Lecture 22: Matchings in bipartite graphs

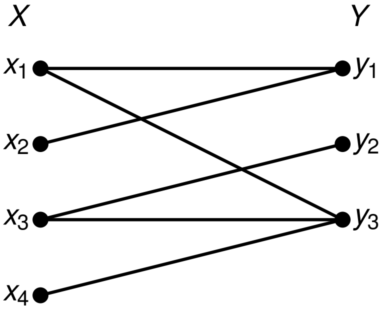
Janko Gravner

MAT 145

Mar. 3, 2021

Bipartite graph

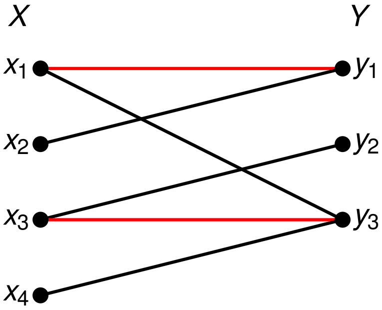
A bipartite graph has vertices divided into two disjoint sets $X = \{x_1, \ldots, x_m\}$ and $Y = \{x_1, \ldots, x_n\}$, and a set of edges Δ so that every edge is between a vertex in X and a vertex in Y. (The edges can be oriented or unoriented; it does not matter for this topic.) For example:

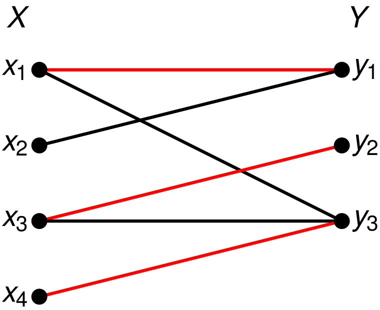


An edge between x_i and y_j can be thought of as a possible match between x_i and y_j. We can think of X and Y as sets of men and women in a matchmaking situation, as sets of jobs and workers, as sets of tasks and computer processors, etc.

Matching in a bipartite graph

A *matching* in a bipartite graph $G = (X, \Delta, Y)$ is a subset M of Δ, such that no two edges of M meet at a single vertex. Here are two matchings: $\{(x_1, y_1), (x_3, y_3)\}$ and $\{(x_1, y_1), (x_3, y_2), (x_4, y_3)\}$





In a matching M, vertices which are not part of an edge in M are called *free*.

Maximal matching

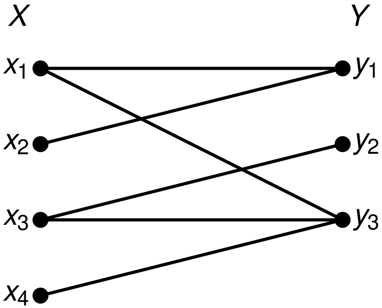
We want to find a *maximal* matching M^* in the graph G, which has the largest possible number of edges, i.e., matched pairs:

$$|M^*| = \max\{|M| : M \text{ is a matching}\} = \rho(M).$$

The first thing to observe is that if M is inclusion-maximal (no edge can be added while preserving the matching condition), it is not necessarily maximal; see our first matching on the previous slide. Also, M^* is not necessarily unique. In our example, $\rho(G) = 3$, and we have 3 maximal matchings.

This is not a new problem

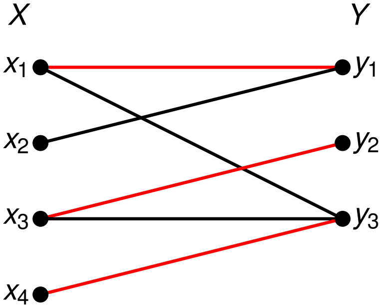
For a bipartite graph G, we can form the *compatibility matrix*:



	y_1	y_2	y_3
x_1	♡		♡
x_2	♡		
x_3		♡	♡
x_4			♡

This is not a new problem

A matching is exactly the same as a placement of nonattacking rooks on ♡'s:



	y_1	y_2	y_3
x_1	♡		♡
x_2	♡		
x_3		♡	♡
x_4			♡

We can compute the rook polynomial by eliminating the ♡ in the second column:

$$1 + 5x + 5x^2 + x(1 + 4x + 3x^2) = 1 + 6x + 9x^2 + 3x^3,$$

so we have 3 maximal matchings. If we just want to determine $\rho(G)$ and find one M^*, this is not an efficient method.

Alternating chain

Suppose M is a matching in $G = (X, \Delta, Y)$. Suppose that

$$v_1, w_1, v_2, w_2, \ldots, v_k, w_k$$

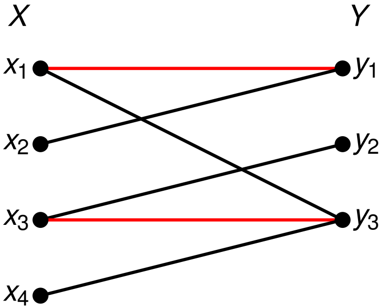
is a sequence of *distinct* vertices such that $v_i \in X$, $w_i \in Y$ for all i, v_1 and w_k are free, and moreover the successive pairs of vertices alternate between being connected by edges not in M and edges in M:

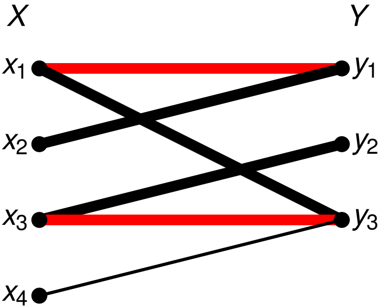
- $v_1 - w_1, \ldots, v_k - w_k \in \Delta \setminus M$; and
- $w_1 - v_2, \ldots, w_{k-1} - v_k \in M$.

Then the sequence of vertices together with the edges joining them in order is an *alternating chain*.

Alternating chain

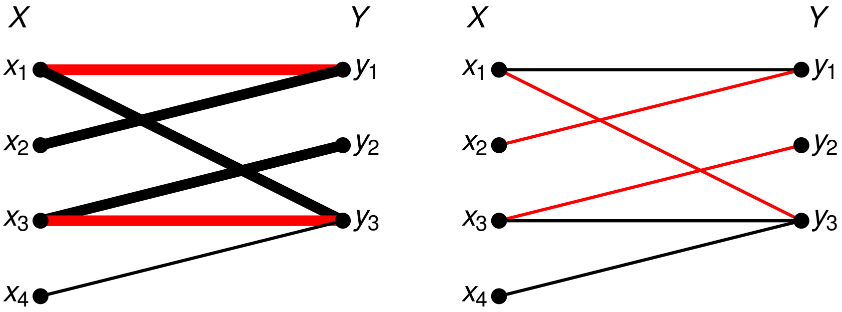
Here is an alternating chain for the matching on the left, beginning at x_2 and ending at y_2.





Alternating chain

If M has an alternating chain, then it cannot be maximal, because we can switch the status of all edges on the chain and obtain a larger matching.



Even more ...

Alternating chain

Theorem

A matching M of $G = (X, \Delta, Y)$ is maximal if and only if it has no alternating chain.

Proof.

Assume M' is a matching and $|M'| > |M|$. We need to find an alternating chain. Let Δ^* be the set of edges that consist of edges that are in M or in M', but not in both, that is,

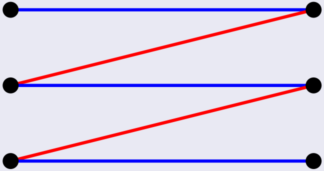
$$\Delta^* = (M' \setminus M) \cup (M \setminus M')$$

As $|M'| > |M|$, $|M' \setminus M| > |M \setminus M'|$. In the graph (X, Δ^*, Y), every vertex is an endpoint of at most one edge in $M' \setminus M$ and also of at most one edge in $M \setminus M'$. So Δ^* can be partitioned into paths and cycles of the following four types, with red edges in $M \setminus M'$ and blue in $M' \setminus M$.

Alternating chain

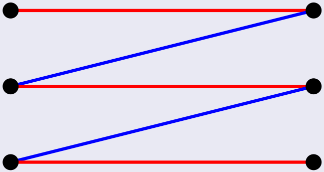
Proof, continued.

Type 1:



Begin and end in $M' \setminus M$.

Type 2:

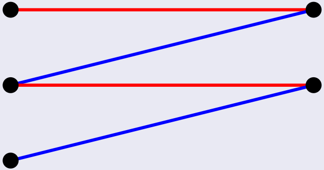


Begin and end in $M \setminus M'$.

Alternating chain

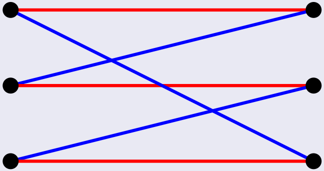
Proof, continued.

Type 3:



Begin in $M' \setminus M$ and end in $M \setminus M'$, or vice versa.

Type 4:



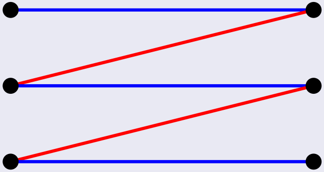
Cycle.

Alternating chain

Proof, continued.

Type 1 component is the only one with more blue than red edges, so there must be at least one such component.

Type 1:



Begin and end in $M' \setminus M$.

But this is also an alternating chain for M, as the first and last vertex are free for M: they cannot be endpoints any additional edge in M', and so they cannot be endpoints any edge in M (or that edge would be in $M \setminus M'$ and the path would continue). □

Matching algorithm

There is an efficient algorithm that proves an alternating chain does not exist and otherwise finds one.

Start with a matching M.

(A) Label all free X-vertices by $[*]$.

(B) If w is any unlabeled Y-vertex joined by an edge in $\Delta \setminus M$ to a labeled X-vertex v, label w by $[v]$ (only give *one* label).

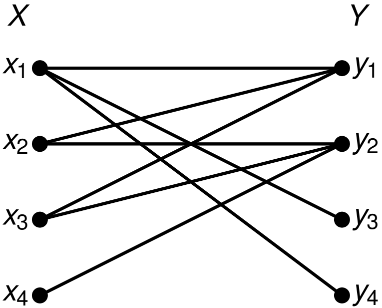
(C) If v is any unlabeled X-vertex joined by an edge in M to a labeled Y-vertex w, label v by $[w]$.

(D) Repeat steps (B) and (C) repeatedly until:
  - (f) a free Y-vertex is labeled; or
  - (n) nothing more is is labeled.

In case (f), M has an alternating chain, traced backwards from the free Y-vertex. In case (n), if there is an alternating chain, all vertices on it have to be labeled, so eventually we label a free Y-vertex. So, in case (n), M has no alternating chain.

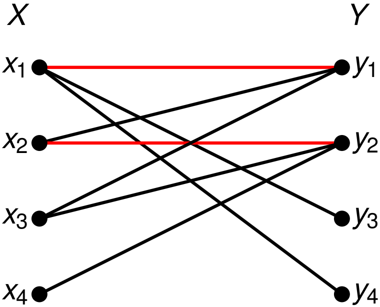
Matching algorithm: example

Example 22.1. Find the maximal matching in the following bipartite graph G.



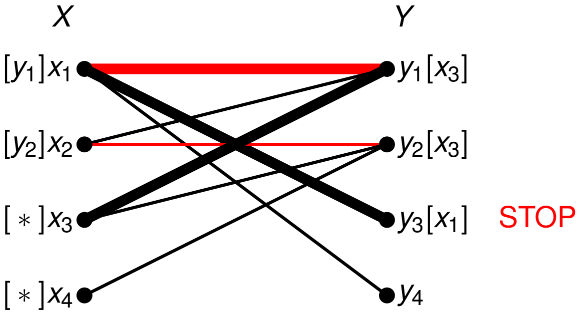
Matching algorithm: example

We start with an obvious matching and run the algorithm.



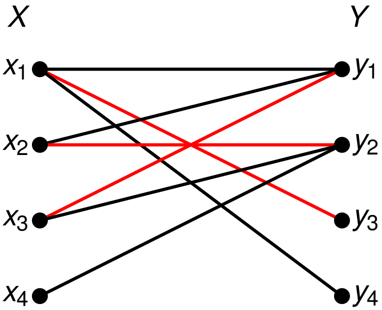
Matching algorithm: example

The algorithm stops at a free Y vertex and therefore we have a alternating chain ...



Matching algorithm: example

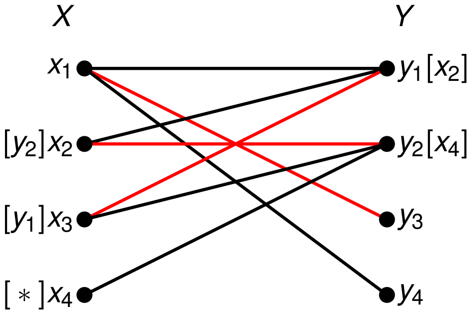
...which gives us a larger matching:



We run the algorithm on this matching to see if it is maximal.

Matching algorithm: example

The algorithm stops without finding a free Y-vertex. This is a maximal matching.



In this case, then, $\rho(G) = 3$.