)$ denotes the number of the components of any graph G.

Theorem 5 [12]. *Let $G(V,E)$ be a two-edge-connceted undirected graph and let $l,u : V \to Z_+^V$ with $l \leqslant u$. Then G has a strongly connected orientation $D(V,A)$ satisfying $l(v) \leqslant \deg_A^{\mathrm{in}}(v) \leqslant u(v)$, for each $v \in V$, iff each $U \subseteq V : |E[U]| + \kappa(G-U) \leqslant u(U)$ and $|E[U]| + |\delta(U)| - \kappa(G-U) \geqslant l(U)$.*

2.3. Nash-Williams' Orientation Theorem

Nash-Williams' Theorem is an extension of the Robbins' Theorem, where $\lambda_D(s,t)$ denotes the maximum number of arc-disjoint $s-t$ paths in any orientation D of a graph G.

Theorem 6 (Nash-Williams' Orientation Theorem [30]). *An undirected graph $G(V,E)$ has an orientation $D(V,A)$ with $\lambda_D(s,t) \geqslant \left\lfloor \frac{1}{2}\lambda_G(s,t) \right\rfloor$, for all s, $t \in V$.*

2.4. k-arc-connected orientations of 2k-edge-connected graphs

Nash-Williams' Theorem directly implies the following theorem.

Theorem 7 ([30,33]). *An undirected graph G has a k-arc-connected orientation iff G is* 2*k-edge-connected.*

The following theorem can be established by the complexity of the Edmonds–Giles Problem.

Theorem 8 ([5,16,33]). *A k-arc-connected orientation of a* 2*k-edge-connected undirected graph G can be found in polynomial-time.*

The case where lower and upper bounds of the in-degrees of the vertices are prescribed in Theorem 7 is as follows.

Theorem 9 [10]. *Let $G(V,E)$ be a* 2*k-edge-connected undirected graph and let $l,u : V \to Z_+^V$ with $l \leqslant u$. Then G has a k-arc-connected orientation D with*

$l(v) \leqslant \deg_D^{\text{in}}(v) \leqslant u(v)$, for each $v \in V$, iff $|E[W]| + |\delta(P)| \geqslant \kappa(P) + \max\left\{\sum_{v\in W} l(v), \sum_{v\in W}(\deg_G(v) - u(v))\right\}$, for each subpartition P of V with non-empty classes, where $W := V \setminus \bigcup P$.

2.5. Other graph orientation issues

Edmonds' Disjoint Arborescences Theorem implies the following theorem.

Theorem 10 ([10,33]). *Let $G(V,E)$ be an undirected graph and $r \in V$. Then G has an orientation such that each non-empty subset U of $V \backslash \{r\}$ is entered by at least k arcs iff G contains k edge-disjoint spanning trees.*

Other important graph orientations include graph orientations satisfying parity and connectivity conditions [13–15], orientations preserving prescribed shortest paths [23], and applying submodularity to orientation problems [8,9]. Meanwhile, a strongly polynomial-time algorithm for finding a minimum-cost k-arc connected orientation was proposed in [17].

Previous research related to various graph orientation problems, as described above, seldom deal with edge-orientation problems on weighted graphs, i.e., the input graphs of most orientation problems so far have been unweighted. We will give an original research for edge-orientation problems on graphs with costs (weights) on both vertices and edges. Although the author in [36] recently pointed out that the Out-Degree-EOP arises in the design of restorable telecommunication networks and proposed an $O(m^2)$-time algorithm to solve the problem on general graphs, this paper will demonstrate a more intuitive and easier approach to design another $O(mn\delta \log \chi)$-time algorithm for solving the same problem, where $\delta = \max_{v\in V}\{\deg(v)\}$ and $\chi = \max\left\{1, \delta - \left\lceil \frac{m}{n} \right\rceil\right\}$. The time-complexity of our algorithm can be reduced to $O(mn)$ when δ is a constant. This greatly improves on the result of [36]. In general, our research is original and is the first study which extracts edge-orienting problems on weighted graphs from practical applications and then solves the extracted problems.

3. The Out-Degree-EOP on general graphs

Our idea for solving the Out-Degree-EOP has arisen from the following corresponding decision problem, where $\delta = \max_{v\in V}\{\deg(v)\}$.

The out-degree bounded edge-orientation problem (the Out-Degree Bounded EOP): Given a graph $G(V,E)$ and an integer $1 \leqslant k \leqslant \delta$, determine whether there exists an edge-orientation scheme A^* such that $\theta(A^*) = \max_{v\in V}\{\text{outdeg}(v)\} \leqslant k$.

Lemma 1. *For any graph $G(V,E)$, let $G_Q(V_Q,E_Q)$ be the subgraph induced by any vertex subset Q of V. Then, $\theta(G_Q) \leqslant \theta(G)$.*

Proof. Let A be an edge-orientation scheme of G with $\theta(A) = \theta(G)$. Then, outdeg(v) $\leqslant \theta(G)$, for each $v \in V$. Suppose that A_Q is the edge-orientation scheme via deleting all edges in $(E - E_Q)$. Verifying that A_Q is an edge-orientation scheme of G_Q and outdeg(v) $\leqslant \theta(G)$, for all $v \in V_Q$, within A_Q is easy. This implies that $\theta(G_Q) \leqslant \theta(G)$. □

Lemma 2. *Given a graph $G(V,E)$ and an integer $1 \leqslant k \leqslant \delta$, if the answer of the Out-Degree Bounded EOP is 'YES', then $k * n \geqslant m$. This means that the smallest possible value of k such that the Out-Degree Bounded EOP has a solution is $\lceil \frac{m}{n} \rceil$.*

Proof. Suppose that A is an edge-orientation scheme of G such that $\theta(A) \leqslant k$. Then, outdeg(v) $\leqslant k$, for each $v \in V$. We must have $m = \sum_{v\in V}\text{outdeg}(v) \leqslant \sum_{j=1}^{n} k = k * n$ according to the definition of edge-orientation schemes. □

Now, consider a graph $G(V,E)$ and an integer k such that $k * n \geqslant m$ for the Out-Degree Bounded EOP. Make a restriction that outdeg(v) $\leqslant k$, for each $v \in V$, after assigning the orientations of all edges. Therefore, a partial edge-orientation scheme H can be obtained through executing the following procedure.

Procedure Partial-EOS
Input: A graph $G(V,E)$ and an integer k such that $k * n \geqslant m$.
Output: A partial edge-orientation scheme H such that outdeg(v) $\leqslant k$, for each $v \in V$.
Method:

```
H = ∅; /* H is set to be empty, initially. */
for each vertex u
   outdeg(u) = 0;
   for each edge e = (u, v) incident with u
      if (outdeg(u) ≤ (k − 1)) and (the orientation of e is undetermined)
      {
         H = H ∪ {u → v};
         /* The orientation of e is from u to v. */
         outdeg(u) = outdeg(u) + 1;
      }
      else
         discard e; /* Let the orientation of e be undetermined */
      endif
   endfor
```

endfor
End Partial-EOS

An edge $e = (x, y)$ now is called an *undetermined edge* if its orientation has not been determined. Otherwise, e is called a *determined edge*. If the orientations of all edges in G currently have been assigned within H, then $\theta(G) = \theta(H) \leqslant k$ and the answer of the Out-Degree Bounded EOP is certainly 'YES'. Handling the situation where undetermined edges exist is the remaining task.

Lemma 3. *Suppose that* $e = (x, y)$ *is an undetermined edge when Procedure Partial-EOS has terminated. Then, the following properties hold.* (1) $\text{outdeg}(x) = \text{outdeg}(y) = k$, *and* (2) *there must exist a vertex* z *such that* $\text{outdeg}(z) \leqslant (k-1)$ *and all edges incident with it are determined edges.*

Proof. (1) This is trivial from the logic flow of the codes in Procedure Partial-EOS.

(2) It is trivial that $\text{outdeg}(v) \leqslant k$, for each $v \in V$, after executing Procedure Partial-EOS. The fact that $e = (x, y)$ is undetermined implies that if $\text{outdeg}(v) = k$, for each $v \in V$, then $m \geqslant k * n + 1$. This contradicts the assumption that $m \leqslant k * n$. Next, assume that z is a vertex such that $\text{outdeg}(z) = \lambda \leqslant (k-1)$. If there exists an undetermined edge (z, u), then the orientation of this edge must have been assigned as "$z \to u$" during the execution of Procedure Partial-EOS and $\text{outdeg}(z) \geqslant (\lambda + 1)$. Another contradiction occurs. □

Let $V_{=k} = \{v \in V \mid \text{outdeg}(v) = k\}$ and $V_{<k} = \{v \in V \mid \text{outdeg}(v) < k\}$. Lemma 3 implies that $V_{<k} \neq \emptyset$, $V_{=k} \neq \emptyset$ and $V = V_{=k} \cup V_{<k}$. For each $u \in V$, let $\Omega(u) = \{v \in V \mid \text{there exists a directed path from } v \text{ to } u \text{ within } H\}$, i.e., $\Omega(u)$ is the set of vertices which can reach u via directed edges within H, and $\Omega(Q)$ represents the set $\bigcup_{u \in Q} \Omega(u)$, for any $Q \subseteq V$. Similarly, let $\Gamma(u) = \{v \in V \mid \text{there exists a directed path from } u \text{ to } v \text{ within } H\}$, i.e., $\Gamma(u)$ is the set of vertices that can be reached from u via directed edges within H, and $\Gamma(Q)$ represents the set $\bigcup_{u \in Q} \Gamma(u)$, for any $Q \subseteq V$.

For any undetermined edge $e = (x, y)$, if a directed path P_e: $u_0 \to u_1 \to \cdots \to u_{q-1} \to u_q$ exists such that $u_0 \in \{x, y\}$, $\{u_1, \ldots, u_{q-1}\} \subseteq V_{=k}$, and $u_q \in V_{<k}$, then let H_e be the new edge-orientation scheme modified from H by performing the following two steps. After then, checking that $\theta(H_e) = \theta(H) = k$ is an easy task.

Step 1: Reverse the orientations of all edges in P_e.
Step 2: If u_0 is x then assign the orientation of e as "$x \to y$". Otherwise, the orientation of e is assigned as "$y \to x$".

Lemma 4. *Let H be any partial edge-orientation scheme of $G(V,E)$ by executing Procedure Partial-EOS and assume that $e_1,\ldots,e_t$, $t \geqslant 1$, are undetermined edges. The undetermined edges are now examined from e_1 to e_t sequentially. The answer of the Out-Degree Bounded EOP is 'Yes' iff a directed path P_{e_j} exists, for all $1 \leqslant j \leqslant t$.*

Proof. The definition of P_{e_j}'s directly implies that if a directed path P_{e_j} exists, for all $1 \leqslant j \leqslant t$, then the answer of the Out-Degree Bounded EOP is 'Yes'.

The case when P_{e_j} does not exist for some $e_j = (x_j, y_j)$ implies that $x_j \notin \Omega(V_{<k})$ and $y_j \notin \Omega(V_{<k})$, i.e., $(\Gamma(x_j) \cup \Gamma(y_j)) \subseteq V_{=k}$, by the definitions of $V_{=k}$ and $V_{<k}$. Let $G_{e_j}(V_{e_j}, E_{e_j})$ be the subgraph induced by the vertex-set $V_{e_j} = \{x_j, y_j\} \cup (\Gamma(x_j) \cup \Gamma(y_j))$. Then, $\text{outdeg}(v) = k$, for each $v \in V_{e_j}$, and $|E_{e_j}| \geqslant k * |V_{e_j}| + 1$. We must have $\theta(G_{e_j}) > k$. The inequality $\theta(G) > k$ can be derived from Lemma 1, i.e., the answer of the Out-Degree Bounded EOP is 'No'. □

The following algorithm can now be designed for correctly solving the Out-Degree-EOP on general graphs.

Algorithm Out-Degree-EOP
Input: A graph $G(V,E)$ with n vertices and m edges.
Output: An edge-orientation scheme A^* such that $\theta(A^*) = \max_{v \in V}\{\text{out-deg}(v)\}$ is minimized.
Method:

```
lb = ⌈m/n⌉; ub = max{deg(v)|v ∈ V}; A* = ∅;
while (lb ⩽ ub)
    mid = (lb + ub)/2;
    H = Partial-EOS(G, mid);
    /* Find a partial edge-orientation scheme H such that θ(H) ⩽ mid. */
    E_U = the set of undetermined edges;
    if (E_U ≠ ∅)
    {/* Assume that E_U = {e_1, ..., e_t} */
        for each edge e_j, 1 ⩽ j ⩽ t
            P_{e_j} = Find-Path();
            /* Find a directed path as stated above. */
            if (P_{e_j} is not empty)
            {
                Adjust(H);
                /* Adjust H by reversing the orientations of all edges in P_{e_j}
                and assign the orientation of e_j accordingly. */
                E_U = E_U − {e_j};
            }
```

```
            else /* P_{e_j} does not exist. */
               break; /* Terminate the for ... endfor loop. */
            endif
         endfor
      }
      endif
      if (E_U == ∅)
      {A* = H; ub = mid − 1;}
         else
            lb = mid + 1;
      endif
   endwhile
   return A*;
End Out-Degree-EOP
```

Theorem 11. *The Out-Degree-EOP on general graphs can be solved in* $O(mn\delta\log\chi)$*-time, where* $\chi = \max\left\{1, \delta - \left\lceil\frac{m}{n}\right\rceil\right\}$.

Proof. Determining the time-complexity of Algorithm Out-Degree-EOP is the task here. Assume that $\{e_1, \ldots, e_t\}$ is the set of undetermined edges after executing Procedure Partial-EOS. Let $e_j = (u_j, v_j)$, $1 \leqslant j \leqslant t$. The discussions so far imply that $u_j, v_j \in V_{=k}$, $1 \leqslant j \leqslant t$. Thus, it can be derived that $t \leqslant \delta * |V_{=k}| \leqslant \delta * n$.

The time-complexity of the algorithm can be summarized as follows:

(1) If $m = 1$, the number of iterations of the **while** ... **endwhile** loop is 1. Otherwise, it is at most $\log\left(\delta - \left\lceil\frac{m}{n}\right\rceil\right)$.
(2) The number of iterations of the **for** ... **endfor** loop is at most $\delta * n$.
(3) The time-complexity for Procedure Partial-EOS is clear $O(m)$.
(4) Procedure Find-Path and Procedure Adjust can be done in $O(m)$-time, respectively.

The time-complexity of Algorithm Out-Degree-EOP is $O(mn\delta\log\chi)$-time via summarizing the above reasoning, where $\chi = \max\left\{1, \delta - \left\lceil\frac{m}{n}\right\rceil\right\}$. □

If the degrees of all vertices are bounded by a constant, i.e., δ is a constant, then the time-complexity of Algorithm Out-Degree-EOP can be clearly reduced to $O(mn)$.

Theorem 12. *The Out-Degree-EOP on general graphs with bounded degrees can be solved in* $O(mn)$*-time.*

4. The Vertex-Weighted EOP on general graphs

This section will generalize the algorithmic result of the Out-Degree-EOP to the Vertex-Weighted EOP. Suppose that $G(V, E, C)$ is an input instance of the Vertex-Weighted EOP and assume that $\beta = \max_{v\in V}\{C(v)\}$. For any edge-orientation scheme A, define $\Delta(v) = C(v) + \text{outdeg}(v)$, for each $v \in V$. The following lemma can be established from the definition of the Vertex-Weighted EOP directly.

Lemma 5. *Let A be any edge-orientation scheme of the Vertex-Weighted EOP on the graph $G(V, E, C)$. Then, $\pi(A) = \max_{v\in V}\{\Delta(v)\} \geqslant \beta$.*

Lemma 5 implies that we can start from the following corresponding decision problem, where δ has been defined as $\max_{v\in V}\{\deg(v)\}$ in the previous sections.

The bounded vertex-weighted edge-orientation problem (the Bounded Vertex-Weighted EOP): Given a graph $G(V, E, C)$ and an integer $0 \leqslant k \leqslant \delta$, determine whether there exists an edge-orientation scheme A^* such that $\pi(A^*) \leqslant (\beta + k)$.

Lemma 6. *For any vertex-weighted graph $G(V, E, C)$, let$G_Q(V_Q, E_Q)$ be the subgraph induced by any vertex subset Q of V. Then, $\pi(G_Q) \leqslant \pi(G)$.*

Proof. Suppose that A is any edge-orientation scheme of G such that $\pi(A) = \pi(G)$. Then, $\Delta(v) \leqslant \pi(G)$, for each $v \in V$. Let A_Q be the edge-orientation scheme via deleting the edges in $(E - E_Q)$. Ascertaining that A_Q is a feasible edge-orientation scheme of G_Q and $\Delta(v) \leqslant \pi(G)$, for each $v \in V_Q$, within A_Q can be easily done. This implies that $\pi(G_Q) \leqslant \pi(G)$. $\square$

Given any non-negative integer k, define $\text{quota}(v) = (\beta + k) - C(v)$, for each $v \in V$.

Lemma 7. *Given a vertex-weighted graph $G(V, E, C)$ and a non-negative integer k, if the answer of the Bounded Vertex-Weighted EOP is 'YES', then $\sum_{v\in V}\text{quota}(v) \geqslant m$.*

Proof. Suppose that A is an edge-orientation scheme such that $\pi(A) \leqslant (\beta + k)$. This implies that the following inequalities hold, for each $v \in V$:

$$\begin{aligned}\Delta(v) \leqslant (\beta + k) \quad &\Rightarrow \quad (C(v) + \text{outdeg}(v)) \leqslant (\beta + k)\\ &\Rightarrow \quad \text{outdeg}(v) \leqslant ((\beta + k) - C(v))\\ &\Rightarrow \quad \text{outdeg}(v) \leqslant \text{quota}(v).\end{aligned}$$

We must have $m = \sum_{v \in V} \text{outdeg}(v) \leqslant \sum_{v \in V} \text{quota}(v)$ according to the definition of edge-orientation schemes. $\square$

Consider a graph $G(V, E, C)$ and a non-negative integer k such that $\sum_{v \in V} \text{quota}(v) \geqslant m$ for the Bounded Vertex-Weighted EOP. Make a restriction that $\Delta(v) \leqslant (\beta + k)$, for each $v \in V$, after assigning the orientations of all edges. A partial edge-orientation scheme R can be obtained by performing the following procedure:

Procedure Partial-EOS-2
Input: A graph $G(V, E, C)$ and an integer $k \geqslant 0$ such that $\sum_{v \in V} \text{quota}(v) \geqslant m$.
Output: A partial edge-orientation scheme R such that $\Delta(v) \leqslant (\beta + k)$, for each $v \in V$.
Method:

```
R = ∅; /* R is set to be empty, initially. */
for each vertex u
{
    outdeg(u) = 0; Δ(v) = C(v);
    for each edge e = (u, v) incident with u
        if Δ(v) ⩽ ((β + k) − 1) and (the orientation of e is undetermined)
        {
            R = R ∪ {u → v}; /* The orientation of e is from u to v. */
            outdeg(u) = outdeg(u) + 1; Δ(v) = Δ(v) + 1;
        }
        else
            discard e; /* Let the orientation of e be undetermined */
        endif
    endfor
}
endfor
```

End Partial-EOS-2

If all edges in G are already determined, then $\pi(G) = \pi(R) \leqslant (\beta + k)$ and the answer of the Bounded Vertex-Weighted EOP is 'YES'. The remaining task is to assign the orientations of all undetermined edges.

Lemma 8. *Suppose that $e = (x, y)$ is an undetermined edge when ProcedurePartial-EOS-2 terminated. The following properties must hold.* (1) $(\Delta(x) + 1) > (\beta + k)$ *and* $(\Delta(y) + 1) > (\beta + k)$. (2) *There must exist a vertex z such that $\Delta(z) \leqslant ((\beta + k) - 1)$ and all edges incident with it are determined edges.*

Proof. (1) The statement is trivially true based upon the logic of the codes in Procedure Partial-EOS-2.

(2) It is trivial that $\Delta(v) \leqslant (\beta + k)$, for each $v \in V$, after executing Procedure Partial-EOS-2. Since $e = (x, y)$ is undetermined, if $\Delta(v) = C(v) + \text{out-deg}(v) > (\beta + k)$, for each $v \in V$, then $\text{outdeg}(v) > ((\beta + k) - C(v)) \geqslant \text{quota}(v)$. This implies that $m = \sum_{v \in V} \text{outdeg}(v) > \sum_{v \in V} \text{quota}(v) + 1$. This is contradictory to the assumption that $m \leqslant \sum_{v \in V} \text{quota}(v)$. Next, assume that z is a vertex such that $\Delta(z) = \lambda \leqslant ((\beta + k) - 1)$. If there exists an undetermined edge (z, u), then the orientation of this edge must have been assigned as "$z \rightarrow u$" during the execution of Procedure Partial-EOS-2 and $\text{outdeg}(z) \geqslant \lambda + 1$. Another contradiction occurs. □

Let $V_1 = \{v \in V | (\Delta(v) + 1) > (\beta + k)\}$ and $V_2 = \{v \in V | (\Delta(v) + 1) \leqslant (\beta + k)\}$. Lemma 8 implies that $V_1 \neq \emptyset$, $V_2 \neq \emptyset$ and $V = V_1 \cup V_2$. For any undetermined edge $e = (x, y)$, if a directed path P_e: $u_0 \rightarrow u_1 \rightarrow \cdots \rightarrow u_{q-1} \rightarrow u_q$ exists such that $u_0 \in \{x, y\}$, $\{u_1, \ldots, u_{q-1}\} \subseteq V_1$, and $u_q \in V_2$, then let H_e be the new edge-orientation scheme modified from H by performing the following two steps. After that, the property that $\pi(H_e) = \pi(H) = (\beta + k)$ can be easily verified.

Step 1: Reserve the orientations of all edges in P_e.
Step 2: Assign the orientation of e as "$x \rightarrow y$" if u_0 is x. Otherwise, the orientation of e is assigned as "$y \rightarrow x$".

Lemma 9. *Let H be any partial edge-orientation scheme of a graph $G(V, E, C)$ by executing Procedure Partial-EOS-2. Assume that $e_1, \ldots, e_t$, $t \geqslant 1$, are all undetermined edges and they are examined from e_1 to e_t one by one. The answer of the Bounded Vertex-Weighted EOP is 'Yes' iff a directed path P_{e_j} exists, for all $1 \leqslant j \leqslant t$.*

Proof. The definition of P_{e_j}'s directly implies that if a directed path P_{e_j} exists, for all $1 \leqslant j \leqslant t$, then the answer of the Bounded Vertex-Weighted EOP is 'Yes'.

The notations $\Omega(Q)$ and $\Gamma(Q)$ have been defined in Section 2 for each vertex $Q \subseteq V$. The case when P_{e_j} does not exist for some $e_j = (x_j, y_j)$ implies that $x_j \notin \Omega(V_2)$ and $y_j \notin \Omega(V_2)$, i.e., $(\Gamma(x_j) \cup \Gamma(y_j)) \subseteq V_1$ by the definitions of V_1 and V_2. Let $G_{e_j}(V_{e_j}, E_{e_j})$ be the subgraph induced by the vertex-set $V_{e_j} = \{x_j, y_j\} \cup (\Gamma(x_j) \cup \Gamma(y_j))$. Then, $(\Delta(v) + 1) > (\beta + k)$, for each $v \in V_{e_j}$, and $|E_{e_j}| \geqslant \left(\sum_{v \in V_{e_j}} \text{quota}(v) + 1\right)$. We must have $\pi(G_{e_j}) > (\beta + k)$. We can derive that $\pi(G) > (\beta + k)$ by Lemma 6, i.e., the answer of the Bounded Vertex-Weighted EOP with G and k is 'No'. □

The following algorithm can now be designed to correctly solve the Vertex-Weighted EOP on general graphs.

Algorithm Vertex-Weighted-EOP
Input: A graph $G(V, E, C)$ with n vertices and m edges.
Output: An edge-orientation scheme A^* such that $\pi(A^*) = \max_{x \in V}\{C(v) + \text{outdeg}(x)\}$ is minimized.
Method:

```
lb = 0; ub = max_{v∈V}{deg(v)};
A* = ∅;
while (lb ⩽ ub)
    mid = (lb + ub)/2;
    H = Partial-EOS-2(G, mid);
    /*
    Find a partial edge-orientation scheme H such that π(H) ⩽ (β +
    mid).
    */
    E_U = the set of undetermined edges;
    if (E_U ≠ ∅)
    {/* Assume that E_U = {e_1, ..., e_t} */
        for each edge e_j, 1 ⩽ j ⩽ t
            P_{e_j} = Find-Path-2();
            /* Find a directed path as stated above. */
            if (P_{e_j} is not empty)
            {
                Adjust-2(H);
                /* Adjust H by reversing the orientations of all edges in P_{e_j}
                and assign the orientation of e_j accordingly. */
                E_U = E_U − {e_j};
            }
            else /* P_{e_j} does not exist. */
               break; /* Terminate the for ... endfor loop. */
            endif
        endfor
    }
    endif
    if (E_U == ∅)
    {A* = H; ub = mid − 1;}
    else
        lb = mid + 1;
    endif
endwhile
return A*;
```

End Vertex-Weighted-EOP

Theorem 13. *The Vertex-Weighted EOP on general graphs can be solved in* $O(mn\delta\log\delta)$*-time.*

Proof. Assume that $\{e_1,\ldots,e_t\}$ is the set of undetermined edges after executing Procedure Partial-EOS-2. Let $e_j=(u_j,v_j)$, $1\leqslant j\leqslant t$. From the discussions so far, we know that u_j, $v_j\in V_1$, $1\leqslant j\leqslant t$. Thus, we can derive that $t\leqslant\delta*|V_1|\leqslant\delta*n$.

The time-complexity of Algorithm Vertex-Weighted EOP is $O(mn\delta\log\delta)$ by the following reasoning:

(1) The number of iterations of the **while** ... **endwhile** is at most $\log\delta$.
(2) The number of iterations of the **for** ... **endfor** loop is at most $\delta*n$.
(3) The time-complexity for Procedure Partial-EOS-2 is clear $O(m)$.
(4) Both Procedure Find-Path-2 and Procedure Adjust-2 can be done in $O(m)$-time. □

The following theorem holds by the same reasoning of Theorem 12.

Theorem 14. *The Vertex-Weighted EOP on general graphs with bounded degrees can be solved in* $O(mn)$*-time.*

5. NP-hardness of the EOP on bipartite graphs and chordal graphs

This section will present very different algorithmic results with respect to previous sections–the EOP is NP-hard on bipartite graphs and chordal graphs. A graph $G(V,E)$ is called a *bipartite graph* [20] if V consists of two disjoint sets X and Y such that $(u,v)\in E$ implies that either ($u\in X$ and $v\in Y$) or ($u\in Y$ and $v\in X$).

Another special version of the EOP and a corresponding decision problem are proposed.

The edge-weighted only edge-orientation problem (the Edge-Weighted Only EOP): Let $G(V,E,W)$ be a graph in which each edge $e=(u,v)$ is associated with two positive weights, $W(u\rightarrow v)$ and $W(v\rightarrow u)$. Denote $\sigma(A)$ as $\max_{x\in V}\{\sum_{x\rightarrow z}W(x\rightarrow z)\}$, for each edge-orientation scheme A. The problem is to identify an edge-orientation scheme A^* such that $\sigma(A^*)$ is minimized. Let $\sigma(G)=\min\{\sigma(A)\,|\,A$ is an edge-orientation scheme of $G\}$ hereafter.

The edge-weighted bounded edge-orientation problem (the Edge-Weighted Bounded EOP): Given a positive-edge-weighted graph $G(V,E,W)$ and a positive constant k, determine whether there exists an edge-orientation scheme A^* such that $\sigma(A^*)=\max_{x\in V}\{\sum_{x\rightarrow z}W(x\rightarrow z)\}\leqslant k$.

The following NP-complete problem is used for reduction [18].

The monotone three satisfiability problem (the M3SAT problem): Given a set C of Boolean clauses in the conjunctive normal form in which each clause contains either only positive literals, say u_i's, or only negative literals, say $\overline{u_i}$'s, and each clause contains exactly three literals, the task is to determine whether the given Boolean formula is satisfiable or not.

Lemma 10. *The Edge-Weighted Bounded EOP is NP-complete on bipartite graphs.*

Proof. It is clear that the Edge-Weighted Bounded EOP belongs to the class of NP problems. Suppose that there is an instance of the M3SAT problem with the variable-set $U = \{u_1, \ldots, u_h\}$ and the clause-set $C = \{c_1, \ldots, c_\alpha, c_{\alpha+1}, \ldots, c_\beta\}$, where $\{c_1, \ldots, c_\alpha\}$ is the set of clauses containing only positive literals and $\{c_{\alpha+1}, \ldots, c_\beta\}$ is the set of clauses containing only negative literals. Let $\overline{U} = \{\overline{u_1}, \ldots, \overline{u_h}\}$. A positive-edge-weighted graph $G(X \cup Y, E, W)$ can be constructed as follows, for any constant $k > 0$.

$X = U \cup \{c_{\alpha+1}, \ldots, c_\beta\}$ and $Y = \overline{U} \cup \{c_1, \ldots, c_\alpha\}$;
$E = \{(c_i, u_j) | c_i \text{ contains } u_j, 1 \leqslant i \leqslant \alpha \text{ and } 1 \leqslant j \leqslant h\} \cup \{(u_i, \overline{u_i}) | 1 \leqslant i \leqslant h\} \cup \{(c_i, \overline{u_j}) | c_i \text{ contains } \overline{u_j}, \alpha + 1 \leqslant i \leqslant \beta \text{ and } 1 \leqslant j \leqslant h\}$;
$W(u_i \to \overline{u_i}) = W(\overline{u_i} \to u_i) = k$, for each edge $(u_i, \overline{u_i})$, $1 \leqslant i \leqslant h$;
$W(c_i \to u_j) = k/2$ and $W(u_j \to c_i) = k/\deg(u_j)$, for each edge (c_i, u_j), $1 \leqslant i \leqslant \alpha$ and $1 \leqslant j \leqslant h$;
$W(c_i \to \overline{u_j}) = k/2$ and $W(\overline{u_j} \to c_i) = k/\deg(\overline{u_j})$, for each edges $(c_i, \overline{u_j})$, $\alpha + 1 \leqslant i \leqslant \beta$ and $1 \leqslant j \leqslant h$.

Each edge in G connects a vertex in X and another vertex in Y. Therefore, G is a bipartite graph. Meanwhile, $\deg(c_i) = 3$, for all $1 \leqslant i \leqslant \beta$. The remaining task is to show that there exists an edge-orientation scheme H such that $\max_{v \in V}\left\{\sum_{x \to z} W(x \to z)\right\} \leqslant k$ in G iff the Boolean formula $c_1 \bullet \cdots \bullet c_\alpha \bullet c_{\alpha+1} \bullet \cdots \bullet c_\beta$ is satisfiable.

Assume that there is an assignment satisfying the input Boolean formula. Let $u_{z_1} = \cdots = u_{z_p} = \text{TRUE}$ and $u_{w_1} = \cdots = u_{w_q} = \text{FALSE}$, where $p + q = h$. Then, $z_i \neq w_j$, for all i and j. An edge-orientation scheme H can be obtained via executing the following code segment.

$U^{\mathrm{T}} = \{u_{z_1}, \ldots, u_{z_p}\}$; $U^{\mathrm{F}} = \{\overline{u_{w_1}}, \ldots, \overline{u_{w_q}}\}$;
for each $e = (\overline{u_{z_i}}, u_{z_i})$, $1 \leqslant i \leqslant p$
  assign the orientation of e from $\overline{u_{z_i}}$ to u_{z_i};
endfor
for each $e = (u_{w_j}, \overline{u_{w_j}})$, $1 \leqslant j \leqslant q$

```
    assign the orientation of e from u_{w_j} to \overline{u_{w_j}};
endfor
for each e = (c_j, u_i), 1 ≤ j ≤ α and u_i ∈ U
    if u_i ∈ U^T
        assign the orientation of e from u_i to c_j;
    else
        assign the orientation of e from c_j to u_i;
    endif
endfor
for each e = (c_j, \overline{u_i}), α + 1 ≤ j ≤ β and \overline{u_i} ∈ \overline{U}
    if \overline{u_i} ∈ U^F
        assign the orientation of e from \overline{u_i} to c_j;
    else
        assign the orientation of e from c_j to \overline{u_i};
    endif
endfor
```

Verifying that $\max_{x\in V}\left\{\sum_{x\to z}W(x\to z)\right\}\leqslant k$ can be achieved based upon the following reasoning:

1. Each clause c_j, $1\leqslant j\leqslant\beta$, must contain at least one true literal in this assignment. This implies that $\mathrm{indeg}(c_j)\geqslant 1$ within H. We can further claim that $\sum_{c_j\to u,u\in U\cup\overline{U}}W(c_j\to u)\leqslant \mathrm{outdeg}(c_j)*(k/2)\leqslant 2*(k/2)=k$ since $\deg(c_j)=3$.
2. The above codes guarantee that $\mathrm{outdeg}(v)=0$, for all $v\in(U\cup\overline{U})-(U^{\mathrm{T}}\cup U^{\mathrm{F}})$.
3. $\sum_{v\to c_j}W(v\to c_j)\leqslant \mathrm{outdeg}(v)*(k/\deg(v))\leqslant\deg(v)*(k/\deg(v))=k$, for each $v\in(U^{\mathrm{T}}\cup U^{\mathrm{F}})$.

Next, if there exists an edge-orientation scheme H such that $\max_{x\in V}\left\{\sum_{x\to z}W(x\to z)\right\}\leqslant k$, then either $\mathrm{outdeg}(u_i)=0$ or $\mathrm{outdeg}(\overline{u_i})=0$ since $W(u_i\to\overline{u_i})=W(\overline{u_i}\to u_i)=k$, for each edge $(u_i,\overline{u_i})$. In addition, for each clause c containing any literal v, $W(c,v)=\frac{k}{2}$ implies that $\mathrm{outdeg}(c)\leqslant 2$, i.e., there must exist a literal y in c such that the orientation of the edge (c,y) is from y to c. Let S denote the set $\{v\in(U\cup\overline{U})|\mathrm{outdeg}(v)>0\}$. The assignment in which the literals corresponding to S are assigned to be TRUE certainly satisfies the input Boolean formula. □

Theorem 15. *The EOP is NP-hard on bipartite graphs.*

The class of chordal graphs is the second class of graphs considered in this section. An edge is a *chord* of any cycle in any graph G if it connects two non-consecutive vertices of this cycle. A graph is called a *chordal graph* [19] if each

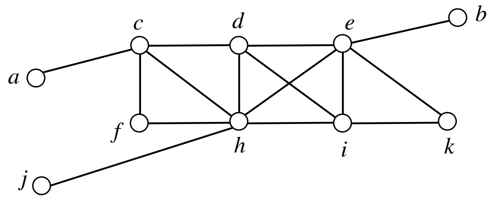

Fig. 3. A chordal graph $G(V, E)$.

cycle with a length greater than three has a chord. The graph depicted in Fig. 3 is an example of a chordal graph.

Lemma 11. *The Edge-Weighted Bounded EOP is NP-complete on chordal graphs.*

Proof. Suppose that there is an instance of the M3SAT problem with the variable-set $U = \{u_1, \ldots, u_h\}$ and the clause-set $C = \{c_1, \ldots, c_\alpha, c_{\alpha+1}, \ldots, c_\beta\}$, where $\{c_1, \ldots, c_\alpha\}$ is the set of clauses containing only positive literals and $\{c_{\alpha+1}, \ldots, c_\beta\}$ is the set of clauses containing only negative literals. Let $\overline{U} = \{\overline{u_1}, \ldots, \overline{u_h}\}$. A positive-edge-weighted graph $G(V, E, W)$ can be constructed as follows for any positive constant k.

$V = U \cup \overline{U} \cup C$;
$E = \{(c_j, u_i), (c_j, \overline{u_i}) | c_j \text{ contains } u_i, 1 \leqslant j \leqslant \alpha \text{ and } 1 \leqslant i \leqslant h\} \cup \{(c_j, u_i), (c_j, \overline{u_i}) | c_j \text{ contains } \overline{u_i}, \alpha + 1 \leqslant j \leqslant \beta \text{ and } 1 \leqslant i \leqslant h\} \cup \{(u_i, \overline{u_i}) | 1 \leqslant i \leqslant h\} \cup \{(c_s, c_t) | 1 \leqslant s \neq t \leqslant \beta\}$;
$W(u_i \to \overline{u_i}) = W(\overline{u_i} \to u_i) = k$, for each edge $(u_i, \overline{u_i})$, $1 \leqslant i \leqslant h$;
$W(c_j \to u_i) = k/(\beta - j + 5)$ and $W(u_i \to c_j) = k/\deg(u_i)$, for each edge (c_j, u_i), $1 \leqslant j \leqslant \beta$ and $1 \leqslant i \leqslant h$;
$W(c_j \to \overline{u_i}) = k/(\beta - j + 5)$ and $W(\overline{u_i} \to c_j) = k/\deg(\overline{u_i})$, for each edge $(c_j, \overline{u_i})$, $1 \leqslant j \leqslant \beta$ and $1 \leqslant i \leqslant h$;
$W(c_j \to c_s) = k/(\beta - j + 5)$, for all $1 \leqslant j \leqslant \beta$ and $j < s$;
$W(c_j \to c_s) = \infty$, for all $1 \leqslant j \leqslant \beta$ and $j > s$;

It is clear that $|\text{Neighbors}(c_j) \cap (U \cup \overline{U})| = 6$, for all $1 \leqslant j \leqslant \beta$. To show that G is chordal, assume that a cycle Φ with a length greater than three exists and the cycle is $v_1 — v_2 — \cdots — v_p — v_1$, $p \geqslant 4$, i.e., its vertex-set is $\{v_1, v_2, \ldots, v_p\}$. The following cases should be handled based upon the construction rules of G.

Case 1. $u_i \notin \Phi$ and $\overline{u_i} \notin \Phi$, for some $1 \leqslant i \leqslant h$.
All vertices of Φ must be clause vertices in this case and a chord must exist since they form a clique of G.

Case 2. $u_i \in \Phi$ and $\overline{u_i} \notin \Phi$, for some $1 \leqslant i \leqslant h$.
Φ must be in the form $\cdots$—c_s—u_i—$\overline{c_t}$—$\cdots$ in this situation and the edge $(c_s, \overline{c_t})$ is a chord of Φ.

Case 3. $u_i \notin \Phi$ and $\overline{u_i} \in \Phi$, for some $1 \leqslant i \leqslant h$.
This case is just a symmetrical case to Case 2.

Case 4. $u_i \in \Phi$ and $\overline{u_i} \in \Phi$, for some $1 \leqslant i \leqslant h$.
Φ must be in the form $\cdots$—c_s—u_i—$\overline{u_i}$—$\overline{c_t}$—$\cdots$ in this case and the edge $(c_s, \overline{u_i})$ is a chord of Φ.

The remaining task is to show that there exists an edge-orientation scheme H such that $\max_{x\in V}\left\{\sum_{x\to z} W(x \to z)\right\} \leqslant k$ in G iff the Boolean formula $c_1 \bullet \cdots \bullet c_\alpha \bullet c_{\alpha+1} \bullet \cdots \bullet c_\beta$ is satisfiable.

Assume that there is an assignment satisfying the input Boolean formula. Let $u_{z_1} = \cdots = u_{z_p} = \text{TRUE}$ and $u_{w_1} = \cdots = u_{w_q} = \text{FALSE}$, where $p + q = h$. Then, $z_i \neq w_j$, for all i and j. An edge-orientation scheme H can be obtained via executing the following code segment:

```
U^T = {u_{z_1}, ..., u_{z_p}}; U^F = {\overline{u_{w_1}}, ..., \overline{u_{w_q}}};
for each e = (\overline{u_{z_i}}, u_{z_i}), 1 ⩽ i ⩽ p
    assign the orientation of e from \overline{u_{z_i}} to u_{z_i};
endfor
for each e = (u_{w_j}, \overline{u_{w_j}}), 1 ⩽ j ⩽ q
    assign the orientation of e from u_{w_j} to \overline{u_{w_j}};
endfor
for each e = (c_j, u_i), 1 ⩽ j ⩽ α and u_i ∈ U
    if u_i ∈ U^T
        assign the orientation of e from u_i to c_j;
    else
        assign the orientation of e from c_j to u_i;
    endif
endfor
for each e = (c_j, \overline{u_i}), α + 1 ⩽ j ⩽ β and \overline{u_i} ∈ \overline{U}
    if \overline{u_i} ∈ U^F
        assign the orientation of e from \overline{u_i} to c_j;
    else
        assign the orientation of e from c_j to \overline{u_i};
    endif
endfor
for each c_j, 1 ⩽ j ⩽ β
    for each undetermined edge (c_j, v)
        if v ∈ (U ∪ \overline{U})
            assign the orientation of e from c_j to v;
        else /* v ∈ C and assume that v = c_t. */
```

```
        if j < t
          assign the orientation of e from c_j to c_t;
        endif
      endif
    endfor
  endfor
```

Verifying that $\max_{x\in V}\left\{\sum_{x\to z}W(x\to z)\right\}\leqslant k$ can be achieved based upon the following reasoning:

1. Each clause c_j, $1\leqslant j\leqslant\beta$, must contain at least one true literal in this assignment. This implies that $\text{indeg}(c_j)\geqslant 1$ within H. We can further claim that $\sum_{c_j\to v,v\in(C\cup U\cup\overline{U})}W(c_j\to v)\leqslant \text{outdeg}(c_j)*k/(\beta-j+5)\leqslant(\beta-j+5)*(k/(\beta-j+5))=k$ based upon the rules of assigning weights of all edges since $|\text{Neighbors}(c_j)\cap(U\cup\overline{U})|=6$.
2. The above codes guarantee that $\text{outdeg}(v)=0$, for each $v\in(U\cup\overline{U})-(U^{\mathrm{T}}\cup U^{\mathrm{F}})$.
3. $\sum_{v\to c_j}W(v\to c_j)\leqslant \text{outdeg}(v)*(k/\deg(v))\leqslant\deg(v)*(k/\deg(v))=k,\quad$ for each $v\in(U^{\mathrm{T}}\cup U^{\mathrm{F}})$.

Next, if there exists an edge-orientation scheme H such that $\max_{x\in V}\left\{\sum_{x\to z}W(x\to z)\right\}\leqslant k$, then either $\text{outdeg}(u_i)=0$ or $\text{outdeg}(\overline{u_i})=0$, for each edge $(u_i,\overline{u_i})$, since $W(u_i\to\overline{u_i})=W(\overline{u_i}\to u_i)=k$. Meanwhile, for each clause c_j, $1\leqslant j\leqslant\beta$, our rules of assigning weights of edges imply that $\text{outdeg}(c_j)\leqslant(\beta-j+5)$, i.e., there must exist a literal y in c_j such that the orientation of the edge (c_j,y) is from y to c_j. Let S denote the set $\{v\in(U\cup\overline{U})|\text{outdeg}(v)>0\}$. Set all literals corresponding to S as TRUE. This assignment consequently satisfies the Boolean formula. □

Theorem 16. *The EOP is NP-hard on chordal graphs.*

6. An O($n\log n$)-time algorithm for the EOP on trees

This section will propose an O($n\log n$)-time algorithm to solve the EOP on trees. Given a tree T and any vertex r, the tree will be denoted by $T(r)$ hereafter. A general tree $T(r)$ and its subtrees are shown in Fig. 4.

Considering each subtree $T(x_j)$, the orientation of the edge (r,x_j) can be either from r to x_j or from x_j to r. If the orientation is from r to x_j, then $W(r\to x_j)$ will be added to the total cost of r and an optimal edge-orientation scheme of $T(x_j)$ can be solved recursively and independently. We denote that $\mu(T(x_j,r\to x_j))=\min\{\mu(A)\,|\,A$ is an edge-orientation scheme of $T(x_j)$ in which

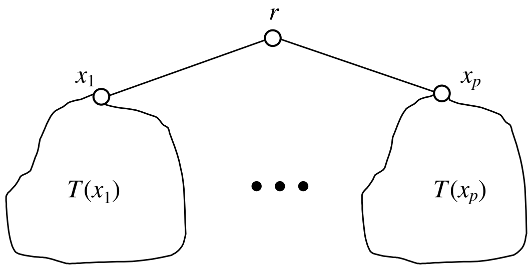

Fig. 4. The subtrees of $T(r)$.

the orientation of the edge (r, x_j) is from r to $x_j\}$. Otherwise, the orientation is from x_j to r. $W(x_j \to r)$ will be added to the total cost of x_j by the definition of the EOP. Replace $C(x_j)$ by $C(x_j) + W(x_j \to r)$ and then recursively find an optimal edge-orientation scheme of $T(x_j)$. We denote that $\mu(T(x_j, x_j \to r)) = \min\{\mu(A) \mid A$ is an edge-orientation scheme of $T(x_j)$ in which the orientation of the edge (r, x_j) is from x_j to $r\}$.

If the subtree $T(x_j)$ only consists of x_j, then $\mu(T(x_j, r \to x_j)) = C(x_j)$ and $\mu(T(x_j, x_j \to r)) = C(x_j) + W(x_j \to r)$. Otherwise, a feasible edge-orientation scheme H of $T(r)$ can be obtained via assigning orientations of all edges (r, x_j) from x_j to r and the following formula can be derived.

$$\mu(T(r)) \leqslant \mu(H) = \max\left\{C(r), \max_{1 \leqslant j \leqslant p}\{\mu(T(x_j, x_j \to r))\}\right\}. \tag{6.1}$$

The task required here is to determine a subset $Q = \{(r, x_{q_1}), \ldots, (r, x_{q_\alpha})\}$ (may be empty) of $\{(r, x_1), \ldots, (r, x_p)\}$ such that reversing the orientations of all edges in Q can obtain $\mu(T(r))$. The following code segment is designed for identifying such a set:

$Q = \emptyset$; $K = \{T(x_1), \ldots, T(x_p)\}$;
current $\mu = \max\{C(r), \max_{1 \leqslant j \leqslant p}\{\mu(T(x_j, x_j \to r))\}\}$;
sort $T(x_j)$ into non-decreasing order using $\mu(T(x_j, x_j \to r))$ as keys, $1 \leqslant j \leqslant p$;
while $(Q \neq K)$
  max $\mu = \max\{\mu(T(x_z, x_z \to r)) \mid T(x_z) \in K - Q\}$;
  $H = \{T(x_s) \in K - Q \mid \mu(T(x_s, x_s \to r))$ is equal to max $\mu\}$;
  new $\mu = \max\left\{C(r) + \sum_{T(x_s) \in H} W(r \to x_s),\right.$
    $\left.\max\{\mu(T(x_z, x_z \to r)) \mid T(x_z) \in K - (H \cup Q)\}\right\}$;
  new$C_r = C(r) + \sum_{T(x_s) \in H} W(r \to x_s)$;
  do case
  case new $\mu <$ current μ
    {
      $C(r) =$ newC_r;

```
            Q = Q ∪ H; current μ = new μ;
        }
    case new μ = current μ
        if new C_r ⩽ C(r)
        {
            C(r) = new C_r;
            Q = Q ∪ H; current μ = new μ;
        }
    case new μ > current μ
        exit(); /* Terminate the while ...endwhile loop. */
    endcase
endwhile
```

Lemma 12. *Let Q_s denote the set derived after the sth iteration of the while loop in the above code segment. Then,* $\max\left\{C(r) + \sum_{T(x_z)\in Q_s} W(r \to x_z), \max\{\mu(T(x_z, x_z \to r)) | T(x_z) \in K - Q_s\}\right\} \leqslant \max\left\{C(r) + \sum_{T(x_z)\in Q_{s-1}} W(r \to x_z), \max\{\mu(T(x_z, x_z \to r)) | T(x_z) \in K - Q_{s-1}\}\right\}$.

Lemma 13. *Let Q be the set obtained after terminating the above code segment. Then,* $\max\left\{C(r) + \sum_{T(x_z)\in Q} W(r \to x_z), \max\{\mu(T(x_z, x_z \to r)) | T(x_z) \in K - Q\}\right\} \leqslant \max\left\{C(r) + \sum_{T(x_z)\in H} W(r \to x_z), \max\{\mu(T(x_z, x_z \to r)) | T(x_z) \in K - H\}\right\}$, *for each subset H of* $\{T(x_1), \ldots, T(x_p)\}$.

Proof. Suppose that H is any optimal edge-orientation scheme of T. Let Q_s denote the set derived after the sth iteration of the while loop in the above code segment. We give the following reasoning step by step:

(1) If Q_1 is not empty, then $H \in \{R$ is an edge-orientation scheme of $T | r \to x_z$, for all $T(x_z) \in Q_1.\}$
(2) If Q_2 is not empty, then $H \in \{R$ is an edge-orientation scheme of $T | r \to x_z$, for all $T(x_z) \in Q_2.\}$
(3) ...

The above reasoning just implies that $H \in \{R$ is an edge-orientation scheme of $T | r \to x_z$, for each $T(x_z) \in Q\}$ and $\max\left\{C(r) + \sum_{T(x_z)\in Q} W(r \to x_z), \max\{\mu(T(x_z, x_z \to r)) | T(x_z) \in K - Q\}\right\} \leqslant \max\left\{C(r) + \sum_{T(x_z)\in H} W(r \to x_z), \max\{\mu \times (T(x_z, x_z \to r)) | T(x_z) \in K - H\}\right\}$, for each subset H of $\{T(x_1), \ldots, T(x_p)\}$. $\square$

Theorem 17. *The EOP on trees can be solved in* $O(n \log n)$*-time.*

Proof. Lemmas 12 and 13 imply that an optimal edge-orientation scheme of $T(r)$ can be obtained by recursively deriving $\mu(T(x_j, r \to x_j))$ and $\mu(T(x_j, x_j \to r))$ for each subtree $T(x_j)$ and then examining each child of r constant times after sorting all subtrees $T(x_j)$. Let Time(n) be the time-complexity of the EOP on $T(r)$. The following equations can be established, where $V(T(x_j))$ is the vertex-set of each subtree $T(x_j)$:

$$\mathrm{Time}(n) = \sum_{j=1}^{p} [\mathrm{Time}(|V(T(x_j))|)] + O(p \log p) = O(n \log n). \qquad \square$$

7. Conclusions

Graph orientation is a fundamental and important topic with rich research results. Most of the previous research dealt with unweighted graphs. This paper has proposed a new research issue, the edge-orientation problem (the EOP), which transforms undirected graphs with costs on vertices and weights on edges to digraphs in order to minimize some practical cost measurements. Two important variants of the EOP, the Out-Degree-EOP and the Vertex-Weighted EOP, have been proposed and studied. Table 1 summarizes the results achieved and makes a comparison of our results to previous works.

In the future, some research directions deserve detailed attention. First, the techniques used and the algorithmic results will greatly help the implementation of the WFQ on real networks to enhance utilization of links. Studying how to really apply the EOP to the WFQ and network flow control is an interesting and meaningful issue. Solving the EOP on other classes of graphs, such

Table 1
Comparison of our results to previous works

Problem	Our result	Previous results
The Out-Degree-EOP	(1) $O(mn\delta \log \chi)$-time on general graphs, where $\delta = \max_{v \in V}\{\deg(v)\}$, in which $\deg(v)$ denotes the degree of any vertex v and $\chi = \max\left\{1, \delta - \left\lceil \frac{m}{n} \right\rceil\right\}$ (2) $O(mn)$-time on graphs with bounded degrees	$O(m^2)$-time on general graphs [36]
The Vertex-Weighted EOP	(1) $O(mn\delta \log \delta)$-time on general graphs (2) $O(mn)$-time on graphs with bounded degrees	None
The EOP	(1) NP-hard on bipartite graphs and chordal graphs (2) $O(n \log n)$-time on trees	None

as cactus graphs, block graphs, interval graphs, is another important future research topic. Finally, examining the EOP and its other practical variants will be very meaningful in identifying the properties which will help solve fundamental problems such as shortest paths and domination.

References

[1] S.N. Bhatt, J. Crowcroft, QoS sensitive flows: issues in IP packet handling, IEEE Internet Computing Journal 4 (4) (2000) 48–57.

[2] F. Boesch, R. Tindell, Robbins's theorem for mixed multigraphs, The American Mathematical Monthly 87 (1980) 716–719.

[3] F.R.K. Chung, M.R. Garey, R.E. Tarjan, Strongly connected orientations of mixed multigraphs, Networks 15 (1985) 477–484.

[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, second ed., The MIT Press, New York, 2001.

[5] W.H. Cunningham, A. Frank, A primal-dual algorithm for submodular flows, Mathematics of Operations Research 10 (1985) 251–262.

[6] W.C. Feng, D.D. Kandlur, D. Saha, K.G. Shin, Understanding and improving TCP performance over networks with minimum rate guarantees, IEEE-ACM Transactions on Networking 7 (2) (1999) 173–187.

[7] M.L. Fisher, D.S. Hochbaum, Database location in computer network, Journal of Association Computer Machine 27 (4) (1980) 781–785.

[8] A. Frank, Orientations of graphs and submodular flows, in: A.J.W. Hilton (Ed.), Special volume for the retirement of Crispin Nash-Williams from Reading, Congressus Numerantium 113 (1996) 111–142.

[9] A. Frank, Conservative weighting and ear-decomposition of graphs, Combinatorica 13 (1993) 65–81.

[10] A. Frank, On the orientation of graphs, Journal of Combinatorial Theory Series B 28 (1980) 251–261.

[11] A. Frank, Kombinatorikus Algoritmusok, Algorithmikus Bizonyítások (Hungarian; Combinatorial Algorithms, Algorithmic Proofs), Ph.D. Thesis, Department of Mathematics, Eötvös Loránd Tudományegyetem, Budapest, 1976.

[12] A. Frank, A. Gyárfás, How to orient the edges of a graph? in: A. Hajnal, V.T. Sós (Eds.), Proceedings of 5th Hungarian Colloquium on Combinatirics, Keszthely, 1976, Combinatorics, vol. I, North-Holland, Amsterdam, 1978, pp. 353–364.

[13] A. Frank, T. Jordán, Z. Szigeti, An orientation theorem with parity constraints, Discrete Applied Mathematics 115 (2001) 37–47.

[14] A. Frank, Z. Király, Graph orientations with edge-connection and parity constraints, Combinatorica 22 (2002) 47–70.

[15] A. Frank, Z. Király, Parity constrained k-edge-connected orientations, in: G. Cornuéjols, R.E. Burkard, G.J. Woeginger (Eds.), Integer Programming and Combinatorial Optimization, Proceedings of 7th IPCO Conference, Graz, 1999, Lecture Notes in Computer Science 1610, Springer, Berlin, 1999, pp. 191–201.

[16] S. Fujishige, H. Röck, U. Zimmermann, A strongly polynomial algorithm for minimum cost submodular flow problems, Mathematics of Operations Research 14 (1989) 60–69.

[17] H.N. Gabow, Centroids, representations, submodular flows, Journal of Algorithms 18 (1995) 586–628.

[18] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Bell Laboratories/Freeman & Co., Murray Hill/NJ, 1978.

[19] F. Gavril, The intersection graphs of subtrees in tree are exactly the chordal graphs, Journal of Combinatorial Theory Series B 16 (1974) 47–56.
[20] M.C. Golumbics, Algorithmic Graph Theory and Perfect Graphs, Academic Press, Inc., New York, 1980.
[21] S.L. Hakimi, On the degrees of the vertices of a directed graphs, Journal of the Franklin Institute 279 (1965) 290–308.
[22] S.L. Hakimi, E.F. Schmeichel, Locating replicas of a database on a network, Networks 30 (1) (1997) 31–36.
[23] R. Hassin, N. Megiddo, On orientations and shortest paths, Linear Algebra and Its Applications 114/115 (1989) 589–602.
[24] X. Jia, A distributed algorithm of delay bounded multicast routing for multimedia applications in wide area networks, IEEE/ACM Transactions on Networking Journal 6 (6) (1998) 828–837.
[25] Y. Jiang, C.K. Tham, C.C. Ko, Challenges and approaches in providing QoS monitoring, International Journal of Network Management 10 (6) (2000) 323–334.
[26] W. Korfhage, Dynamically load balancing prioritized processes in distributed systems, ISCA Journal 2 (3) (1995) 137–144.
[27] H. Lorenz, A. Orda, QoS routing in networks with uncertain parameters, IEEE-ACM Transactions on Networking 6 (6) (1998) 768–778.
[28] D. Medihi, Models for network design, servicing, and monitoring of ATM networks based on the virtual path concept, Computer Networks and ISDN Systems Journal 29 (1997) 373–386.
[29] D.L. Mills, Adaptive hybrid clock discipline algorithm for the network time protocol, IEEE-ACM Transactions on Networking 6 (5) (1998) 505–514.
[30] C. St. J.A. Nash-Williams, On orientations, connectivity and odd-vertex-parings in finite graphs, Canadian Journal of Mathematics 12 (1960) 555–567.
[31] L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, second ed., Morgan Kaufmann Publishers, CA, USA, 2000.
[32] H.E. Robbins, A theorem on graphs, with an application to a problem of traffic control, The American Mathematical Monthly 46 (1939) 281–283.
[33] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, vol. B, Springer-Verlag, Berlin, Heidelberg, New York, 2003.
[34] C.K. Tham, Y. Jiang, C.C. Ko, Monitoring QoS distribution in distributed multimedia networks, International Journal of Network Management 10 (2) (2000) 75–90.
[35] T. Tsichiya, T. Kikuno, Dependable evaluation of the weighted voting system in the presence of node and link facilities, International Journal of Computer Systems Science & Engineering 15 (3) (2000) 181–190.
[36] V. Venkateswaran, Minimizing maximum indegree, Discrete Applied Mathematics 143 (2004) 374–378.
[37] W.C.-K. Yen, The link-orientation problem and its variants with applications, Shih Hsin University Journal 11 (2001) 265–285.
[38] W.C.K. Yen, S.-J. Yang, The link-orientation problem on weighted networks, in: Proceedings of the 16th International Conference on Computer and Their Applications (CATA-2001), Seattle, Washington, USA, 2001, pp. 325–329.
[39] W.C.K. Yen, C.Y. Tang, The searchlight guarding problem on weighted split graphs and weighted cographs, Networks 35 (4) (2000) 195–206.
[40] W.C.K. Yen, C.Y. Tang, An optimal algorithm for solving the searchlight guarding problem on weighted interval graphs, Information Sciences 100 (1997) 1–25.