BIL694-Lecture 2: Matchings and Covers

Lecturer: Lale Özkahya

Resources for the presentation:
http://www.inf.ed.ac.uk/teaching/courses/dmmr/
http://www.cs.princeton.edu/courses/archive/spr11/cos423/Lectures/GraphMatching.pdf
http://www.cs.princeton.edu/courses/archive/spr11/cos423/Lectures/NonbipartiteMatching

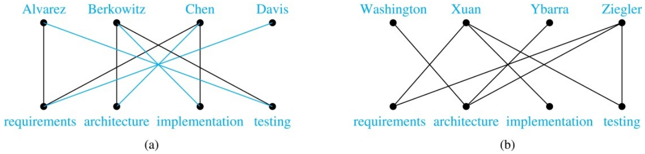
Outline

Outline

Bipartite Graphs and Matchings

Bipartite graphs used extensively in app's involving matching elements of two sets:

Job assignments - vertices represent the jobs and the employees, edges link employees with jobs they are qualified for. Maximize # of employees matched to jobs.



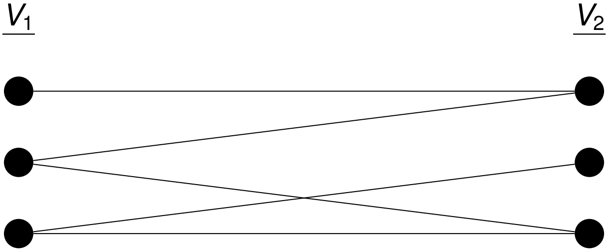
Marriage/dating - vertices represent men & women and edges link a man & woman if they are acceptable to each other as partners.

Bipartite graphs

A bipartite graph is a (undirected) graph $G = (V, E)$ whose vertices can be partitioned into two disjoint sets (V_1, V_2), with $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$, such that for every edge $e \in E$, $e = \{u, v\}$ such that $u \in V_1$ and $v \in V_2$. In other words, every edge connects a vertex in V_1 with a vertex in V_2.

Equivalently, a graph is bipartite if and only if it is possible to color each vertex red or blue such that no two adjacent vertices are the same color.

Example of a Bipartite Graph



Matchings in Bipartite Graphs

A **matching**, M, in a graph, $G = (V, E)$, is a subset of edges, $M \subseteq E$, such that there does not exist two distinct edges in M that are incident on the same vertex. In other words, if $\{u, v\}, \{w, z\} \in M$, then either $\{u, v\} = \{w, z\}$ or $\{u, v\} \cap \{w, z\} = \emptyset$.

A **maximum matching** in graph G is a matching in G with the maximum possible number of edges.

Perfect/complete matchings

For a graph $G = (V, E)$, we say that a subset of edges, $W \subseteq E$, **covers** a subset of vertices, $A \subseteq V$, if for all vertices $u \in A$, there exists an edge $e \in W$, such that e is incident on u, i.e., such that $e = \{u, v\}$, for some vertex v.

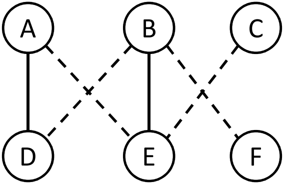
In a bipartite graph $G = (V, E)$ with bipartition (V_1, V_2), a **complete matching** with respect to V_1, is a matching $M' \subseteq E$ that covers V_1, and a **perfect matching** is a matching, $M^* \subseteq E$, that covers V.

Question: When does a bipartite graph have a perfect matching?

A bipartite graph

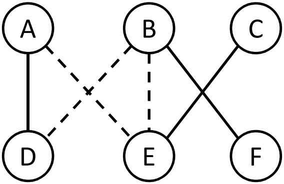
Solid edges are a matching

(maxim**al** but not maxim**um**)



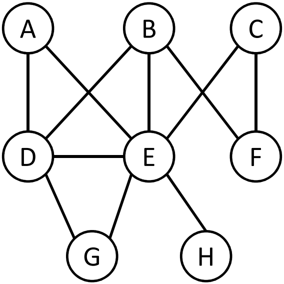
A *maximal* matching is one to which no additional edge can be added

Another matching, perfect hence maximum

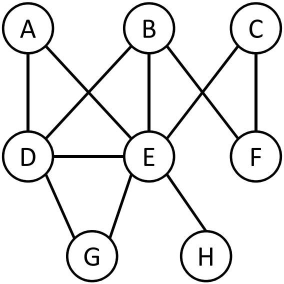


A nonbipartite graph

Does this graph have a perfect matching?



No: Each of A, G, H must be matched to D or E



Alternating Path and Augmenting Path in Bipartite (A, B)-graph

A path in G which starts in A at an unmatched vertex and then contains, alternately, edges from $E \setminus M$ and from M, is an alternating path with respect to M.

Alternating Path and Augmenting Path in Bipartite (A, B)-graph

A path in G which starts in A at an unmatched vertex and then contains, alternately, edges from $E \setminus M$ and from M, is an alternating path with respect to M.

An alternating path P that ends in an unmatched vertex of B is called an augmenting path, because we use it to turn M into a larger matching.

Alternating Path and Augmenting Path in Bipartite (A, B)-graph

A path in G which starts in A at an unmatched vertex and then contains, alternately, edges from $E \setminus M$ and from M, is an alternating path with respect to M.

An alternating path P that ends in an unmatched vertex of B is called an augmenting path, because we use it to turn M into a larger matching.

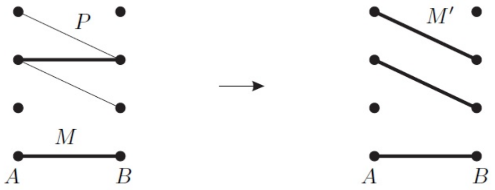


Figure: Augmenting the matching M by the alternating path P

Matchings in Bipartite Graphs: Hall's Condition

A matching M in a bipartite (A, B)-graph is said to saturate A if each vertex in A is contained in some edge of M.

Matchings in Bipartite Graphs: Hall's Condition

A matching M in a bipartite (A, B)-graph is said to saturate A if each vertex in A is contained in some edge of M.

Hall's Condition: The condition that $|N(S)| \geq |S|$ for all $S \subset A$ is called the Hall's condition for finding a matching that saturates A.

Matchings in Bipartite Graphs: Hall's Condition

A matching M in a bipartite (A, B)-graph is said to saturate A if each vertex in A is contained in some edge of M.

Hall's Condition: The condition that $|N(S)| \geq |S|$ for all $S \subset A$ is called the Hall's condition for finding a matching that saturates A.

Theorem (Hall, 1935): G contains a matching that saturates A if and only if $|N(S)| \geq |S|$ for all $S \subset A$.

A proof of Hall's Theorem

Proof by induction:

- Apply induction on $|A|$. For $|A| = 1$, clearly the theorem holds. Let $|A| \geq 2$ and assume that Hall's condition is sufficient of a mathing that saturates A when $|A|$ is smaller.

A proof of Hall's Theorem

Proof by induction:

- Apply induction on $|A|$. For $|A| = 1$, clearly the theorem holds. Let $|A| \geq 2$ and assume that Hall's condition is sufficient of a mathing that saturates A when $|A|$ is smaller.
- Case 1: $|N(S)| \geq |S| + 1$ for every non-empty *proper* $S \subset A$.

  pick an edge ab, let $G' := G - \{a, b\}$ with $a \in A$, $b \in B$. Then

A proof of Hall's Theorem

Proof by induction:

- Apply induction on $|A|$. For $|A| = 1$, clearly the theorem holds. Let $|A| \geq 2$ and assume that Hall's condition is sufficient of a mathing that saturates A when $|A|$ is smaller.
- Case 1: $|N(S)| \geq |S| + 1$ for every non-empty *proper* $S \subset A$.

  pick an edge ab, let $G' := G - \{a, b\}$ with $a \in A$, $b \in B$. Then

  $$|N_{G'}(S)| \geq |N_G(S)| - 1 \geq |S|$$

  for every $S \subset A \setminus \{a\}$.

A proof of Hall's Theorem

Proof by induction:

- Apply induction on $|A|$. For $|A| = 1$, clearly the theorem holds. Let $|A| \geq 2$ and assume that Hall's condition is sufficient of a mathing that saturates A when $|A|$ is smaller.
- Case 1: $|N(S)| \geq |S| + 1$ for every non-empty *proper* $S \subset A$.

  pick an edge ab, let $G' := G - \{a, b\}$ with $a \in A$, $b \in B$. Then

  $$|N_{G'}(S)| \geq |N_G(S)| - 1 \geq |S|$$

  for every $S \subset A \setminus \{a\}$.
- G' contains a matching that saturates $A \setminus \{a\}$ by inductive hypothesis, this matching together with ab is a matching of G.

A proof of Hall's Theorem

Proof by induction (continues):

- Case 2: There exists a proper subset $A' \subsetneq A$ with $|N(A')| = |A'|$, let $B' = N(A')$.

A proof of Hall's Theorem

Proof by induction (continues):

- Case 2: There exists a proper subset $A' \subsetneq A$ with $|N(A')| = |A'|$, let $B' = N(A')$.
- $G' := G[A' \cup B']$ contains a matching saturating A' (Ind. Hypo.)

A proof of Hall's Theorem

Proof by induction (continues):

- Case 2: There exists a proper subset $A' \subsetneq A$ with $|N(A')| = |A'|$, let $B' = N(A')$.
- $G' := G[A' \cup B']$ contains a matching saturating A' (Ind. Hypo.)
- $G - G'$ also satisfies Hall's condition. Why?
  (Consider $N_G(S \cup A')$ if $S \subset A - A'$ does not satisfy Hall's condition).

A proof of Hall's Theorem

Proof by induction (continues):

- Case 2: There exists a proper subset $A' \subsetneq A$ with $|N(A')| = |A'|$, let $B' = N(A')$.
- $G' := G[A' \cup B']$ contains a matching saturating A' (Ind. Hypo.)
- $G - G'$ also satisfies Hall's condition. Why?
  (Consider $N_G(S \cup A')$ if $S \subset A - A'$ does not satisfy Hall's condition). $G - G'$ contains a matching saturating $A \setminus A'$. Done.

Matchings in Bipartite Graphs: König-Egerváry Theorem

A subset $U \subset V$ in a graph $G = (V, E)$ is called a vertex cover if every edge of G is incident with a vertex in U.

Matchings in Bipartite Graphs: König-Egerváry Theorem

A subset $U \subset V$ in a graph $G = (V, E)$ is called a vertex cover if every edge of G is incident with a vertex in U.

Theorem (König, 1931), Egerváry (1931): The maximum size of a matching in a bipartite (X, Y)-graph is equal to the minimum order of a vertex cover of its edges.

Matchings in Bipartite Graphs: König-Egerváry Theorem

A subset $U \subset V$ in a graph $G = (V, E)$ is called a vertex cover if every edge of G is incident with a vertex in U.

Theorem (König, 1931), Egerváry (1931): The maximum size of a matching in a bipartite (X, Y)-graph is equal to the minimum order of a vertex cover of its edges.

Proof: Let Q be a minimum vertex cover. Trivially, a vertex cover different vertex to cover each edge in M. Therefore, $|Q| \geq |M|$.

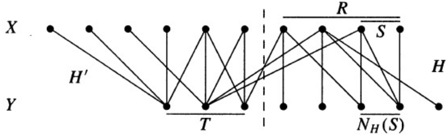
Matchings in Bipartite Graphs: König-Egerváry Theorem

A subset $U \subset V$ in a graph $G = (V, E)$ is called a vertex cover if every edge of G is incident with a vertex in U.

Theorem (König, 1931), Egerváry (1931): The maximum size of a matching in a bipartite (X, Y)-graph is equal to the minimum order of a vertex cover of its edges.

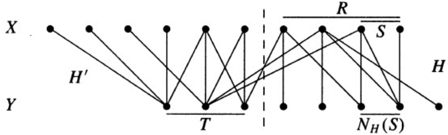
Proof: Let Q be a minimum vertex cover. Trivially, a vertex cover
different vertex to cover each edge in M. Therefore, $|Q| \geq |M|$.
Next: Show that a minimum vertex cover Q has also at most $|M|$ vertices.

Matchings in Bipartite Graphs: König-Egerváry Theorem



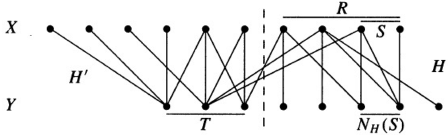
- Partition Q into the sets R and T, where $R = Q \cap X$ and $T = Q \cap Y$.

Matchings in Bipartite Graphs: König-Egerváry Theorem



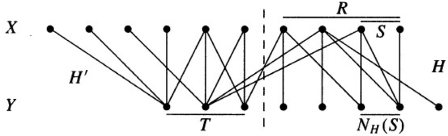
- Partition Q into the sets R and T, where $R = Q \cap X$ and $T = Q \cap Y$.
- Let $H = G[R \cup (Y - T)]$ and $H' = G[T \cup (X - R)]$.
  Use Hall's theorem to show that H has a matching saturating R and H' has a matching saturating T.

Matchings in Bipartite Graphs: König-Egerváry Theorem



- Partition Q into the sets R and T, where $R = Q \cap X$ and $T = Q \cap Y$.
- Let $H = G[R \cup (Y - T)]$ and $H' = G[T \cup (X - R)]$.
  Use Hall's theorem to show that H has a matching saturating R and H' has a matching saturating T.
- To do that, we need to show that Hall's condition holds for these graphs. (Observe that there is no edge between the sets $Y - T$ and $X - R$. If Hall's condition does not hold for some S, we could obtain a smaller vertex cover, contradiction.)

Matchings in Bipartite Graphs: König-Egerváry Theorem



- Partition Q into the sets R and T, where $R = Q \cap X$ and $T = Q \cap Y$.
- Let $H = G[R \cup (Y - T)]$ and $H' = G[T \cup (X - R)]$. Use Hall's theorem to show that H has a matching saturating R and H' has a matching saturating T.
- To do that, we need to show that Hall's condition holds for these graphs. (Observe that there is no edge between the sets $Y - T$ and $X - R$. If Hall's condition does not hold for some S, we could obtain a smaller vertex cover, contradiction.)
- Since H and H' are vertex-disjoint, these the union of these two mathings is a matching of G. Done.

Outline

Efficient matching algorithm?

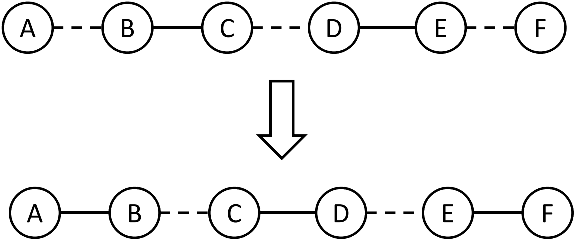
Iterative improvement: Start with any matching. Find a way to improve it by making local changes. Repeat until no improvement is possible. Hope: Any local maximum is a global maximum

Alternating path: a path whose edges are alternately in and out of the matching

Augmenting path: an alternating path between two free vertices

Augmentation: given an augmenting path, change its unmatched edges to matched and vice-versa, increasing the size of the matching by one

A, F free



A, F matched

Augmenting path algorithm

Start with the empty matching. While there is an augmenting path, do an augmentation.

Theorem: A matching has maximum size iff there is no augmenting path

Proof: to follow

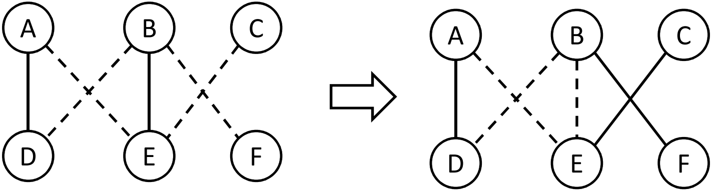
How to find augmenting paths?

How to choose augmenting paths?

augmenting path

C, E, B, F

after augmentation



Matching Theorem: Let M be any matching, let M' be a maximum-size matching, and let $k = |M'| - |M|$. Then M has k vertex-disjoint augmenting paths

Proof: Let $M' \oplus M$ be the *symmetric difference* of M' and M, the set of edges in M' or M but not both. Each vertex is incident to at most two edges in $M' \oplus M$. The connected components of the subgraph induced by the edges in $M' \oplus M$ are thus simple paths and simple cycles.

Proof (cont.): On each such path or cycle, edges of M' and M alternate. Each cycle contains the same number of edges in M' as in M. Each path contains the same number of edges in M' as in M to within one. A path that contains one more edge of M' than M is an augmenting path for M. In $M' \oplus M$ there are exactly k more edges in M' than edges in M. Thus the subgraph induced by the edges in $M' \oplus M$ contains k vertex-disjoint augmenting paths for M (and no augmenting paths for M').

Corollary: If *M* is a matching whose size is *k* less than maximum, then *M* has an augmenting path of at most n/k vertices.

Both the theorem and its corollary are true for *all* graphs, not just bipartite ones

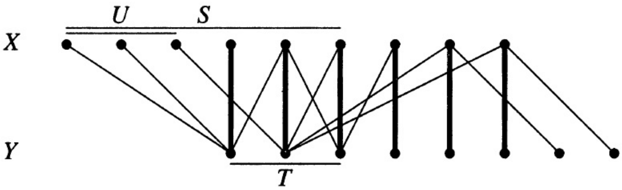
Augmenting Path Algorithm (West, Algorithm 3.2.1)

Input: An X, Y-bigraph G, a matching M in G, and the set U of M-unsaturated vertices.

Idea: Explore M-alternating paths from U, letting $S \subseteq X$ and $T \subseteq Y$ be the sets of vertices reached.

Mark vertices of S that have been explored for path extensions. As a vertex is reached, record the vertex from which it is reached.
Initialization: $S = U$ and $T = \emptyset$.

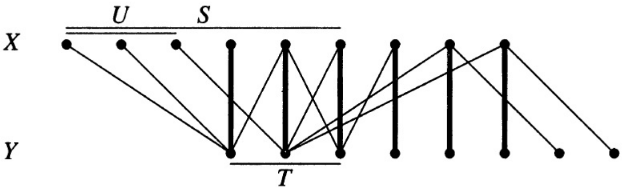
Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching.

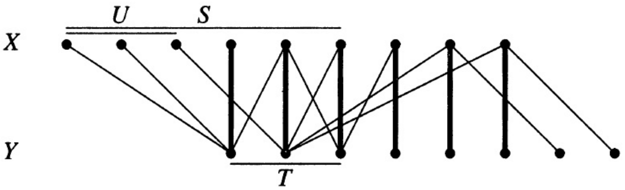
Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$.

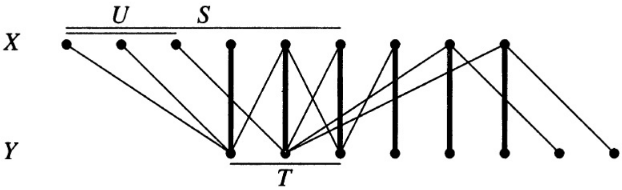
Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$.
- If y is unsaturated, terminate and report an M-augmenting path from U to y.

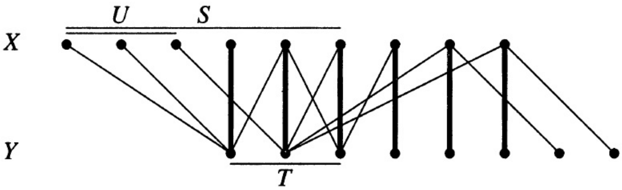
Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$.
- If y is unsaturated, terminate and report an M-augmenting path from U to y.
  Otherwise, y is matched to some $w \in X$ by M. In this case, include y in T (reached from x) and include w in S (reached from y).

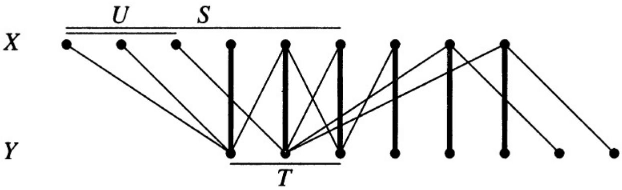
Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$.
- If y is unsaturated, terminate and report an M-augmenting path from U to y.
  Otherwise, y is matched to some $w \in X$ by M. In this case, include y in T (reached from x) and include w in S (reached from y).
- After exploring all such edges incident to x, mark x and iterate.

Augmenting Path Algorithm (West, Algorithm 3.2.1)



Iteration:

- If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching. Otherwise, select an unmarked $x \in S$. To explore x, consider each $y \in N(x)$ such that $xy \notin M$.
- If y is unsaturated, terminate and report an M-augmenting path from U to y.
  Otherwise, y is matched to some $w \in X$ by M. In this case, include y in T (reached from x) and include w in S (reached from y).
- After exploring all such edges incident to x, mark x and iterate.

Thm. The A.P. algorithm produces an M-augmenting path or a vertex cover of size $|M|$, which is $R = T \cup (X - S)$.

Stable Matchings

Problem: To establish n "stable" marriages given n men and n women.

unstable pair: If man x and woman a are paired with other partners, but x prefers a to his current partner and a prefers x to her current partner, then they might leave their current partners and switch to each other. In this case, we say that the unmatched pair (x, a) is an unstable pair.

stable matching: A perfect matching is a stable matching if it yields no unstable unmatched pair.

Stable Matchings

Problem: To establish n "stable" marriages given n men and n women.

unstable pair: If man x and woman a are paired with other partners, but x prefers a to his current partner and a prefers x to her current partner, then they might leave their current partners and switch to each other. In this case, we say that the unmatched pair (x, a) is an unstable pair.

stable matching: A perfect matching is a stable matching if it yields no unstable unmatched pair.

3.2.16. Example. Given men x, y, z, w, women a, b, c, d, and preferences below, the matching $\{xa, yb, zd, wc\}$ is a stable matching.

Men $\{x, y, z, w\}$	Women $\{a, b, c, d\}$
$x : a > b > c > d$	$a : z > x > y > w$
$y : a > c > b > d$	$b : y > w > x > z$
$z : c > d > a > b$	$c : w > x > y > z$
$w : c > b > a > d$	$d : x > y > z > w$

Gale-Shapley Proposal Algorithm

Idea: Produces a stable matching using proposals by maintaining information who has proposed to whom and who has rejected whom.
Iteration:

- Each man proposes to the highest woman on his preference list who has not previously rejected him.

Gale-Shapley Proposal Algorithm

Idea: Produces a stable matching using proposals by maintaining information who has proposed to whom and who has rejected whom.
Iteration:

- Each man proposes to the highest woman on his preference list who has not previously rejected him.
- If each woman receives exactly one proposal, stop and use the resulting matching.

Gale-Shapley Proposal Algorithm

Idea: Produces a stable matching using proposals by maintaining information who has proposed to whom and who has rejected whom.
Iteration:

- Each man proposes to the highest woman on his preference list who has not previously rejected him.
- If each woman receives exactly one proposal, stop and use the resulting matching.
- Otherwise, every woman receiving more than one proposal rejects all of them except the one that is highest on her preference list.

Gale-Shapley Proposal Algorithm

Idea: Produces a stable matching using proposals by maintaining information who has proposed to whom and who has rejected whom.
Iteration:

- Each man proposes to the highest woman on his preference list who has not previously rejected him.
- If each woman receives exactly one proposal, stop and use the resulting matching.
- Otherwise, every woman receiving more than one proposal rejects all of them except the one that is highest on her preference list.
- Every woman receiving a proposal says "maybe" to the most attractive proposal received.

Why does this algorithm produce a stable matching?

Outline

Matching and r-factor

A set M of independent edges in a graph $G = (V, E)$ is called a matching.

Matching and r-factor

A set M of independent edges in a graph $G = (V, E)$ is called a matching.

A subgraph of a graph G that contains all vertices of G is called a spanning subgraph of G.

Matching and r-factor

A set M of independent edges in a graph $G = (V, E)$ is called a matching.

A subgraph of a graph G that contains all vertices of G is called a spanning subgraph of G.
A k-regular spanning subgraph is called a k-factor.
(A 1-factor is a matching.)

Matching and r-factor

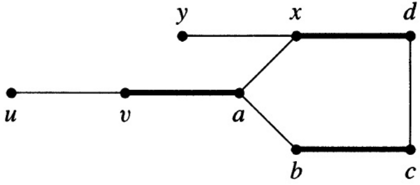
A set M of independent edges in a graph $G = (V, E)$ is called a matching.

A subgraph of a graph G that contains all vertices of G is called a spanning subgraph of G.
A k-regular spanning subgraph is called a k-factor.
(A 1-factor is a matching.)

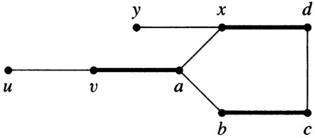
A matching that contains all vertices of a graph G is called a perfect matching (or a 1-factor) of G.

Example

In the example, M contains only the bold edges.
If we search for a shortest M-augmenting path, we observe that u reaches x via a *unsaturated* edge ax.
If we do not consider a longer path reaching x via a saturated edge, then we miss the augmenting path u, v, a, b, c, d, x, y.

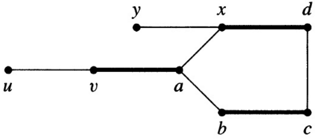


Edmonds Blossom Algorithm



Definition Let M be a matching in a graph G and let u be an M-unsaturated vertex. A **flower** is the union of two M-alternating paths from u that reach a vertex x on steps of opposite parity.

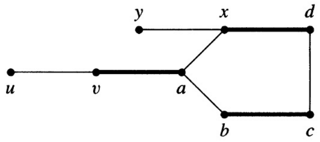
Edmonds Blossom Algorithm



Definition Let M be a matching in a graph G and let u be an M-unsaturated vertex. A **flower** is the union of two M-alternating paths from u that reach a vertex x on steps of opposite parity.

The **stem** of the flower is the maximal common initial path. The blossom of the flower is the odd cycle obtained by deleting the stem. In the example, the path u, v, a is the stem and the blossom is the 5-cycle.

Edmonds Blossom Algorithm

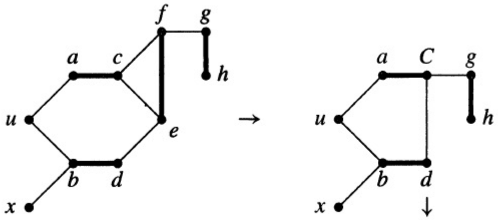


Definition Let M be a matching in a graph G and let u be an M-unsaturated vertex. A **flower** is the union of two M-alternating paths from u that reach a vertex x on steps of opposite parity.

The **stem** of the flower is the maximal common initial path. The blossom of the flower is the odd cycle obtained by deleting the stem. In the example, the path u, v, a is the stem and the blossom is the 5-cycle.

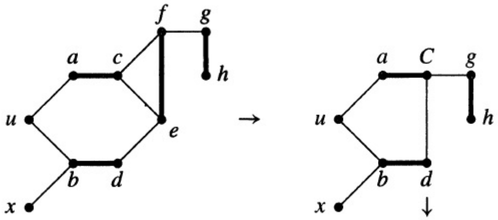
We continue our search along any unsaturated edge from the blossom to a vertex not yet reached (y in the example)
Since each vertex of a blossom is saturated by an edge of M, no saturated edge emerges from a blossom (except the stem).

Edmonds Blossom Algorithm: Example 3.3.16



Consider the blossom as a single "supervertex" and search from all vertices of the supervertex blossom simultaneously along unsaturated edges.

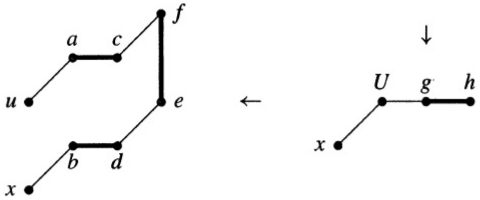
Edmonds Blossom Algorithm: Example 3.3.16

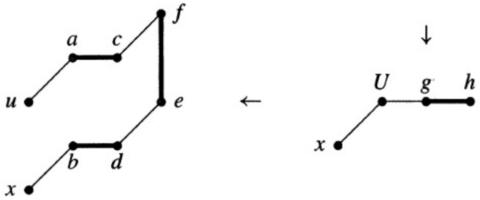


Consider the blossom as a single "supervertex" and search from all vertices of the supervertex blossom simultaneously along unsaturated edges.

By contracting the edges of a blossom B, we obtain a new saturated vertex b incident to the last saturated edge of the stem. Its other incident edges are the unsaturated edges joining vertices of B to the vertices outside B.

Edmonds Blossom Algorithm: Example 3.3.16





By contracting all blossoms like that, we find an M-alternating path in the final graph from u to an unsaturated vertex x, then we can undo the contractions to obtain an M-augmenting path to x.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.
Proof: Exercise.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.
Proof: Exercise.
Corollary (Petersen, 1891): Every regular graph of positive even degree has a 2-factor.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.
Proof: Exercise.
Corollary (Petersen, 1891): Every regular graph of positive even degree has a 2-factor.

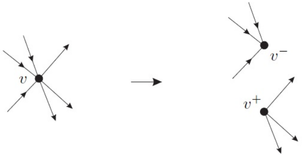


Figure: Splitting vertices in the proof.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.

Proof: Exercise.

Corollary (Petersen, 1891): Every regular graph of positive even degree has a 2-factor.

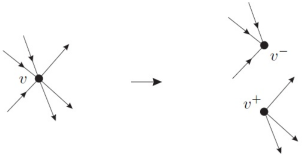


Figure: Splitting vertices in the proof.

- Say G is $2k$-regular. Then G contains an Euler Tour $v_0 e_0 \ldots e_{\ell-1} v_\ell$, with $v_\ell = v_0$.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.
Proof: Exercise.
Corollary (Petersen, 1891): Every regular graph of positive even degree has a 2-factor.

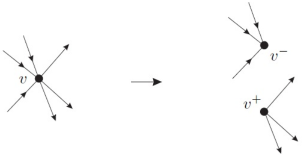


Figure: Splitting vertices in the proof.

- Say G is $2k$-regular. Then G contains an Euler Tour $v_0 e_0 \ldots e_{\ell-1} v_\ell$, with $v_\ell = v_0$.
- Replace every vertex v by a pair (v^-, v^+)
and every edge $e_i = v_i v_{i+1}$ by the edge $v_i^+ v_{i+1}^-$ to obtain a new graph G'.

Corollary: If G is a k-regular bipartite graph with $k \geq 1$, then G has a perfect matching.

Proof: Exercise.

Corollary (Petersen, 1891): Every regular graph of positive even degree has a 2-factor.

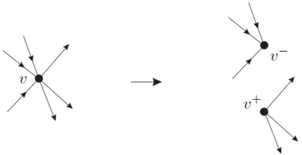


Figure: Splitting vertices in the proof.

- Say G is $2k$-regular. Then G contains an Euler Tour $v_0 e_0 \ldots e_{\ell-1} v_\ell$, with $v_\ell = v_0$.
- Replace every vertex v by a pair (v^-, v^+) and every edge $e_i = v_i v_{i+1}$ by the edge $v_i^+ v_{i+1}^-$ to obtain a new graph G'.
- Since G' is a k-regular bipartite graph, by the previous corollary, G' has a perfect matching (1-factor).

Given a graph G, let us denote by $\mathcal{C}_G$ the set of its components, and by $o(G)$ the number of its odd components, those of odd order.

Given a graph G, let us denote by $\mathcal{C}_G$ the set of its components, and by $o(G)$ the number of its odd components, those of odd order.

Tutte's Condition: If G has a 1-factor, then

$$o(G-S) \le |S| \quad \text{for all } S \subset V(G)$$

since every odd component of $G-S$ will have a factor edge between S and itself.

Given a graph G, let us denote by $\mathcal{C}_G$ the set of its components, and by $o(G)$ the number of its odd components, those of odd order.

Tutte's Condition: If G has a 1-factor, then

$$o(G - S) \leq |S| \quad \text{for all } S \subset V(G)$$

since every odd component of $G - S$ will have a factor edge between S and itself.

Surprisingly, this necessary condition is also sufficient as stated in the theorem below.
Theorem (Tutte, 1947): A graph has a 1-factor if and only if Tutte's condition holds.

Corollary (Berge-Tutte Formula, 1958)

The largest number of vertices saturated by a matching in G is

$$\min_{S \subseteq V(G)} \{n(G) - d(S)\},$$

where $d(S) = o(G - S) - |S|$.

Corollary (Berge-Tutte Formula, 1958)

The largest number of vertices saturated by a matching in G is

$$\min_{S \subseteq V(G)} \{n(G) - d(S)\},$$

where $d(S) = o(G - S) - |S|$.

Proof:

- For every S, at most $|S|$ edges can match vertices of S to the $o(G - S)$ odd components. Thus every matching has at least $o(G - S) - |S|$ unsaturated vertices.

Corollary (Berge-Tutte Formula, 1958)

The largest number of vertices saturated by a matching in G is

$$\min_{S \subseteq V(G)} \{n(G) - d(S)\},$$

where $d(S) = o(G - S) - |S|$.

Proof:

- For every S, at most $|S|$ edges can match vertices of S to the $o(G - S)$ odd components. Thus every matching has at least $o(G - S) - |S|$ unsaturated vertices.
- $$d := \max\{o(G - S) - |S| : S \subseteq V(G)\}.$$

  and let $G' = G \vee K_d$.

Corollary (Berge-Tutte Formula, 1958)

The largest number of vertices saturated by a matching in G is

$$\min_{S \subseteq V(G)} \{n(G) - d(S)\},$$

where $d(S) = o(G - S) - |S|$.

Proof:

- For every S, at most $|S|$ edges can match vertices of S to the $o(G - S)$ odd components. Thus every matching has at least $o(G - S) - |S|$ unsaturated vertices.
- $$d := \max\{o(G - S) - |S| : S \subseteq V(G)\}.$$

  and let $G' = G \vee K_d$.
- Note that $d(S)$ has the same parity as $n(G)$ for each S. Thus, $n(G')$ is even.

Proof (continued):

- If G' satisfies Tutte's condition $o(G' - S') \leq |S'|$ for all S', then we obtain a matching of the desired size in G contained in a matching of G'.

Proof (continued):

- If G' satisfies Tutte's condition $o(G' - S') \leq |S'|$ for all S', then we obtain a matching of the desired size in G contained in a matching of G'.
- If $S' = \emptyset$, Tutte's condn. holds.
  If S' is nonempty, but does not contain all of K_d, then $o(G' - S') \leq 1$, since $G' - S'$ is connected.

Proof (continued):

- If G' satisfies Tutte's condition $o(G' - S') \leq |S'|$ for all S', then we obtain a matching of the desired size in G contained in a matching of G'.
- If $S' = \emptyset$, Tutte's condn. holds.
  If S' is nonempty, but does not contain all of K_d, then $o(G' - S') \leq 1$, since $G' - S'$ is connected.
- Otherwise, S' contains all of K_d. Let $S = S' \cap V(G)$. Then, $o(G' - S') = o(G - S) \leq |S| + d = |S'|$, because Tutte's condn. holds for G. Done, G' has a perfect matching.
- Thus, G has a matching with $n(G) - d$ vertices.

Corollary (Petersen, 1981)

Every bridgeless cubic graph has a 1-factor.

Corollary (Petersen, 1981)

Every bridgeless cubic graph has a 1-factor.

Proof:

- Prove that any bridgeless, cubic graph satisfies Tutte's condition. Given any $S \subset V(G)$, count the edges between S and the odd components of $G - S$.

Corollary (Petersen, 1981)

Every bridgeless cubic graph has a 1-factor.

Proof:

- Prove that any bridgeless, cubic graph satisfies Tutte's condition. Given any $S \subset V(G)$, count the edges between S and the odd components of $G - S$.
- **Observation:** Let m be the number of edges from S to H, where H is an odd component in $G - S$. Since the degree sum of the vertex degrees in H is $3n(H) - m$ and even, m must be odd.

Corollary (Petersen, 1981)

Every bridgeless cubic graph has a 1-factor.

Proof:

- Prove that any bridgeless, cubic graph satisfies Tutte's condition. Given any $S \subset V(G)$, count the edges between S and the odd components of $G - S$.
- **Observation:** Let m be the number of edges from S to H, where H is an odd component in $G - S$. Since the degree sum of the vertex degrees in H is $3n(H) - m$ and even, m must be odd.
- Because G has no bridge (cut-edge), $m \neq 1$. So, $m \geq 3$ and there are at least 3 edges between each odd component of $G - S$ and S.

Corollary (Petersen, 1981)

Every bridgeless cubic graph has a 1-factor.

Proof:

- Prove that any bridgeless, cubic graph satisfies Tutte's condition. Given any $S \subset V(G)$, count the edges between S and the odd components of $G - S$.
- **Observation:** Let m be the number of edges from S to H, where H is an odd component in $G - S$. Since the degree sum of the vertex degrees in H is $3n(H) - m$ and even, m must be odd.
- Because G has no bridge (cut-edge), $m \neq 1$. So, $m \geq 3$ and there are at least 3 edges between each odd component of $G - S$ and S.
- Thus, $3o(G - S) \leq 3|S|$.