Lecture 7: Matchings Part 1: Combinatorial Algorithms.

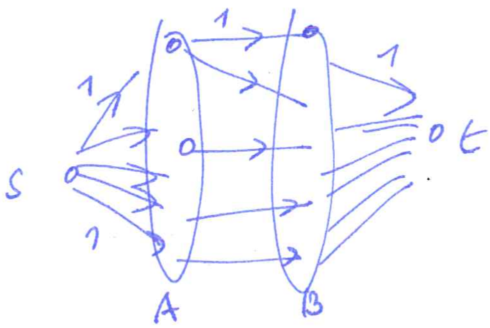
- G: ~~undi~~ undirected graph (V,E).
- Matching: set of edges $M \subseteq E$ s.t. Every ~~ed~~ node has degree ≤ 1.
- Max cardinality matching in G ← or max weight matching.
  "Perfect" matching ≡ every node has degree = 1.

Some examples  etc.

- jobs & machines
- boys & girls.

Bipartite matching is a simpler case. Easier proofs, "faster" algorithms, etc.

(1) via a reduction to maximum flows.



max flow = maxcard matching:

- ~~Since~~ FF = augmenting paths runs in time $O(mf^*) = O(mn)$
- Even & Tarjan / Dinic showed that for unit capacity graphs, the runtime is $O(m \min\{m^{1/2}, n^{2/3}\})$.

(2) Via direct algorithms (also based on "augmenting paths").

Given a matching $M \subseteq G$ (Technically means $M \subseteq E(G)$).

Edges of M denoted by ～～. Non-M edges o—o

- an alternating (simple) paths alternates between M & non-M edges

o～～o——o～～o——o～～o——o etc

- an open vertex is not matched in M.
- an augmenting path is an alternating path P between 2 open vertices. (must be odd length)

⇒ $M \Delta P$ is also an matching, of greater cardinality.
↑ symmetric difference

Fact [Berge] M is a maximum cardinality matching if and only if no augmenting path with respect to M.

Proof: if $\exists$ any path $\Rightarrow$ clearly M not maximum.

Sps M ~~maxim~~ not max. let M^* be maximum.

$M \Delta M^*$ = alternating paths and cycles.

Since $|M^*| > |M|$ must have at least one odd length alternating path wrt M.

with both ends open in M $\Rightarrow$ M-augmenting path. ☒

$\Rightarrow$ just find M-augmenting paths until reach max matching.

How to do this?

Bipartite Case is easier. $G = (L, R, E)$

N.b. showing that M is a max_matching is tricky using this characterization. Have to show a "co-NP type" certificate. Here's another way to show that M is max-matching.

· König's theorem: G, bipartite, let $MM(G)$ = max matching VC = min vertex cover ← set of vertices that touch every vertex.

then $MM(G) = VC(G)$.

[it is easy to show that $\forall VC, \forall$ Matchings, $M \leq VC$]

· So we could also show a VC of cardinality $|M|$ and hence prove optimality.

Very good. But how to find large matching?

Idea basically looks like FF on the reduction graph.

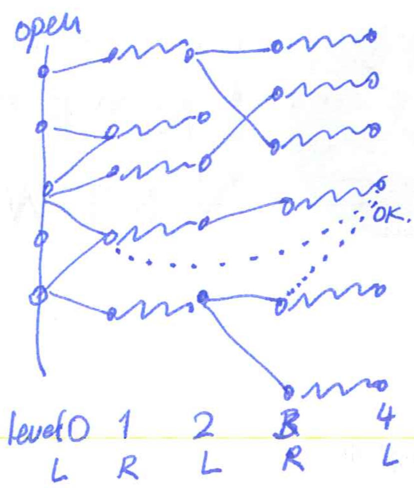
Take all open vertices in L level 0. (marked).

When at level i, (even), $\in L$

must be $\subseteq R$ { level $i+1$ = all unmarked vertices connected to level i vertices by non-M edges.

~~level i+2 =~~ if we see an open vertex at level $i+1$, then found an odd length alt path b/w 2 opens $\Rightarrow$ aug.

(where can edges from level i (even) go?
not to L, because bipartite so to R.)
(to previous odd levels, or to unmarked new vertices)



---

level $i+2$: take matched edge out of the level $(i+1)$ vertices. Since we did not find any open $(i+1)$-level nodes, must do this.

Do until all vertices marked or unreachable.

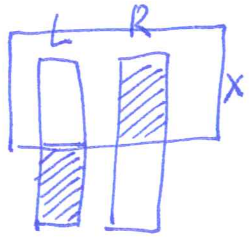
Note: if $\exists$ an M-augmenting path, we will find it this way. in $O(m)$ time. So if we find

$\Rightarrow$ total time $O(mn)$.

[Really just doing Ford Fulkerson].

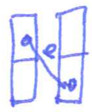
Can we use König's theorem to prove optimality?

sure. let X = marked vertices. then



Claim: $C^* := (L - X) \cup (R \cap X)$ is a vertex cover of size M.

Pf: if e not covered $= (\ell, r)$ $\Rightarrow$ ℓ is ~~labeled~~ marked but r is not ~~labeled~~ marked.



$\Rightarrow$ e cannot be in matching. else when ℓ is marked, ~~next r would be marked.~~ r would just have been marked.

if e in M. ℓ can and will be marked only from r. $\Rightarrow (r \notin X \Rightarrow \ell \notin X)$

If $e \notin M$: $\ell \in X \Rightarrow r \in X$.

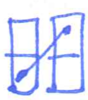
rephrase

$\Rightarrow$ when ℓ is marked, r would be marked next.

$\Rightarrow$ no such e ~~exists~~ exists.

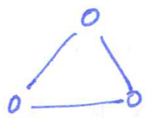
Next: $|C^*| \leq |M|$

- Every vertex in $R \cap X$ has a matching edge incident to it.
- Every vertex in $L - X$ has - - - - - - (else would be picked in level 0).
- there are no edges between $L - X$ and $R \cap X$, so distinct edges.



$\Rightarrow |C^*| \leq M$.

Not true in non-bipartite graphs



$VC^* = 2$
$MM^* = 1.$

~~For general graphs the theory is richer/deeper.~~

Tutte Berge: Suppose pick a set $U \subseteq V$, and delete it from G.

to get components $K_1, K_2 \dots K_k$

How big a matching can graph have?

$$|U| + \sum_{i=1}^{k} \left\lfloor \frac{K_i}{2} \right\rfloor = |U| + \left(\frac{|V|-|U|}{2}\right) - \left(\frac{\#\text{odd comps.}}{2}\right)$$

one edge per vertex in U

if $\mathrm{odd}(G \setminus U) = \#$ of odd components in $G \setminus U$,

$$|M| \le \min_{U \subseteq V} \frac{|U| + |V| - \mathrm{odd}(G \setminus U)}{2}$$

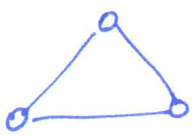
Theorem: [Tutte/Berge] for a graph G,

$$|MM(G)| = \min_{U \subseteq V} \frac{|U| + |V| - \mathrm{odd}(G \setminus U)}{2}.$$

Observation: ~~if G is bipartite then~~ let U = vertex cover of G. $(G \setminus U)$ has only isolated vertices (no edges), so $\mathrm{odd}(G \setminus U) = |V| - |U|$.

$$\Rightarrow RHS \le |U| = |VC^*|.$$

This is clearly a bound stronger than $MM^* \le VC^*$.



take $U = \emptyset \Rightarrow RHS = \frac{0+3-1}{2} = 1 = MM^*$.

etc.

How to prove this? Via an algorithm [Edmonds Blossom Algo].

<u>Again</u>: want to find M-augmenting path (if one exists).

<u>Ideal</u>: find M-aug path P, $M \leftarrow M \triangle P$, repeat.

<u>Unfortunately</u>: can only show.

<u>Thm 1</u>: Suppose ~~G~~ contains an M-augmenting path P. then

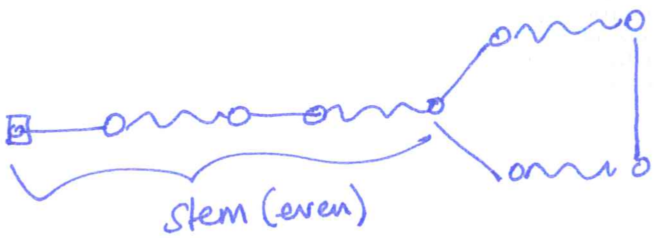
∃ an algorithm ~~that~~ finds either

(a) an M-augmenting path or

(b) a blossom.

} and then we show how to make progress in either case.

<u>What's a blossom?</u>



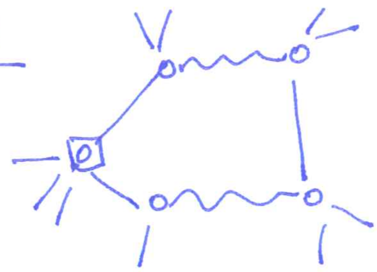
<u>A stem</u>

(~~an~~ alternating path

starting at open vertex)

and a <u>blossom</u> (an almost alternating cycle of odd length)

<u>If we find a flower (with stem S and blossom B)</u>



(i) toggle edges on stem (i.e. $M \leftarrow M \triangle S$)

to get flower with empty stem

open!!

(ii) shrink B into new vertex v_B. Call graph G/B.

(iii) Find a (M/B)-augmenting path P' in G/B (recursively)

↑ just drop the edges of M on B. Keep all others.

(iv) Extend P' to an M-augmenting path P. in G.

<u>Hence</u>: in either case find M-augmenting path. ⇒ make progress.

Q1 How does this algo work?

Q2 Why is this OK? For this we need Thm 2.

Theorem 2: $\exists$ a M-aug path in G $\Leftrightarrow$ $\exists$ a $M/_B$ aug path in $G/_B$.

Pf: if M-aug path P does not hit B $\Rightarrow$ still in G/B.

($\Rightarrow$) Else: at least one end of P not on B.

(P has 2 ends open, B has one open node).

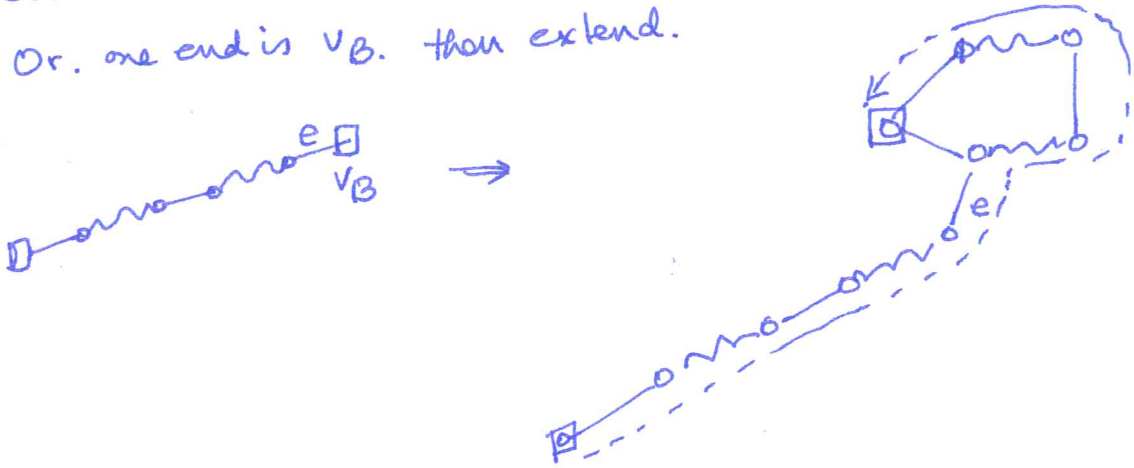
say u. let v be first node on $P \cap B$. then

~~$P \cap B$ is~~ $P[u \to v]$ is M/B aug path in G/B.

[Recall: v_B is open in G/B].

($\Leftarrow$). Consider P' in G/B. Either misses v_B then in G also

Or, one end is v_B. then extend.



Now to theorem 1.

Algo: Start with all open guys on level 0. (Mark them).

Do a simultaneous "BFS" from L0 as follows.

Given L(2i), do the following:-

look for all nonmatching edges from L_{2i}. Say $u \in L_{2i}$, v other end.

(i) if v unmarked, $v \in L_{2i+1}$, mark. (even odd)

(ii) if v at same level: aug or blossom!

Why? look at paths open $\rightsquigarrow$ v, open $\rightsquigarrow$ u and $(uv) \notin M$.

(iii) if v at previous odd level OK. (e-o).

(iv) if v at previous even level L_{2j} ($j<i$).

Not possible! else u would be at level $2j+1 < 2i$.

$\Rightarrow$ either success or all edges (even-odd)

Given L_{2i+1}, ~~get L_{2i+2} as~~ do as follows: $u \in L_{2i+1}$ $(u,v) \in E \cap M$. matching edges only!

(i) if v ~~$\in L$~~ unmarked, $v \in L_{2i+2}$, mark. [o-e]

(ii) if $v \in L_{2i+1}$, ~~path or~~ aug or blossom !!

(iii) v cannot be at previous levels.

$\Rightarrow$ ~~if no success; all edges~~ continue until Level = empty.

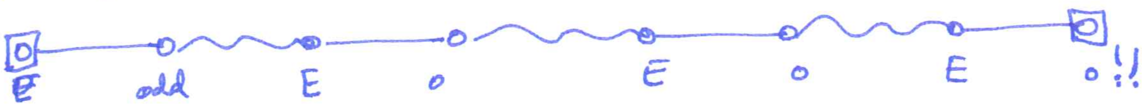
N.b. if ~~all~~ no success $\Rightarrow$ all edges even-odd.

Now AFSOC: $\exists^{M}$ augmenting path P in G, and we don't find aug or blossom

~~So~~ Want a contradiction:—

Label each vertex with parity (even/odd).

- Ends have parity E (open $\Rightarrow L_0 \Rightarrow$ even).
- Endpoints of edges have opposite parity
- But path of odd length $\Rightarrow$ even # of vertices.



Finaly: [Tutte Berge.] Sps $G \overset{*}{\rightsquigarrow} G'$ and here Edmonds did not find any cross edges ("succes")

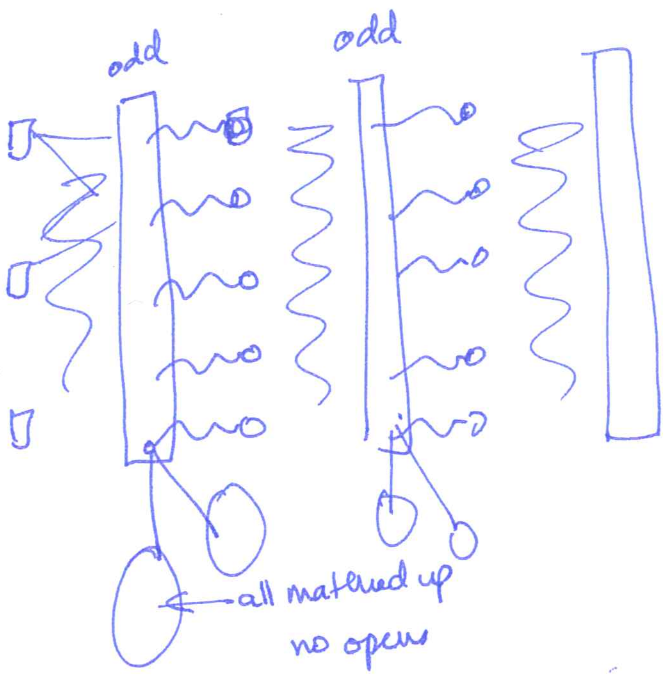
~~take all odd level vertices in G':~~

~~Claim 1: Even~~

marked vertices in G' are bipartite graph

$\Rightarrow$ take all odd vertices. Each one has a successor matched edge $\Rightarrow$ Set $U = \text{odds}$

And all unmarked nodes are matched (not open)



So $(G' \setminus U)$ has $\begin{cases} |\text{Even}| \text{ components all singletons} \Rightarrow \text{odd} \\ \text{all other components matched up} \Rightarrow \text{even.} \end{cases}$

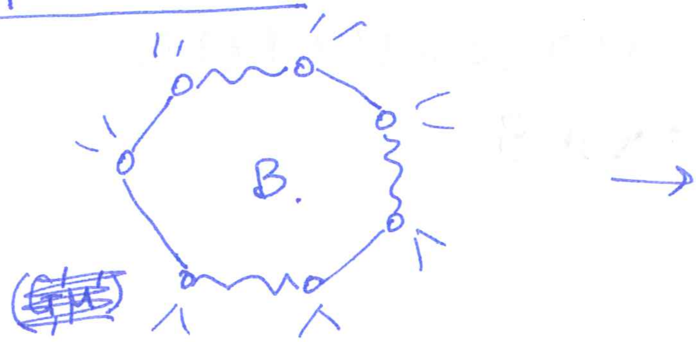
$$\Rightarrow \frac{|n'| + |U| - odd(G' \setminus U)}{2} = \frac{n' + |odd| - |Even|}{2}$$

$$= \frac{n' + |odd| - (n' - |odd| - rest)}{2}$$

$$= \frac{2\,odd + rest}{2} = odd + \frac{rest}{2}$$

$$= M'.$$

Now pull back to G:



V_B

$$\begin{array}{cc} (G', & M') \\ \| & \| \\ G/B & M/B \end{array}$$

- find U' st
$$|M'| = \frac{n' + |U'| - odd(U')}{2}$$

- Want to extend back to U.

$$M = M' + \frac{B-1}{2}.$$

note: V_B was even and open $\Rightarrow$ not in U'. (and in fact in odd component).

$\Rightarrow$ replace V_B by B still means in odd component. (# vertices in that component increases by $B-1$ = even).

$$\Rightarrow |M| = M' + \frac{B-1}{2} = \frac{n' + |U| - odd(U)}{2} + \frac{B-1}{2}$$

$$= \frac{n + |U| - odd(U)}{2}.$$

$\blacksquare$

Tutte's Perfect Matching Thm: G has a PM $\Leftrightarrow$ $o(G \setminus U) \leq |U|$ $\forall U \subseteq V$.

Runtime: find alternating path in time $O(mn)$ $\leftarrow$ $O(n)$ recursions, $O(m)$ time per.

$\Rightarrow$ total: $O(mn^2)$.

Can do better: [Micali Vazirani] $O(m\sqrt{n})$.

[Mucha Sankowski] $O(n^{\omega})$.