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 \( G \),
\[
c(u, v) = w(M).
\]
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 \( x_iy_j \) 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 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 \( v \):

\[
u_i := u_i - \epsilon \text{ if } x_i \in X \setminus R, \\
v_j := v_j + \epsilon \text{ if } y_j \in T.
\]

**Iterate**

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

---

**The Assignment Problem — An example**

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

**Excess Matrix**

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 \\
5 & 4 & 3 & 2 & 1 & 0 \\
8 & 2 & 1 & 0 & 1 & 6 \\
5 & 4 & 2 & 1 & 1 & 0 \\
8 & 5 & 2 & 6 & 0 & 1 \\
5 & 1 & 4 & 2 & 0 & 1
\end{array}
\]

**Equality Subgraph**

\[
\epsilon = 1
\]
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:

\[
\begin{pmatrix}
1 & 0 & 1 & 2 & 2 \\
3 & 1 & 2 & 3 & 4 \\
7 & 6 & 7 & 8 & 7 \\
3 & 1 & 3 & 4 & 5 \\
6 & 3 & 6 & 2 & 8 \\
3 & 4 & 1 & 3 & 5 \\
\end{pmatrix}
\]

\( w(M) = 5 + 7 + 4 + 8 + 4 = 28 = 1 + 0 + 1 + 2 + 2 + 3 + 7 + 3 + 6 + 3 = c(u, v) \)

Hungarian Algorithm — Proof of correctness

\textbf{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.

\textbf{Why is} \((u, v)\), \textbf{created by the iteration, a cover?}

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

\begin{align*}
&x_i \in R, \quad y_j \in Y \setminus T \quad \Rightarrow \quad u_i \text{ and } v_j \text{ do not change.} \\
&x_i \in R, \quad y_j \in T \quad \Rightarrow \quad u_i \text{ does not change} \\
&x_i \in X \setminus R, \quad y_j \in T \quad \Rightarrow \quad u_i \text{ decreases by } \epsilon, \\
&x_i \in X \setminus R, \quad y_j \in Y \setminus T \quad \Rightarrow \quad u_i + v_j \geq w_{i,j} \text{ by definition of } \epsilon.
\end{align*}

\textbf{Why does the algorithm terminate?}

\( M \) is a matching in the new \( G_{u,v} \) as well. So either

\begin{itemize}
\item[(i)] max matching gets larger or
\item[(ii)] \# of vertices reached from \( U \) by \( M \)-alternating paths grows. (\( U \) is the set of unsaturated vertices of \( M \) in \( X \).)
\end{itemize}

Matchings in general graphs

An odd component is a connected component with an odd number of vertices. Denote by \( o(G) \) the number of odd components of a graph \( G \).

\textbf{Theorem.} (Tutte, 1947) A graph \( G \) has a perfect matching \( \text{iff} \ o(G - S) \leq |S| \) for every subset \( S \subseteq V(G) \).

\textbf{Proof.}

\( \Rightarrow \) Easy.

\( \Leftarrow \) (Lovász, 1975) Consider a counterexample \( G \) with the maximum number of edges.

\textbf{Claim.} \( G + xy \) has a perfect matching for any \( xy \not\in E(G) \).
Proof of Tutte’s Theorem — Continued

Define $U := \{ v \in V(G) : d_G(v) = n(G) - 1 \}$

Case 1. $G - U$ consists of disjoint cliques.
Proof: Straightforward to construct a perfect matching of $G$.

Case 2. $G - U$ is not the disjoint union of cliques.
Proof: Derive the existence of the following subgraph.

Obtain contradiction by constructing a perfect matching $M$ of $G$ using perfect matchings $M_1$ and $M_2$ of $G + xz$ and $G + yw$, respectively.

Corollaries

Corollary. (Berge, 1958) For a subset $S \subseteq V(G)$ let $d(S) = o(G - S) - |S|$. Then

$$2\alpha'(G) = \min\{ n - d(S) : S \subseteq V(G) \}.$$  

Proof. $(\leq)$ Easy.
$(\geq)$ Apply Tutte’s Theorem to $G \cup K_d$.

Corollary. (Petersen, 1891) Every 3-regular graph with no cut-edge has a perfect matching.

Proof. Check Tutte’s condition. Let $S \subseteq V(G)$.
Double-count the number of edges between an $S$ and the odd components of $G - S$.
Observe that between any odd component and $S$ there are at least three edges.

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 $G$. 