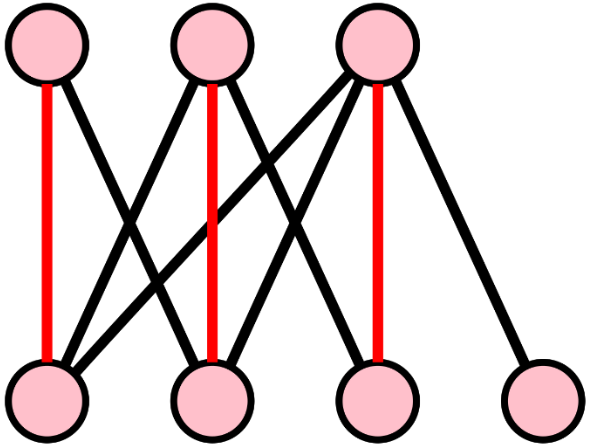
Graph Matching

Reading from Textbook *Algorithms: Special Topics*

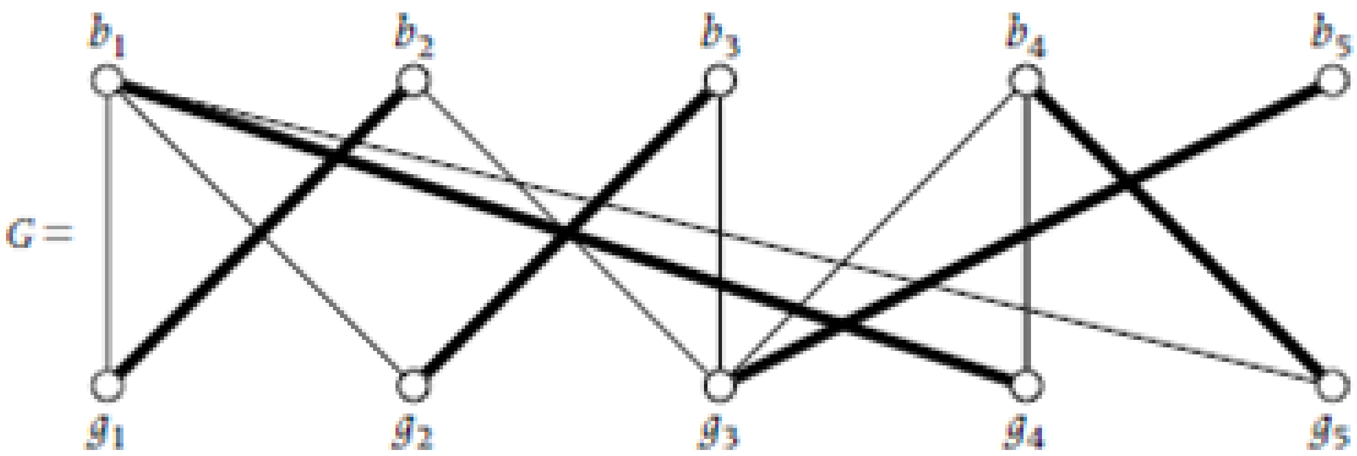
Chapter 4, Section 4.1,

pp. 96-103

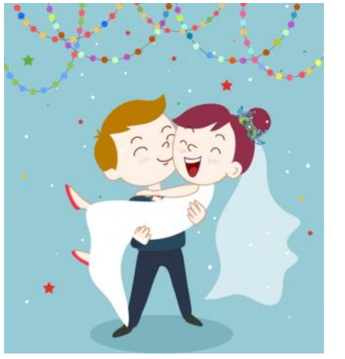


Perfect Matchings in Bipartite Graphs

An *independent set of edges* or *matching* in a graph G is a set of edges that are pairwise vertex disjoint. A matching that spans the vertices is called a *perfect matching*.



The Marriage Problem



The marriage problem is the problem of finding a perfect matching in a bipartite graph where the number of vertices on each side of the bipartition is the same. It gets it name from the interpretation that n boys are to be matched with n girls for marriage, but only if they know each other, i.e., we include the edge $\{b, g\}$ in the bipartite graph whenever boy b knows girl g.

Necessary and Sufficient Condition for Existence of a Perfect Matching

Clearly, if the is a set of *k* boys who collectively know fewer than *k* girls then no perfect matching can exist.

Converse is true!

A solution to the Marriage Problem exists if, and only if, every set of *k* boys collectively knows at least *k* girls, $k \in \{1, 2, \ldots, n\}$. This was discovered by P. Hall and is known as Hall's Theorem.

Neighborhood of a Set of Vertices

The neighborhood of a set *S* of vertices is the set $\Gamma(S)$ of all vertices that are adjacent to at least one vertex from *S*, i.e.,

$$\Gamma(S) = \{v \in V\colon \{v, s\} \in E \text{ for some } s \in S\}$$

Hall's Theorem

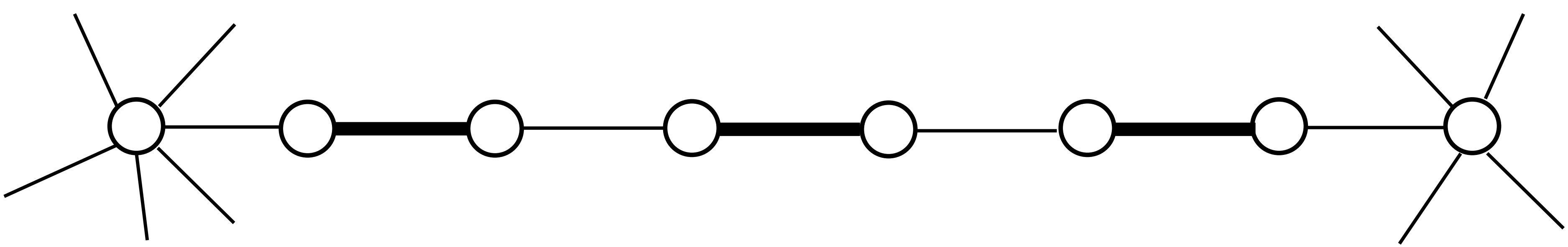
Theorem 4.11* (P. Hall). Let $G = (V, E)$ be a bipartite graph with vertex bipartition $V = X \cup Y$, where $|X| = |Y|$. Then, G contains a perfect match iff $|S| \leq |\Gamma(S)|$ for all $S \subseteq X$.

*There is an error in the text. The specification that *X* and *Y* are of equal size is missing.

M-augmenting path

Given a matching *M*, we say that a vertex *v* of *G* is ***M-matched*** if it is incident with an edge of *M*; otherwise, we say that *v* is ***M-unmatched***.

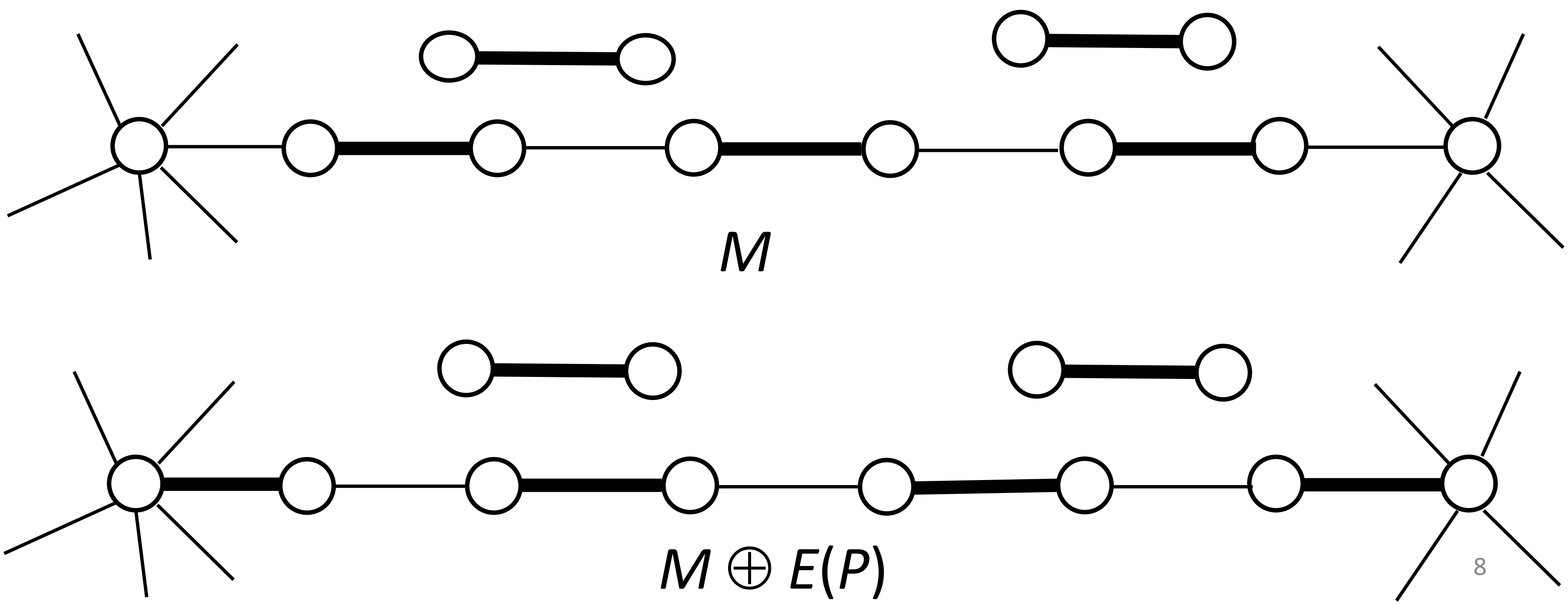
An *M-alternating path* is one where the edges alternately belong to *E*(*G*) – *M* and *M*. An ***M-augmenting path*** is an alternating path where the initial and terminal vertices are both *M*-unmatched.



Applying M-augmenting path

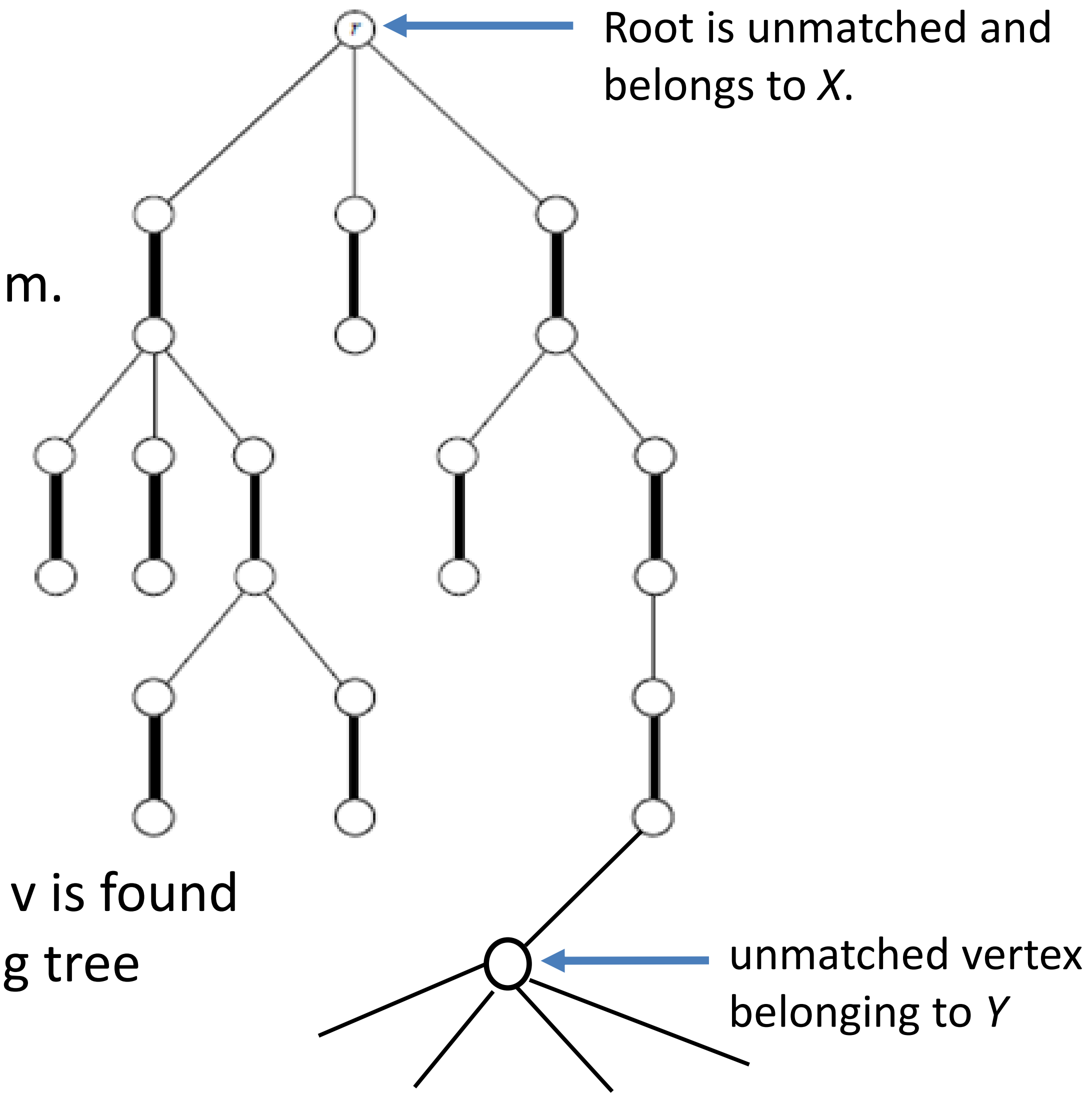
Given an M-augmenting path P, the size of the matching M can be increased by one by removing the edges of M belonging to P, and adding to M the remaining edges of P, i.e.,

$$\text{replacing } M \text{ with } M \oplus E(P)$$



M-alternating Tree

Grow an *M*-alternating tree starting from an unmatched vertex *r* in *X* using a slightly modified BFS or DFS algorithm.



When an unmatched vertex v is found the path in the *M*-alternating tree from the root *r* to *v* is an *M*-augmenting path.

Hungarian Algorithm

The Hungarian algorithm keeps computing M-augmenting paths and increasing the size of the matching by one by replacing M with $M \oplus E(P)$

until either

- a perfect matching has be computed

or

- an M-alternating tree has been computed that cannot be further expanded, i.e., there is no edge from a vertex of X belonging to the tree to a vertex not belonging to the tree.

Correctness of Hungarian Algorithm

X_T = set of vertices in T belonging to X
Y_T = set of vertices in T belonging to Y

Since all vertices in tree are matched except for the root

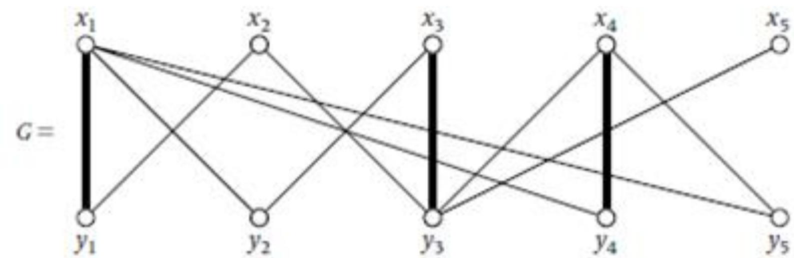
$$|X_T| = |Y_T| + 1$$

Since there are no edges from any vertex in X to a vertex not in the tree $\Gamma(X_T) = Y_T$. Thus, setting $S = X_T$, we have

$$|S| = |\Gamma(S)| + 1 > |\Gamma(S)| .$$

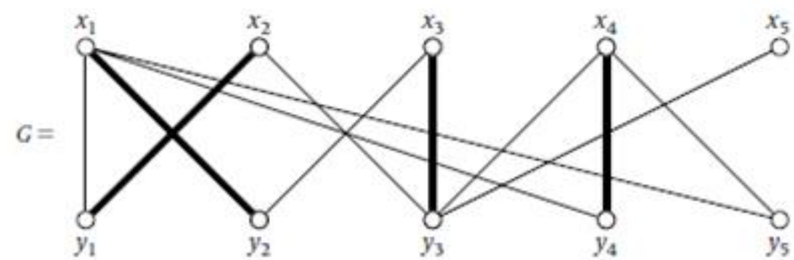
Therefore, there is no perfect matching!

Action for a sample bipartite graph



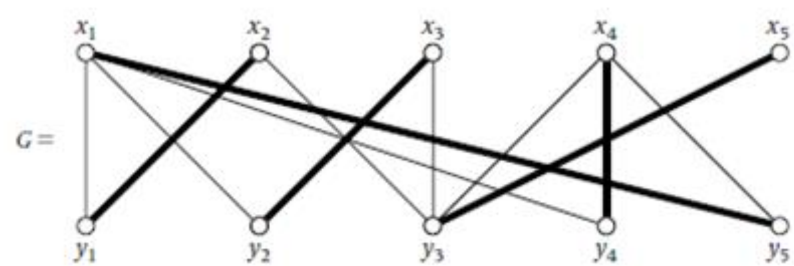
Sample bipartie graph G and initial matching $\{x_1,y_1\}$, $\{x_3,y_3\}$, $\{x_4,y_4\}$

Using augmented path $x_2y_1x_1y_2$



A first M-augmenting path P: $x_2y_1x_1y_2$, found by *Hungarian* with starting M-unmatched vertex x_2 yielding augmented matching $M \oplus E(P)=\{x_1y_2, x_2y_1, x_3y_3, x_4y_4\}$

Using augmented path $x_5y_3x_3y_2x_1y_5$



Complexity Analysis

In the worst-case the Hungarian algorithm for a bipartite graph with *n* vertices and m edges will involve $\frac{n}{2} - 1$ stages of finding an *M*-augmenting path and increasing the size of the matching by one. Finding an *M*-augmenting path using an algorithm similar to BFS or DFS takes time $O(m)$. Therefore, we have worst-case complexity

$$W(n) \in O(mn)$$

The fastest know algorithm for computing a perfect matching in a bipartite graph know as the Hopcroft–Karp algorithm has worst-case complexity

$$W(n) \in O(m\sqrt{n})$$

Perfect Matchings in Cubic Bipartite Graphs

A **cubic** graph is a graph where every vertex has degree 3.

Proposition. Every cubic bipartite graph contains a perfect matching.

Proof of Proposition

Consider **any** set $S \subseteq X$.

Let $H = (V_H, E_H)$ be the subgraph of G induced by the set of edges

$$E_H = \{uv\colon u \in S \text{ and } v \in \Gamma(S)\}.$$

Thus, $V_H \cap X = S$ and $V_H \cap Y = \Gamma(S)$.

Clearly, $d(v) = 3, \forall v \in V_H \cap X$ and $d(v) \leq 3, \forall v \in V_H \cap Y$. Now

$$\sum_{v \in V_H \cap X} d(v) = |E_H| = \sum_{v \in V_H \cap Y} d(v)$$

$$\Leftrightarrow 3|S| = |E_H| \leq 3|\Gamma(S)|$$

$$\Leftrightarrow |S| \leq |\Gamma(S)|$$

It follows from Hall's Theorem that G has a perfect matching.

PSN. Prove the more general proposition.

Proposition. Every r-regular bipartite graph contains as perfect matching.

Corollary

A simple induction shows that any r-regular bipartite graph can be partitioned into perfect matchings, i.e., the edges can be r-colored so that adjacent edges are colored differently.

Hot Matching