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,
the set U of unsaturated vertices in X.

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

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.

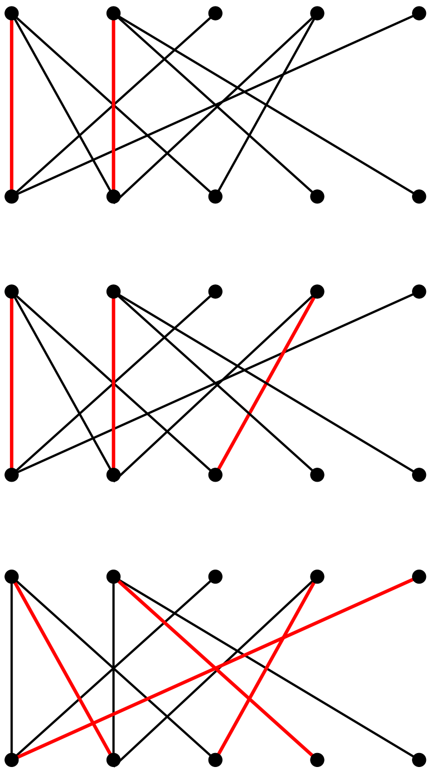
Initialization. $S = U$ and $T = \emptyset$.

Iteration.
IF all vertices in S are marked 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 an unmarked $x \in S$ and explore its neighbors $y \in N(x)$, for which $xy \notin M$.
- 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).
- After exploring all neighbors of x, mark x and **iterate**.

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

Theorem. The running time of the algorithm is $O(nm)$ where n is the number of vertices and m is the number of edges of G.



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_iy_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_iy_j such that $u_i + v_j = w_{i,j}$. 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 $u_i = \max\{w_{i,j} : j = 1,\dots,n\}$ and $v_j = 0$.

Iteration.

Form $G_{u,v}$ and find a maximum matching M in it.
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
  let Q be a vertex cover of size $|M|$ in $G_{u,v}$.
  $R := X \cap Q$
  $T := Y \cap Q$
  $\epsilon := \min\{u_i + v_j - w_{i,j} : x_i \in X \setminus R, y_j \in Y \setminus T\}$
  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

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

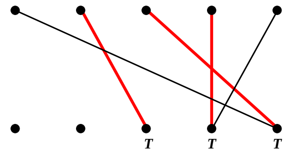
The Assignment Problem — An example

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

Excess Matrix

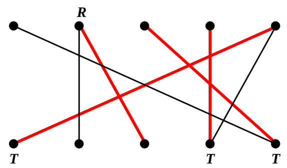
$$\begin{array}{c} \\ 5 \\ 8 \\ 5 \\ 8 \\ 5 \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \\ \begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 & 6 \\ 4 & 2 & 1 & 1 & 0 \\ 5 & 2 & 6 & 0 & 1 \\ 1 & 4 & 2 & 0 & 1 \end{pmatrix} \\ \begin{array}{ccccc} & & T & T &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

Equality Subgraph



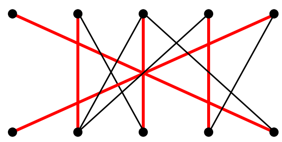
$\epsilon = 1$

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



$\epsilon = 1$

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



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:

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

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

Stable Matchings

Bonnie and Clyde is called an unstable pair if

- Bonnie and Clyde are currently not a couple,
- Bonnie prefers Clyde to her current partner, and
- Clyde prefers Bonnie to his current partner.

A perfect matching (of n woman and n man) is a stable matching if it yields no unstable pair.

Theorem. (Gale-Shapley, 1962) There exists a divorce-free society. More precisely: For any preference rankings of n man and n woman there is a stable matching.

Proof. Algorithmic.

The proof of divorce-free society

Proposal Algorithm (Gale-Shapley, 1962)

Input. Preference ranking by each of n man and n woman.

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

IF each woman receives exactly one proposal, THEN
  stop and **report** the resulting matching as *stable*.

ELSE
  every woman receiving more than one proposal rejects all of them except the one highest on her list.
  Every woman receiving at least one proposal says "maybe" to the most attractive proposal she received.

Iterate.

Theorem. The Proposal Algorithm produces a stable matching.