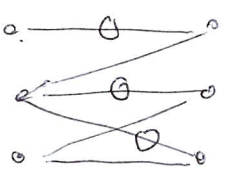
Optimal matchings

Last week: In bipartite graph $G = (S+T, E)$, matching is $M \subseteq E$ such that edges in M are not incident to each other.

$m(G) = \max_{\substack{M \subseteq E \\ \text{matching}}} |M|$ = "matching number of G"



M

Hall's theorem +: $m(G) = |S| - \max_{A \subseteq S} (|A| - |N(A)|)$.

Today: ① How to find a matching of size $m(G)$?

② How to find an "optimal" one given weight function on G?

Let M be a matching in G. Say

- $v \in S \cup T$ **M-saturated** if v is incident to some $k \in M$

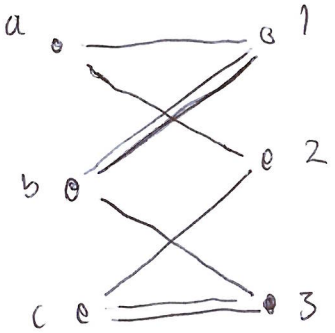
  M-unsaturated otherwise

- a path P in G is **M-alternating** if every other edge lies in M, & its start and end vertices are M-unsaturated

M = ═

a, 2 M-unsat.

a 1 b 3 c 2 M-alternating



Thm (89) A matching M is a maximum matching $\iff$ G has no M-alternating path.

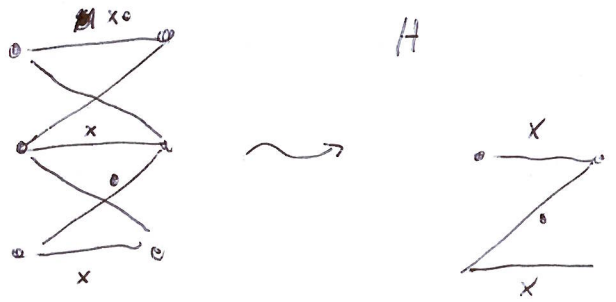
Pf: If G has M-alternating path, P, then

$\Rightarrow$ $M' = M \cup (E_P \setminus E_P \cap M)$

is a matching with $|M'| = |M| + 1$.

$\Leftarrow$ Assume M is not max. matching. Then there is a matching M' with $|M'| > |M|$. Let H be subgraph of G where edges are the ones contained in M or M' but not both.

$\circ = M$
$\times = M'$

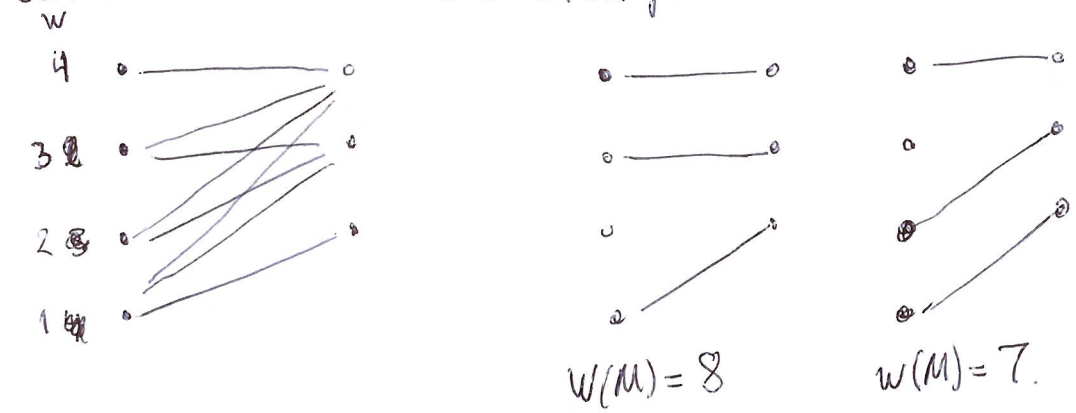


In H, every vertex has degree 1 or 2, so components are paths or circuits, with edges alternating between M & M'. $|M'| > |M|$ means some component must have more edges from M' than from M. This is M-alternating path.

□

Leads to algorithm for finding matching with $|M| = |S|$ if one exists (Algorithm 8.10)

Optimal matchings: Variant 1: Let $w: S \longrightarrow \mathbb{R}_{\geq 0}$ be given. We want a matching M such that $w(M) = \sum_{u \in S \cap V(M)} w(u)$ is as big as possible



Say $A \subseteq S$ is a transversal if there is a matching M with $V(M) \cap S = A$. Goal is to find transversal A with $w(A) = \sum_{u \in A} w(u)$ maximal.

Let $\mathcal{T} \subseteq 2^S$ be set of transversals.

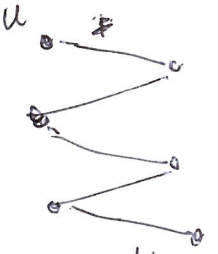
Thm $(S, \mathcal{T})$ is a matroid.

Pf ① $\emptyset = V(\emptyset) \cap S$, ✓

② If $A \subseteq B$ & $B = V(M) \cap S$, then removing edges gives M' s.t. $A = V(M') \cap S$

③ Let $A, B \in \mathcal{T}$ with $|A|+1 = |B|$, let M_A & M_B be the matchings. Pfe of Thm 8.9 gives an M_A-alternating path with edges in $M_A \cup M_B$.

One of end vertices of path lies in S. Increasing size of M_A gives matching M' with $V(M') \cap S = A \cup \{u\}$.

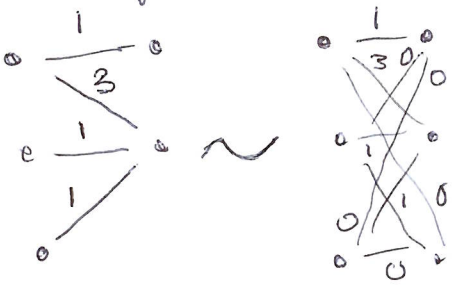


Cor Greedy algorithm solves this optimisation problem.

Optimisation, variant 2: Let $G = (S+T, E)$, $w: E \to \mathbb{R}_{\geq 0}$.

Want matching M so $w(M) = \sum_{k \in M} w(k)$ is as big as possible.

Simplifications: • Add "fake" vertices & edges, set $w = 0$ for these, get $K_{n,n}$. Optimal matchings stay the same.



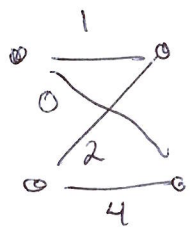
• Let $W = \max_{k \in E} w(k)$, $w'(k) = W - w(k)$.

$w': E \to \mathbb{R}_{\geq 0}$, maximising $w(M) \longleftrightarrow$ minimising $w'(M)$.

• Assume w takes integer values.

Let $S = \{u_1, \dots, u_n\}$, $T = \{v_1, \dots, v_n\}$, let

$\rightsquigarrow w_{ij} = w(u_i, v_j)$ weight matrix.



$$\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}$$

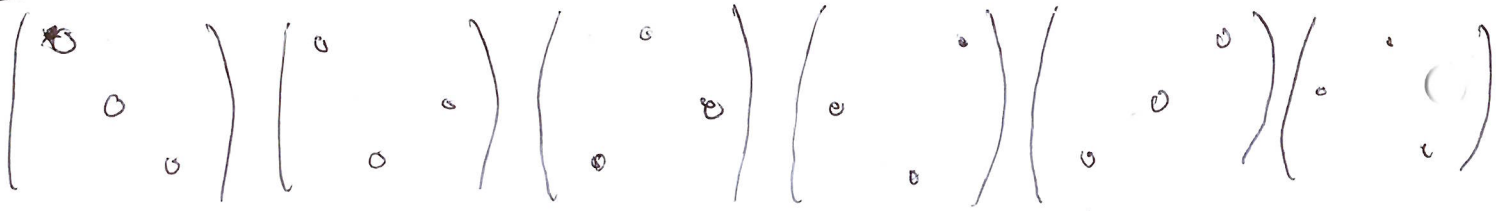
A ~~matching~~ maximum matching in $K_{n,n}$ is given by a choice of n edges $(u_1 v_{i_1}), (u_2 v_{i_2}), \ldots, u_n v_{i_n}$ so that i_j are distinct

$\longleftrightarrow$ a diagonal in the matrix. ($n!$ such)

$$\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix} \qquad 2+0 < 1+4$$

$\hookleftarrow$ minimal.

$n=3$



Want diagonal such that sum of its entries is minimal.

Hungarian algorithm (8.12) Idea: Given a matrix

$(w_{ij}) = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$ with optimal diagonal $D = (1,i_1)(2,i_2)\cdots(n,i_n)$.

Subtract/add some number a from every entry in a row or column.

Then same diagonal D is optimal

$-a$ $\begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$

-2

Ex $\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 \\ 0 & 4 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 & 0 \\ 0 & 3 \end{pmatrix}$

Algorithm: If some row/column has only >0 entries, subtract value of minimal entry. Repeat until every row/column has some 0 entry.

② If there is a diagonal with only 0's, stop – this is optimal.
Else, let D be diagonal with max. no. of zeros, say $d < n$.
Then (Sec. 8.1) can find e rows & f columns, $e+f=d$ covering every 0 in matrix.

Say (i,j) <u>covered</u> if (i,j) lies in one of these rows/columns

Let $\bar{w} = \min_{(i,j) \text{ not covered}} w_{ij}$, & let

$$w'_{ij} = \begin{cases} w_{ij} - \bar{w} & \text{if } (i,j) \text{ uncovered} \\ w_{ij} & \text{if } (i,j) \text{ covered once} \\ w_{ij} + \bar{w} & \text{if } (i,j) \text{ covered twice} \end{cases}$$

<u>Then</u>: • $w'_{ij} \geq 0 \quad \forall i,j$

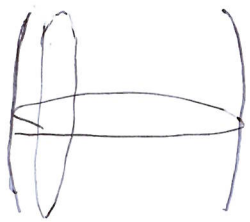
• $\sum w'_{ij} = \sum w_{ij} - \bar{w} \cdot (\#\text{uncovered } (i,j)) + \bar{w}(\#\text{doubly covered } (i,j))$

$$= \sum w_{ij} - \bar{w}(n-e)(n-f) + \bar{w}ef$$

$$= \sum w_{ij} + \bar{w} \cdot n(n-e-f) < \sum w_{ij}.$$

Repeat step 2 until diagonal with only 0's found.

<u>Example</u>: $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 1 & 4 \\ 2 & 5 & 9 \end{pmatrix}$ ① $\rightarrow \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 4 & 5 \end{pmatrix}$ ① $\rightarrow \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 3 & 4 \end{pmatrix}$

② $\rightarrow$ D, $d=2$ $\begin{pmatrix} (0) & 2 & 1 \\ 1 & (0) & 0 \\ 0 & 3 & (4) \end{pmatrix}$ covering $\rightarrow$  Find $\bar{w}$ $\rightarrow$ $\begin{pmatrix} 2 & (1) \\ 3 & 4 \end{pmatrix}$ $\bar{w}=1$

$w'_{ij} \rightarrow \begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 0 & 2 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} & 0 & \\ & & 0 \\ 0 & & \end{pmatrix}$ optimal diagonal found.

$2+1+5$ is weight of minimal matching.