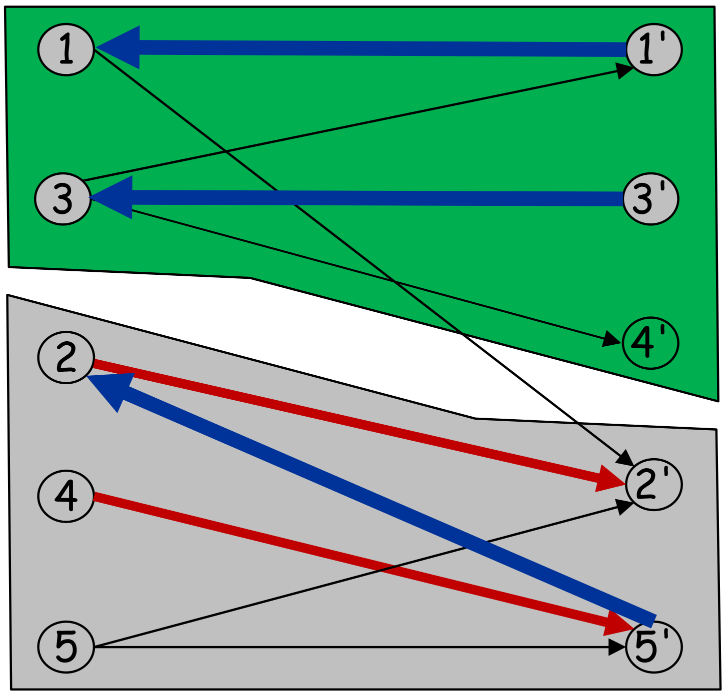
Lecture 2: Maximum Bipartite Matching



Outline

- Problem description
- Alternating and augmenting paths
- Augmenting graph
- Maximum-size bipartite matching algorithm
- Stable set, vertex cover, edge cover
- Matchings covering given vertices

1. Problem Description

Matching, edge cover

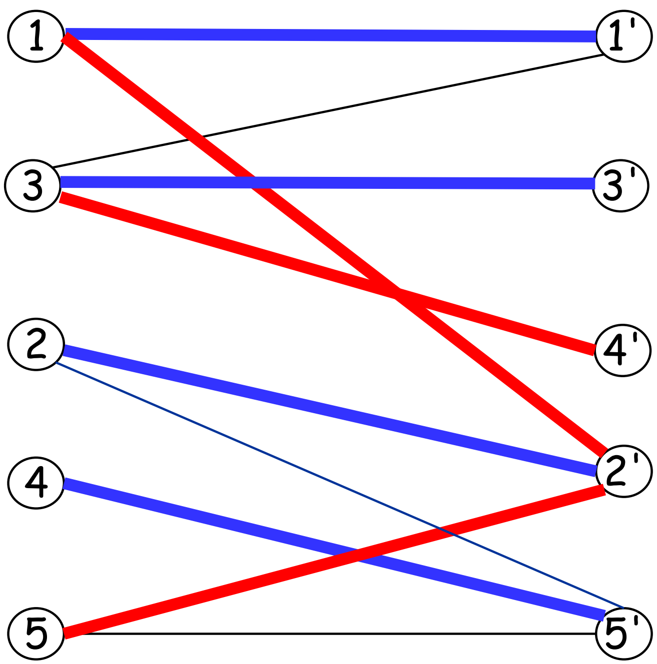
$G = (V, E)$: an undirected graph without isolated vertices

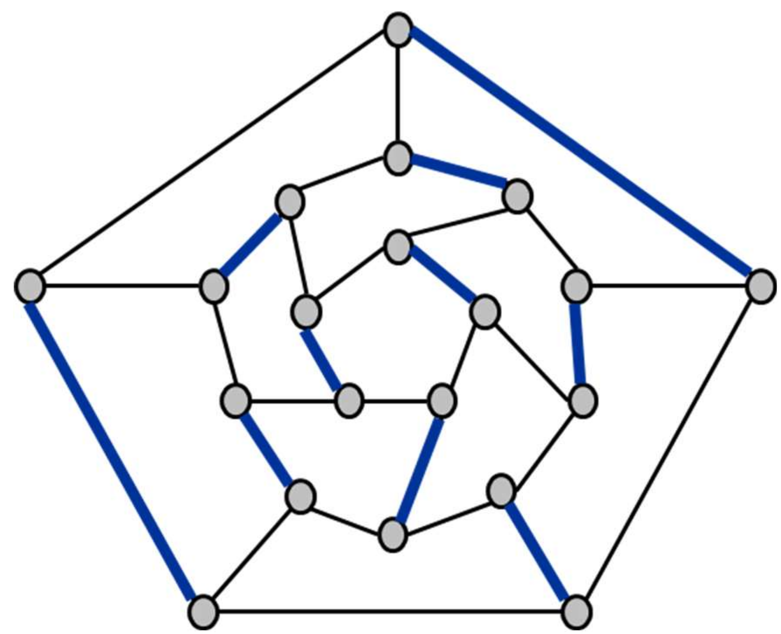
A matching is a subset of node-disjoint edges.

An edge cover is a subset of edges which covers all vertices.

A matching is perfect if it is also an edge cover.

Maximum Matching, Minimum Edge Cover





Stable set, vertex cover

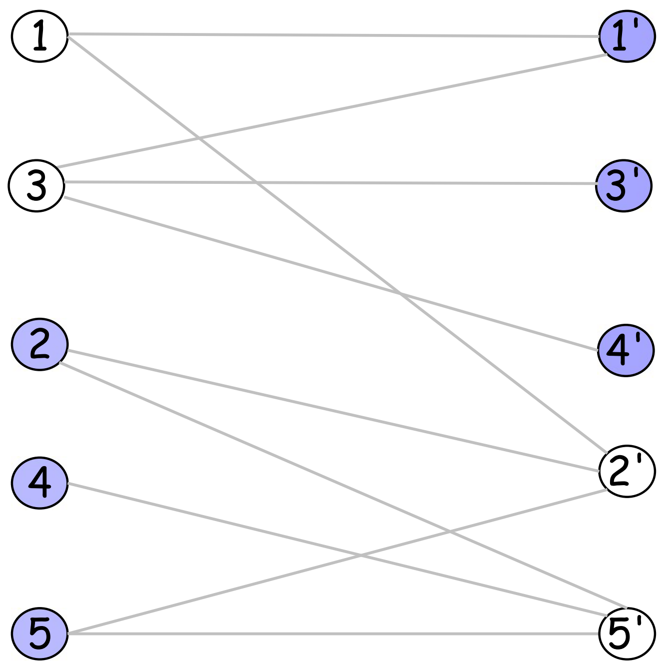
$G = (V, E)$: an undirected graph without isolated vertices

A stable set is a subset of mutually non-adjacent nodes.

A vertex cover is a subset of nodes which covers all edges.

Maximum Stable Set, Minimum Vertex Cover

Observation: I is a stable set $\Leftrightarrow$ $V \setminus I$ is a vertex cover.



Bipartite matching: running time

Application of max flow algorithm

- Generic augmenting path: $O(m \text{ val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.

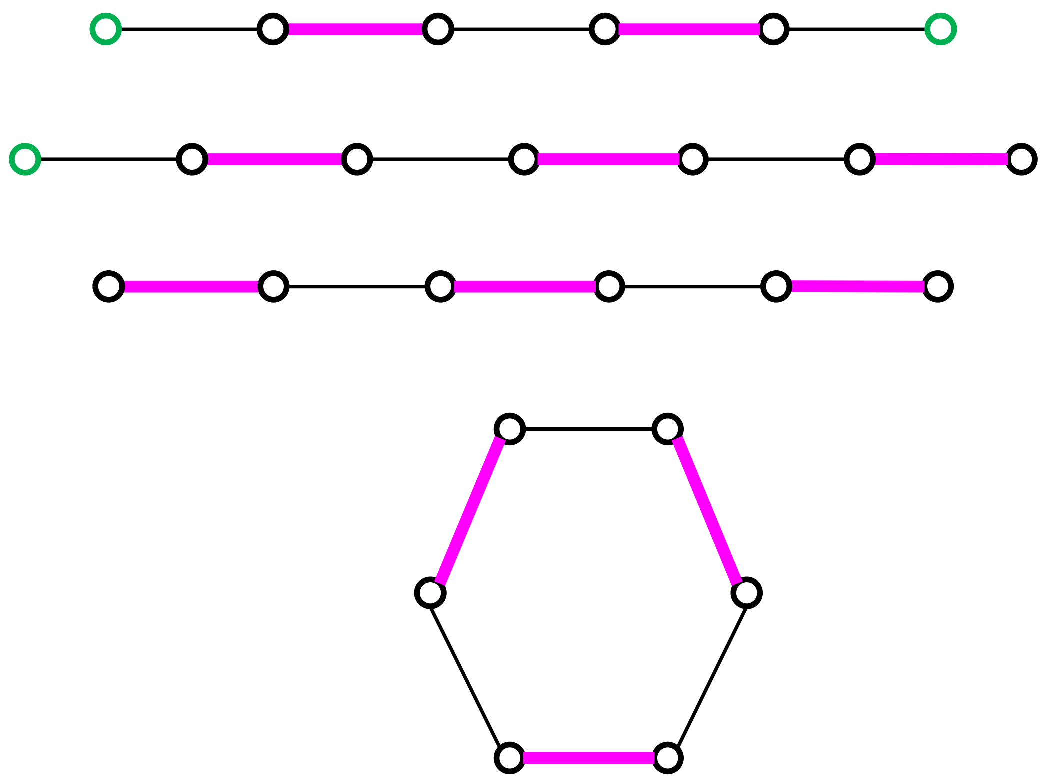
Shortest augmenting path: $O(m\, n^{1/2})$.

Non-bipartite matching.

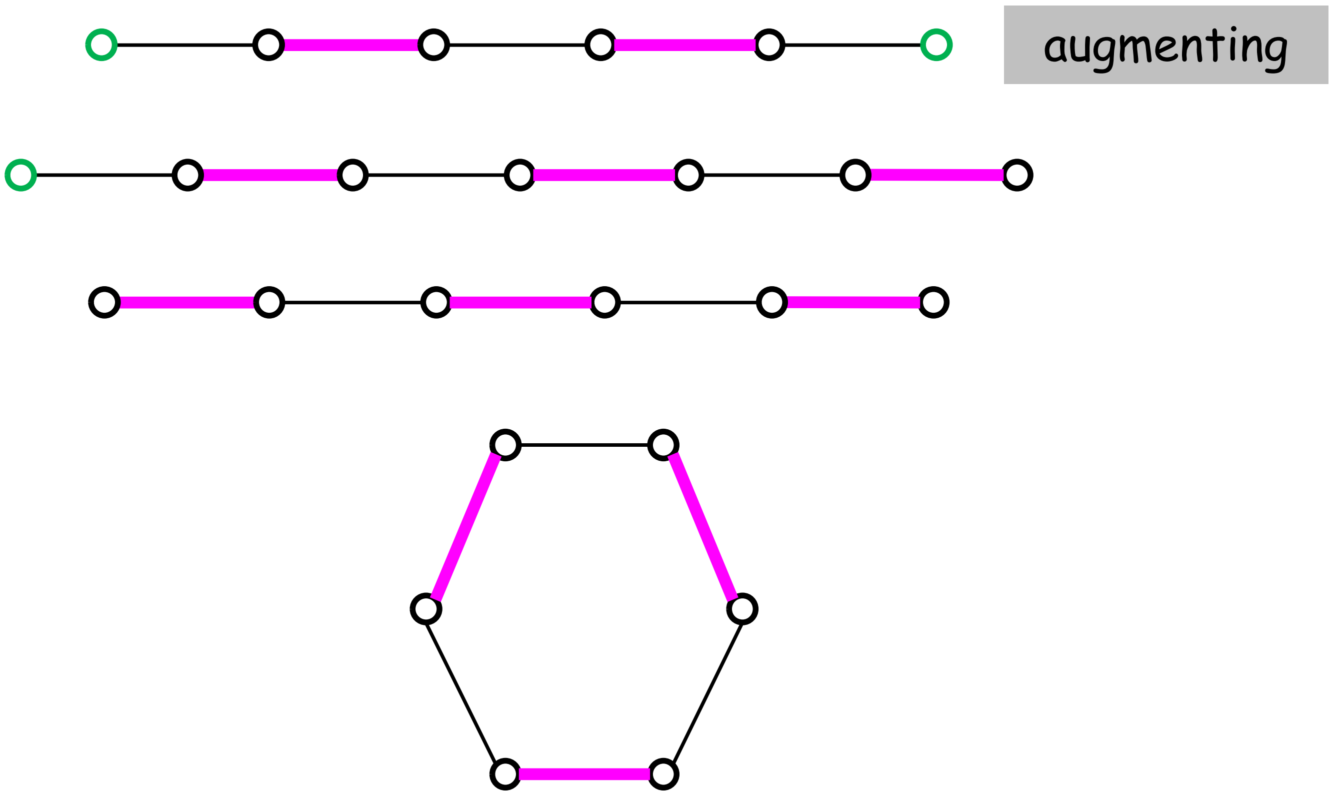
- Structure is more complicated, but well-understood. [Edmonds-Galai]
- Blossom algorithm: $O(n^4)$. [Edmonds 1965]
- Best known: $O(m\, n^{1/2})$??. [Micali-Vazirani 1980]

2. Alternating and Augmenting Paths

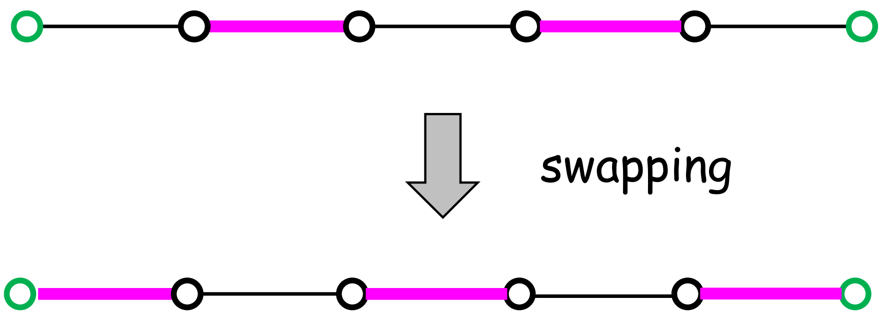
Alternating paths and circuits



Augmenting paths



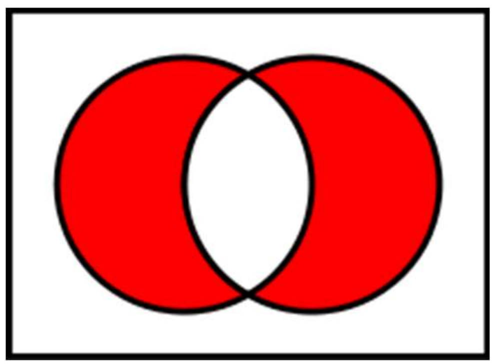
Grow a matching via augmenting paths

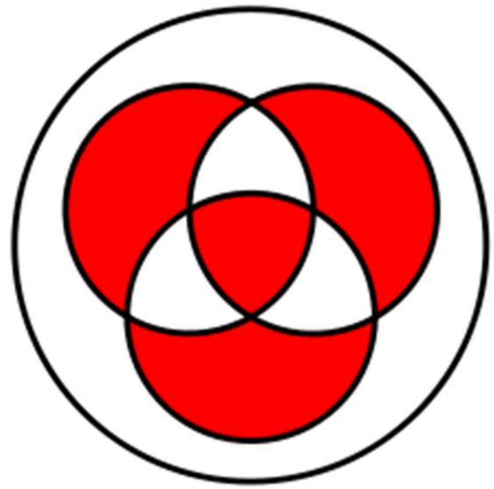


Swap as symmetric difference

Swap $A \cap B$ with $B - A$

Def. $A \oplus B := (A - B) \cup (B - A) = (A \cup B) - (A \cap B)$





parity counting

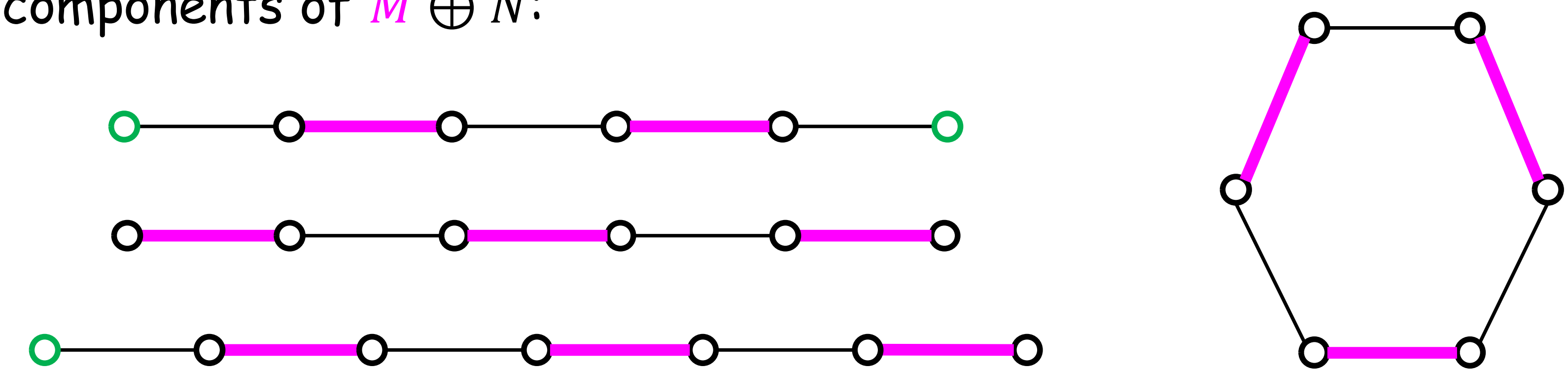
- commutative: $A \oplus B = B \oplus A$
- associative: $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- the empty set is neutral: $A \oplus \emptyset = A$
- every set is its own inverse: $A \oplus A = \emptyset$

Union and symmetric difference of two matchings

components of $M \cup N$: alternating paths and circuits

components of $M \oplus N$ and edge components of $M \cap N$

components of $M \oplus N$:



Thm: If $|N| = |M| + k$ for some $k > 0$, then $M \oplus N$ contains $\geq k$ vertex-disjoint M-augmenting paths.

Pf. Iteratively discard even components, and pairs of M-augmenting path and N -augmenting path. What left are M-augmenting paths.

Augmenting path theorem

Thm: [Petersen (1891)] [Kőnig (1931)] [Berge (1957)]

M is a maximum matching $\Leftrightarrow$ no M-augmenting paths.

Augmenting-path method:

- Start with some initial matching M, possibly the empty one.
- While there is an M-augmenting path P, augment M using P

Challenge: How to choose augmenting path?

Shortest augmenting path

P: a *shortest* M-augmenting path,
Q: an $M \oplus P$-augmenting path

Claim. $|Q| \geq |P| + 2|P \cap Q|$.

Pf. $N = M \oplus P \oplus Q$ is a matching and $|N| = |M| + 2$.

$M \oplus N = P \oplus Q$ contains 2 disjoint M-augmenting paths $P_1,\ P_2$.

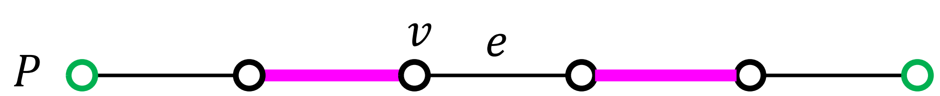
$$|P| + |Q| - 2|P \cap Q| \ = \ |P \oplus Q| = \ |M \oplus N| \geq \ |P_1| + |P_2| \ \geq \ 2|P|$$

Maximal vertex-disjoint shortest augmenting paths

$\mathcal{P}$: a maximal collection of vertex-disjoint *shortest* M-augmenting path
N: the matching obtained by augmenting M along paths in $\mathcal{P}$
Q: an N-augmenting path

Claim. $|Q| >$ lengths of paths in $\mathcal{P}$

Pf. Trivial if Q is vertex-disjoint from all paths in $\mathcal{P}$. Otherwise, Q shares a vertex v with some path P in $\mathcal{P}$. For the unique edge $e \in P \backslash \mathrm{M}$ in incident to v, $e \in N$ hence $e \in Q$. So $|Q| \geq |P| + 2$.



3. Augmenting Graph

Augmenting graph of a matching

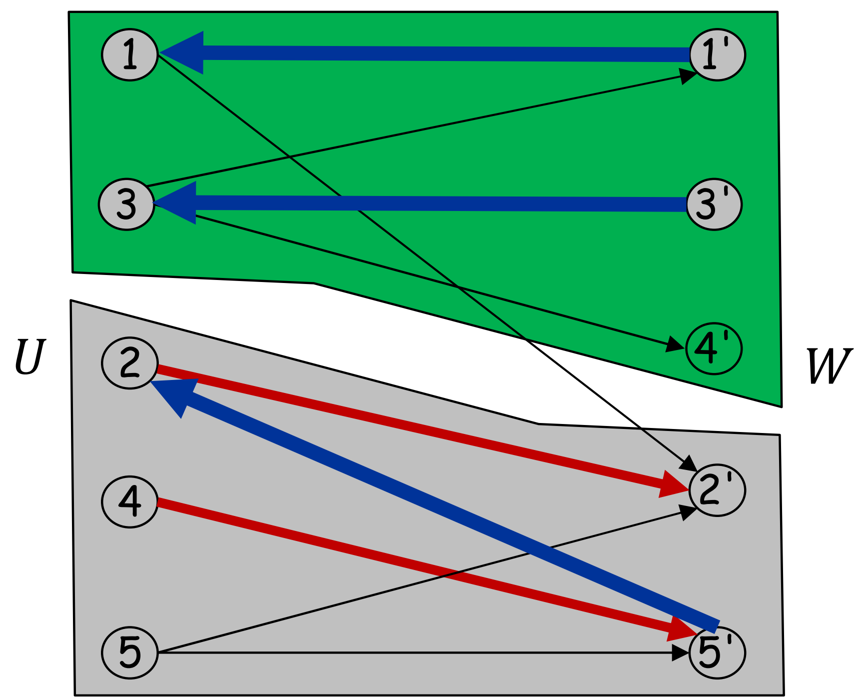
D_M: edges in M are oriented from W to U, others from U to W

U_M, W_M: unmatched vertices in U and W resp.

- $U_M - W_M$ paths $\Leftrightarrow$ M-augmenting paths

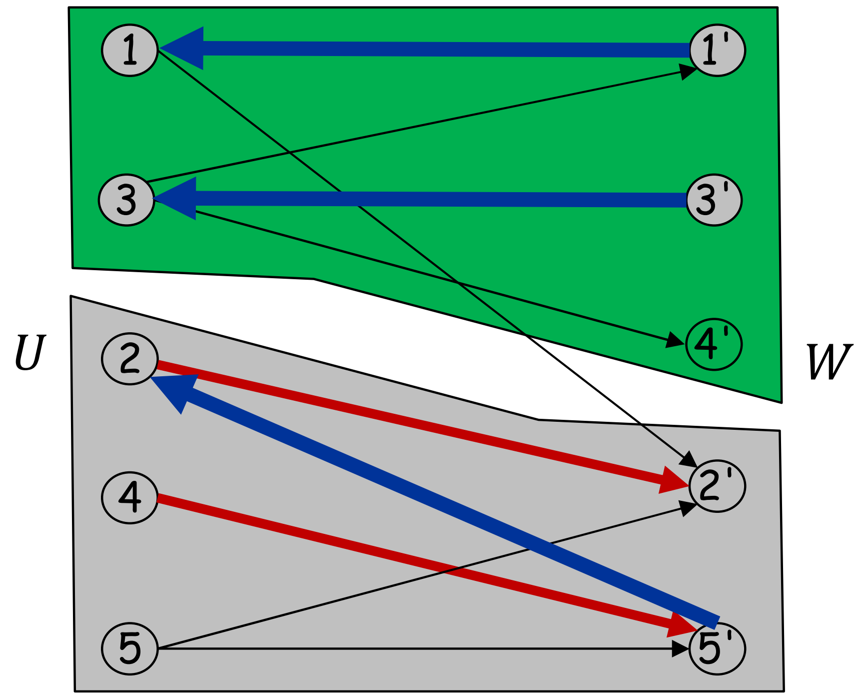
R_M: vertices reachable from U_M

- $R_M \cap W_M = \emptyset \Leftrightarrow M$ is maximum



Essential properties

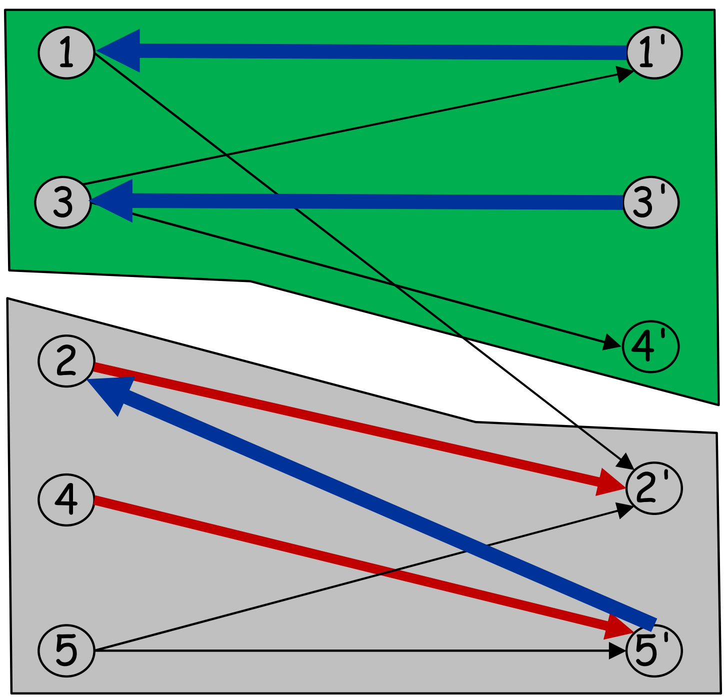
- Each matched vertex in U (resp. W) has exactly one in-neighbor (resp. out-neighbor).
- No arc of D_M leaving R_M, and no edge in M across R_M
- $I := (U \cap R_M) \cup (W \setminus R_M)$ is a stable set
- $C := (U \setminus R_M) \cup (W \cap R_M)$ is a vertex cover

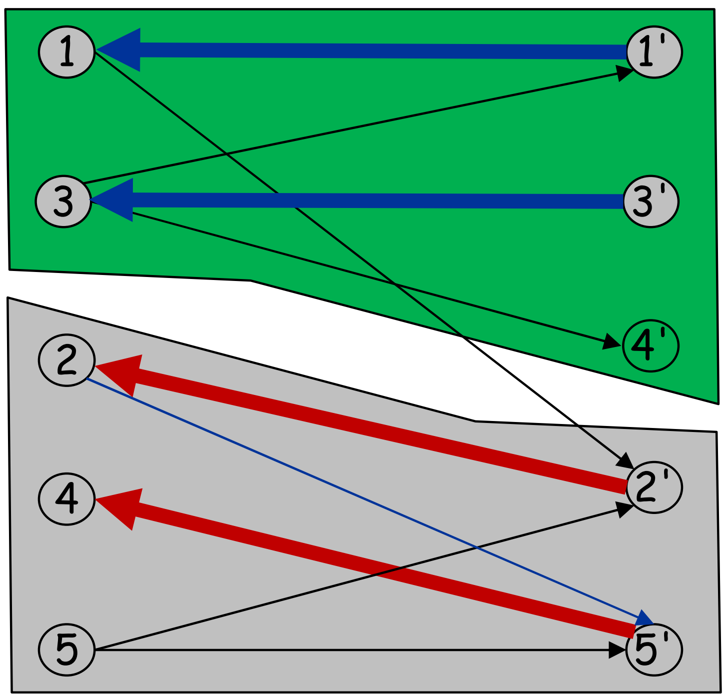


Evolution of the augmenting graph

P: an M-augmenting path; $N = M \oplus P$

- D_N : obtained from D_M by reversing all the arcs in P
- $U_N \subset U_M, W_N \subset W_M$
- $R_N \subseteq R_M$





4. Augmenting-Path Algorithm

Basic algorithm

[van der Waerden 1927] and [Konig 1931]

$M \leftarrow \emptyset$;
repeat
    find a path P (if any) in D_M from U_M to W_M;
    if P is not found, return M;
    $M \leftarrow M \oplus P$;

Analysis. Θ(mn) time: O(n) augmentations, each taking O(m) time

Hopcroft-karp Algorithm

[Hopcroft and Karp 1971,1973]

$M \leftarrow \emptyset$;
repeat
    find a maximal collection of vertex-disjoint
        shortest paths in D_M from U_M to W_M;
    if P is not found, return M;
    augment M along these paths

Analysis. $\Theta(mn^{1/2})$ time: $O(n^{1/2})$ augmentations, each taking $O(m)$ time

Optimality gap

Claim: After each iteration, the length of shortest augmenting paths increases by at least two.

α' := the size of a maximum matching N

Claim: After k iterations, $\alpha' - |M| \leq \alpha'/(k+1)$

Pf. Any M-augmenting path has length $\geq 2k+1$.
$M \oplus N$ contains $|N| - |M|$ vertex-disjoint M-augmenting paths

$$(2k+1)(|N| - |M|) \leq |M \oplus N| \leq |M| + |N|$$

$$(2k)|N| \leq (2k+2)|M|$$

$$|M| \geq \alpha' k/(k+1)$$

Number of iterations

- After $k := \lfloor (\alpha')^{1/2} \rfloor$ iterations,
$$\alpha' - |M| \leq \alpha'/(k+1) \leq \alpha'/\lceil (\alpha')^{1/2} \rceil \leq (\alpha')^{1/2}$$
hence
$$\alpha' - |M| \leq \lfloor (\alpha')^{1/2} \rfloor = k$$

- # of additional iterations $\leq \alpha' - |M| \leq k$

- Total # of iterations $\leq 2k$

4. Stable set, Vertex Cover, Edge Cover

Stable set, vertex cover

$G = (V, E)$: an undirected graph without isolated vertices

Matching number $\alpha' :=$ the maximum size of all matchings
Edge cover number $\tau' :=$ the minimum size of all edge covers
Stable set number $\alpha :=$ the maximum size of all stable sets
Vertex cover number $\tau :=$ the minimum size of all vertex covers

Observation: $\alpha + \tau = |V|$, $\tau \geq \alpha'$, $\tau' \geq \alpha$.

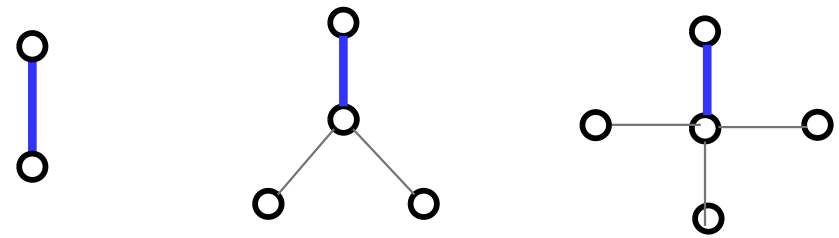
Gallai's theorem [1959]. $\alpha' + \tau' = |V|$

König's Theorem [1931]. If G is bipartite, then $\tau = \alpha'$ and $\tau' = \alpha$.

Algorithmic proof of Gallai's Theorem

Matching ⇒ Edge Cover: Let M be a maximum matching. Obtain an edge cover F by adding to M an edge covering each of the $|V| - 2|M|$ vertices missed by M. Then

$$|F| = |M| + (|V| - 2|M|) = |V| - |M|.$$



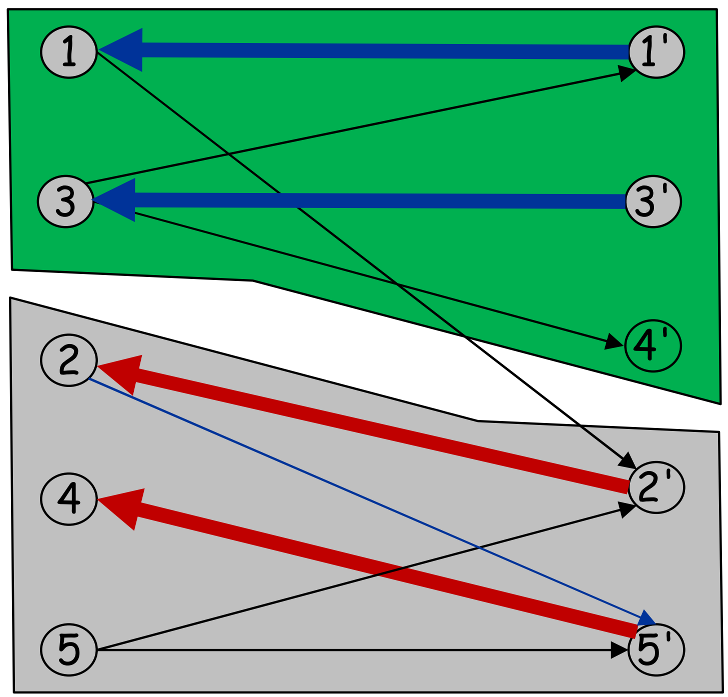
Edge Cover ⇒ Matching: Let F be a minimum edge cover. Then (V, F) is a forest of stars. Obtain a matching M by taking an edge from each of the stars of (V, F). Then

$$|M| = |V| - |F|.$$

Algorithmic proof of König's Theorem

For a maximum matching M in G,

- $C := (U \setminus R_M) \cup (W \cap R_M)$ is a minimum vertex cover, and $|C| = |M|$;
- $I := (U \cap R_M) \cup (W \setminus R_M)$ is a maximum stable set, and $|I| = n - |M|$.



6. Matchings Covering Given Vertices

Given vertices at a single side

$G = (U \cup W, E)$:

Notation. Let S be a subset of nodes, and $N(S)$ be the set of nodes adjacent to S.

Observation. If G has a matching covering $S \subseteq U$, then $|N(S)| \geq |S|$.

Pf. Each node in S has to be matched to a different node in $N(S)$.

Marriage Theorem

Thm. [Frobenius 1917, Hall 1935] G has a matching covering U
$\Leftrightarrow |N(S)| \geq |S|$ for each $S \subseteq U$.

Pf. $\Rightarrow$ trivial

$\Leftarrow$ For each vertex cover C, $N(U \setminus C) \subseteq C \cap W$, and hence

$$|C| = |C \cap U| + |C \cap W| \geq |C \cap U| + |N(U \setminus C)| \geq |C \cap U| + |U \setminus C| = |U|$$

Hall set

Def. A set $S \subseteq U$ minimizing $|N(S)| - |S|$ is called a Hall set.

Fact. $|N(S)| + |N(T)| \geq |N(S \cup T)| + |N(S \cap T)|$

Pf.

$$N(S \cup T) = N(S) \cup N(T)$$
$$N(S \cap T) \subseteq N(S) \cap N(T)$$

$$|N(S)| + |N(T)| = |N(S) \cup N(T)| + |N(S) \cap N(T)| \geq |N(S \cup T)| + |N(S \cap T)|$$

Submodular functions

Def. A set function ρ on V is said to be *submodular* if

$$\rho(X) + \rho(Y) \geq \rho(X \cap Y) + \rho(X \cup Y).$$

Def. For $S \subseteq V$, a set $X \subseteq S$ is said to be a *minimizer* of ρ on S if

$$\rho(X) = \min_{Y \subseteq S} \rho(Y).$$

Fact: Suppose both X and Y are minimizers ρ on S. So are $X \cap Y$ and $X \cup Y$.

$$\rho(X) + \rho(Y) \geq \rho(X \cap Y) + \rho(X \cup Y) \geq \rho(X) + \rho(Y).$$

Fact: $\forall S \subseteq V$, ρ has a unique *minimal* (resp. *maximal*) minimizer on S.

All can be computed in strongly polynomial time.

Given vertices at both sides

$R \subseteq V = U \cup W$: required vertices

Thm [Mendelsohn-Dulmage 1958] G has a matching covering $R \Leftrightarrow$ G has a matching M covering $R \cap U$ and a matching N covering $R \cap W$.

Pf. $\Rightarrow$ trivial

$\Leftarrow$ In each component of $M \cup N$, all R-vertices can be covered by either M or N. Otherwise, the component is an alternating even path between two R-vertices on the same side, say on the U side. But then one end of the path is not covered by $M \cap U$.

Thm. G has a matching covering $R \Leftrightarrow |N(S)| \geq |S|$ for each $S \subseteq R \cap U$ and for each $S \subseteq R \cap W$.

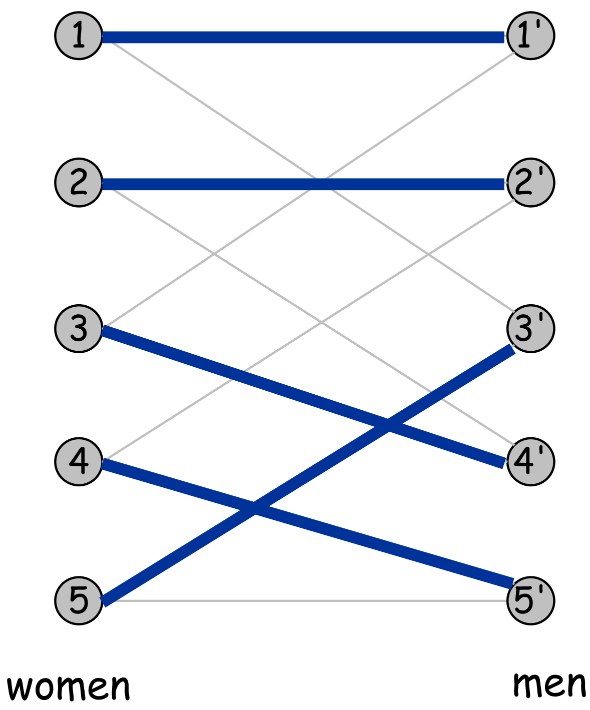
k-Regular bipartite graphs

Dancing problem.

- Exclusive Ivy league party attended by n men and n women.
- Each man knows exactly k women; each woman knows exactly k men.
- Acquaintances are mutual.
- Is it possible to arrange a dance so that each woman dances with a different man that she knows?

Mathematical reformulation. Does every k-regular bipartite graph have a perfect matching?

Ex. Boolean hypercube.



k-Regular bipartite graphs

Thm. [König 1916, Frobenius 1917] Every k-regular bipartite graph with $k > 0$ has a perfect matching.

Pf. $G = (V, E)$: a k-regular bipartite graph
Each vertex cover has size at least $|V|/2$: there are $k|V|/2$ edges and each vertex covers k edge.

Thm. Each k-regular bipartite graph is the union of k perfect matchings.

Faster algorithms in regular bipartite graphs

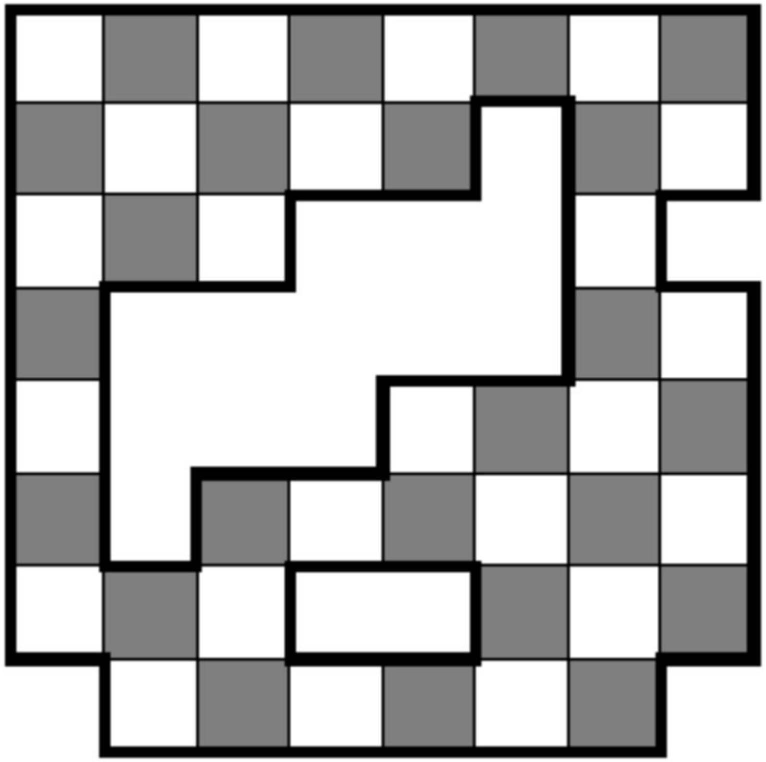
- A perfect matching in a regular bipartite graph can be computed in $O(m)$ time (Corollary 16.11a)

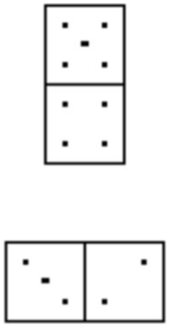
- A decomposition of a k-regular bipartite graph into k perfect matchings can be computed in $O(m \log k)$ time (Theorem 20.14)

Summary

- Augmenting paths
- Augmenting graph
- Hopcroft-Karp algorithm
- Stable set, vertex cover, edge cover
- Matchings covering given vertices

Filling a partial chessboard with dominos





dominos

Completion of partial Latin squares

1	2	3	4
3	4	2	1
2	1	4	3
4	3	1	2

2	4	5	3	1
4	1	3	2	5
3	2	1	5	4