Recall: Matchings

A matching is a set of (non-loop) edges with no shared endpoints. The vertices incident to an edge of a matching M are saturated by M, the others are unsaturated. A perfect matching of G is matching which saturates all the vertices.

Examples. $K_{n,m}$, K_n, Petersen graph, Q_k; graphs without perfect matching

A maximal matching cannot be enlarged by adding another edge.

A maximum matching of G is one of maximum size.

Example. Maximum $\neq$ Maximal

Recall: Characterization of maximum matchings

Let M be a matching. A path that alternates between edges in M and edges not in M is called an M-alternating path.
An M-alternating path whose endpoints are unsaturated by M is called an M-augmenting path.

Theorem(Berge, 1957) A matching M is a maximum matching of graph G iff G has no M-augmenting path.

Proof. ($\Rightarrow$) Easy.
($\Leftarrow$) Suppose there is no M-augmenting path and let M^* be a matching of maximum size.
What is then $M \triangle M^*$???

> **Lemma** Let M_1 and M_2 be matchings of G. Then each connected component of $M_1 \triangle M_2$ is a path or an even cycle.
>
> For two sets A and B, the symmetric difference is $A \triangle B = (A \setminus B) \cup (B \setminus A)$.

Recall: Hall's Condition and consequences

Theorem (Marriage Theorem; Hall, 1935) Let G be a bipartite (multi)graph with partite sets X and Y. Then there is a matching in G saturating X iff $|N(S)| \geq |S|$ for every $S \subseteq X$.

Proof. ($\Rightarrow$) Easy.

($\Leftarrow$) Not *so* easy. Find an M-augmenting path for *any* matching M which does not saturate X.
(Let U be the M-unsaturated vertices in X. Define

$$\begin{aligned} T &:= \{y \in Y : \exists\ M\text{-alternating } U, y\text{-path}\}, \\ S &:= \{x \in X : \exists\ M\text{-alternating } U, x\text{-path}\}. \end{aligned}$$

Unless there is an M-augmenting path, $S \cup U$ **violates** Hall's condition.)

Corollary. (Frobenius (1917)) For $k > 0$, every k-regular bipartite (multi)graph has a perfect matching.

Recall: Application: 2-Factors

A factor of a graph is a spanning subgraph. A k-factor is a spanning k-regular subgraph.

Every regular bipartite graph has a 1-factor.

Not every regular graph has a 1-factor.

But...

Theorem. (Petersen, 1891) Every $2k$-regular graph has a 2-factor.

Proof. Use Eulerian cycle of G to create an auxiliary k-regular bipartite graph H, such that a perfect matching in H corresponds to a 2-factor in G.

Recall: Graph parameters

$\alpha'(G)$ = size of the largest matching in G

A vertex cover of G is a set $Q \subseteq V(G)$ that contains at least one endpoint of every edge.
$\beta(G)$ = size of the smallest vertex cover in G

Claim. For every graph G, $\beta(G) \geq \alpha'(G)$.

Theorem. (König (1931), Egerváry (1931))
If G is bipartite then $\beta(G) = \alpha'(G)$.

Proof of König's Theorem: For any minimum vertex cover Q, apply Hall's Condition to match $Q \cap X$ into $Y \setminus Q$ and $Q \cap Y$ into $X \setminus Q$.

Remarks

1. König's Theorem $\Rightarrow$ For bipartite graphs there always exists a vertex cover **proving** that a particular matching of maximum size is really maximum.

2. This is NOT the case for general graphs: C_5.

How to find a maximum matching in bipartite graphs?

Augmenting Path Algorithm

Input. A bipartite graph G with partite sets X and Y, a matching M in G.

Output. EITHER an M-augmenting path OR a certificate (a cover of the same size) that M is maximum.

Idea. Let U be set of unsaturated vertices in X.
Explore M-alternating paths from U, letting $S \subseteq X$ and $T \subseteq Y$ be the sets of vertices reached.
As a vertex is reached, record the previous vertex on the M-alternating path from which it was reached.
Mark vertices of S that have been fully explored for path extensions (say, put them into a set Q).

Initialization. $S = U$, $Q = \emptyset$, and $T = \emptyset$.

Iteration.

IF $Q = S$ THEN

  stop and **report** that M is a maximum matching and $T \cup (X \setminus S)$, is a cover of the same size.

ELSE

  select $x \in S \setminus Q$ and

  FORALL $y \in N(x)$ with $xy \notin M$ DO

    IF y is unsaturated, THEN

      stop and **report** an M-augmenting path from U to y.

    ELSE

      $\exists w \in X$ with $yw \in M$. **Update**

        $T := T \cup \{y\}$ (y is reached from x),

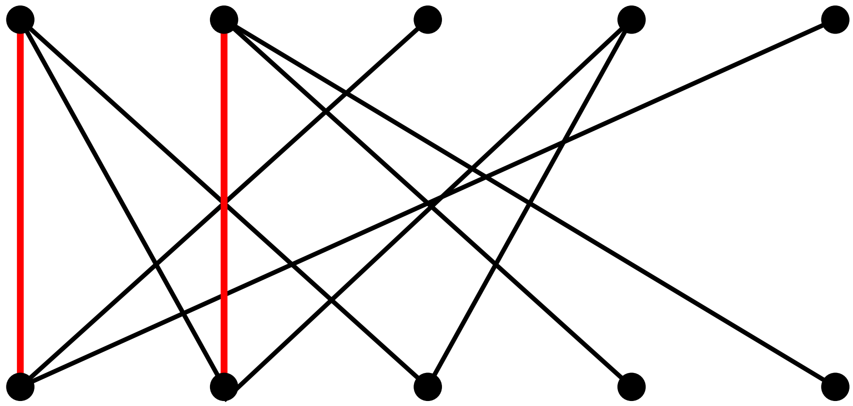
        $S := S \cup \{w\}$ (w is reached from y).

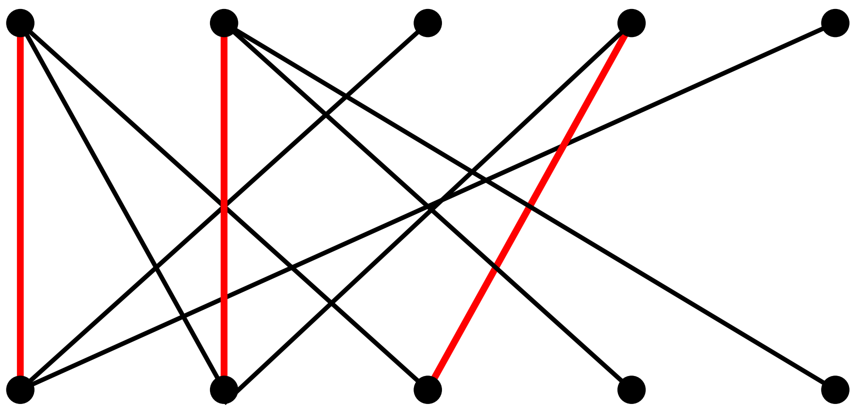
  update $Q := Q \cup \{x\}$

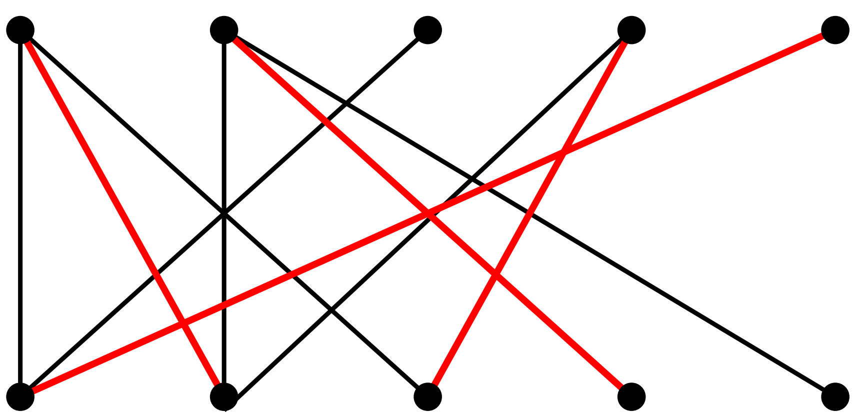
iterate.

Theorem. Repeatadly applying the Augmenting Path Algorithm to a bipartite graph produces a maximum matching and a minimum vertex cover.

If G has n vertices and m edges, then this algorithm finds a maximum matching in $O(nm)$ time.







Proof of correctness

If Augmenting Path Algorithm does what it supposed to, then after at most $n/2$ application we can produce a maximum matching.
Why does the APA terminate? It touches each edge at most once. Hence running time is $O(nm)$.

What if an M-augmenting path is returned? It is OK, since y is an unsaturated neighbor of $x \in S$, and x can be reached from U on an M-alternating path.

What if the APA returns M as maximum matching and $T \cup (X \setminus S)$ as minimum cover?

Since $S = Q$, all edges leaving S were explored, so there is **no edge between S and $Y \setminus T$**.

- Hence $T \cup (X \setminus S)$ is indeed a cover.
- $|M| = |T| + |X \setminus S|$ (By selection of S and T.)

Key Lemma If, in any graph, a cover and a matching have the same size, then they are both optimal.

$$|M| \leq \alpha'(G) \leq \beta(G) \leq |T \cup (X \setminus S)| = |M|.$$

How to find a maximum weight matching in a bipartite graph?

In the maximum weighted matching problem a non-negative weight $w_{i,j}$ is assigned to each edge $x_i y_j$ of $K_{n,n}$ and we seek a perfect matching M to maximize the total weight $w(M) = \sum_{e \in M} w(e)$.

With these weights, a (weighted) cover is a choice of labels $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$, such that $u_i + v_j \geq w_{i,j}$ for all i, j. The cost $c(u, v)$ of a cover (u, v) is $\sum u_i + \sum v_j$. The minimum weighted cover problem is that of finding a cover of minimum cost.

Duality Lemma For a perfect matching M and a weighted cover (u, v) in a bipartite graph G, $c(u, v) \geq w(M)$. Also, $c(u, v) = w(M)$ iff M consists of edges $x_i y_{\pi(i)}$ such that $u_i + v_{\pi(i)} = w_{i,\pi(i)}$ for some permutation $\pi \in S_n$. In this case, M and (u, v) are both optimal.

The algorithm

The equality subgraph $G_{u,v}$ for a weighted cover (u, v) is the spanning subgraph of $K_{n,n}$ whose edges are the pairs $x_i y_j$ such that $u_i + v_j = w_{i,j}$. In the cover, the excess for i, j is $u_i + v_j - w_{i,j}$.

Hungarian Algorithm

Input. A matrix $(w_{i,j})$ of weights on the edges of $K_{n.n}$ with partite sets X and Y.

Idea. Iteratively adjusting a cover (u, v) until the equality subgraph $G_{u,v}$ has a perfect matching.

Initialization. Let $u_i = \max\{w_{i,j} : j = 1, \ldots, n\}$ and $v_j = 0$.

Iteration.
Form $G_{u,v}$ and use APA to find a maximum matching M and minimum vertex cover $Q = T \cup R$, where $R = X \cap Q$ and $T = Y \cap Q$.
IF M is a perfect matching, THEN
    stop and **report** M as a maximum weight matching and (u, v) as a minimum cost cover
ELSE
    $\epsilon :=$ min$\{u_i + v_j - w_{i,j} : x_i \in X \setminus R, y_j \in Y \setminus T\}$
    Update u and v:
        $u_i := u_i - \epsilon$ if $x_i \in X \setminus R$
        $v_j := v_j + \epsilon$ if $y_j \in T$
Iterate

Remarks. By properties of APA:

- $|Q| = |M|$, no M-edge is covered by twice by Q
- $T = \{y \in Y$: there is an M-alternating (U, y)-path$\}$
- $R = \{x \in X$: there is NO M-alternating (U, x)-path$\}$ where $U = \{x \in X$: x is M-unsaturated$\}$.

Theorem The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover.

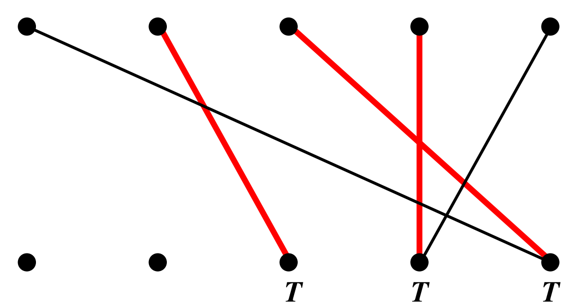
The Assignment Problem — An example

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 7 & 2 \\ 1 & 3 & 4 & 4 & 5 \\ 3 & 6 & 2 & 8 & 7 \\ 4 & 1 & 3 & 5 & 4 \end{pmatrix}$$

Excess Matrix

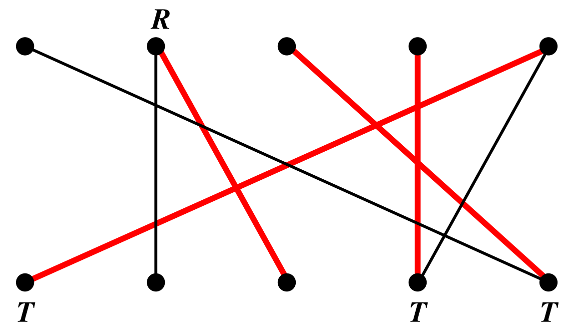
$$\begin{array}{cc} & \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \\ \begin{array}{c} 5 \\ 8 \\ 5 \\ 8 \\ 5 \end{array} & \begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 & 6 \\ 4 & 2 & 1 & 1 & 0 \\ 5 & 2 & 6 & 0 & 1 \\ 1 & 4 & 2 & 0 & 1 \end{pmatrix} \\ & \begin{array}{ccccc} & & T & T & T \end{array} \end{array}$$

Equality Subgraph



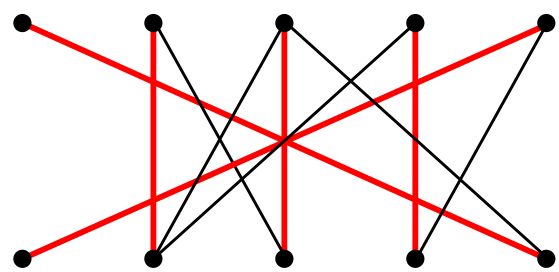
$\epsilon = 1$

$$\begin{array}{c c c}
 & \begin{matrix} 0 & 0 & 1 & 1 & 1 \end{matrix} & \\
\begin{matrix} 4 \\ 7 \\ 4 \\ 7 \\ 4 \end{matrix} & \begin{pmatrix} 3 & 2 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 & 6 \\ 3 & 1 & 1 & 1 & 0 \\ 4 & 1 & 6 & 0 & 1 \\ 0 & 3 & 2 & 0 & 1 \end{pmatrix} & \begin{matrix} \\ R \\ \\ \\ \\ \end{matrix} \\
 & \begin{matrix} T & & & T & T \end{matrix} &
\end{array}$$



$$\epsilon = 1$$

$$\begin{array}{c c}
 & \begin{matrix} 1 & 0 & 1 & 2 & 2 \end{matrix} \\
\begin{matrix} 3 \\ 7 \\ 3 \\ 6 \\ 3 \end{matrix} & \begin{pmatrix} 3 & 1 & 1 & 1 & 0 \\ 2 & 0 & 0 & 2 & 7 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 5 & 0 & 1 \\ 0 & 2 & 1 & 0 & 1 \end{pmatrix}
\end{array}$$



DONE!!

The Duality Lemma states that if $w(M) = c(u, v)$ for some cover (u, v), then M is maximum weight.

We found a maximum weight matching (transversal). The fact that it is maximum is certified by the indicated cover, which has the same cost:

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 0 & 1 & 2 & 2 \end{array} \\ \begin{array}{c} 3 \\ 7 \\ 3 \\ 6 \\ 3 \end{array} & \begin{pmatrix} 1 & 2 & 3 & 4 & \mathbf{5} \\ 6 & \mathbf{7} & 8 & 7 & 2 \\ 1 & 3 & \mathbf{4} & 4 & 5 \\ 3 & 6 & 2 & \mathbf{8} & 7 \\ \mathbf{4} & 1 & 3 & 5 & 4 \end{pmatrix} \end{array}$$

$$\begin{aligned} w(M) &= 5 + 7 + 4 + 8 + 4 = 28 = \\ &= 1 + 0 + 1 + 2 + 2 + \\ &\quad\ 3 + 7 + 3 + 6 + 3 = c(u, v) \end{aligned}$$

Hungarian Algorithm — Proof of correctness

Proof. If the algorithm ever terminates and $G_{u,v}$ is the equality subgraph of a (u, v), which is indeed a cover, then M is a m.w.m. and (u, v) is a m.c.c. by Duality Lemma.

Why is (u, v), created by the iteration, a cover?

Let $x_i y_j \in E(K_{n,n})$. Check the four cases.

$x_i \in R,$	$y_j \in Y \setminus T$	$\Rightarrow$	u_i and v_j do not change.
$x_i \in R,$	$y_j \in T$	$\Rightarrow$	u_i does not change v_j increases.
$x_i \in X \setminus R,$	$y_j \in T$	$\Rightarrow$	u_i decreases by ϵ, v_j increases by ϵ.
$x_i \in X \setminus R,$	$y_j \in Y \setminus T$	$\Rightarrow$	$u_i + v_j \geq w_{i,j}$ by definition of ϵ.

Why does the algorithm terminate?

- $\#$ of vertices reached from U by M-alternating paths grows

  (only edges between S and T can become non-edges during an iteration and these do not participate in such paths.)

- after $\leq n$ iteration an M-unsaturated $y \in Y$ is reached with a (U, y)-augmenting path

- max matching gets larger; can happen $\leq n$-times

- after $\leq n^2$ iteration $G_{u,v}$ has perfect matching