CS6381 : Lecture #14; 9. 25. 2020

Matching. (See also printed notes in homepage)

Let $G = [V; E]$ be an undirected graph.

$M \subseteq E$ is called a matching if no two edges in M are incident at a vertex of G.

i.e each node v in V has atmost one edge incident at v.

Three problems in Matching

P1: given G, find a matching M whose size is maximum.

Called Cardinality Matching Problem

P2: Given G with edge weights $w(e); e \in E$, Find M with maximum total weight

i.e. $\max \sum_{e \in M} w(e)$

Called Weighted Matching Problem

P3: A matching M is called a perfect matching if each node v in G has exactly one edge incident at v.

Find in (G, w), a perfect matching M with min(max) total weight : $\min(\max) \sum_{e \in M} w(e)$.

P1 is included in P2 : $w(e) = 1 \ \forall e \in E$

Now we show how to convert $\underline{\underline{P2 \Leftrightarrow P3}}$

$\Rightarrow$: Given P2, edges with $w(e) \leq 0$ can be dropped without loss. So we assume $w(e) > 0$ $e \in E$.

If $|V|$ is odd $\rightarrow$ add a vertex ~~with edges to all other~~

Complete the graph so that non-existent edges are now added with $w(e) = 0$.

Now solve perfect matching problem.

(removing 0 weight edges solves P2)

Thus P2 can be converted to P3.

$\Leftarrow$ In any perfect matching M, $|M| = \frac{|V|}{2}$

Since $|M|$ is constant in all perfect matchings, if we change weights by adding same constant to each edge, solution does not change.

i.e. $w'(e) = w(e) + k \quad \forall e \in E$

$$\sum_{e \in M} w'(e) = \sum_{e \in M} w(e) + k|M|.$$

P1 is included in P2 : $w(e) = 1 \ \forall e \in E$

Now we show how to convert $P_2 \Leftrightarrow P_3$

$\Rightarrow$: Given P2, edges with $w(e) \leq 0$ can be dropped without loss. So we assume $w(e) > 0$ $e \in E$.

If $|V|$ is odd $\rightarrow$ add a vertex ~~with edges~~

~~to all other~~

Complete the graph so that non-existent edges are now added with $w(e) = 0$.

Now solve perfect matching problem.

(removing 0 weight edges solves P2)

Thus P2 can be converted to P3.

$\Leftarrow$ In any perfect matching M, $|M| = \frac{|V|}{2}$

Since $|M|$ is constant in all perfect matchings, if we change weights by adding same constant to each edge, solution does not change.

i.e $w'(e) = w(e) + k \quad \forall e \in E$

$$\sum_{e \in M} w'(e) = \sum_{e \in M} w(e) + k|M|.$$

Select a "large" positive value for K.

Now consider P2 with edge weights = $w'(e)$.

Weight of <u>any</u> perfect matching exceeds the weight of <u>any</u> non-perfect matching. i.e

$$\sum_{e \in PM} w'(e) = \sum_{e \in PM} w(e) + |M| \cdot K$$

$$> \sum_{e \in (M' - PM)} w(e) + |M'| K$$

Since $|M| > |M'|$, and K is large.

Hence, any optimal solution to P2 with edge weights $w'(e)$ will have to ~~a~~ be a PM.

Hence solution to P2 is also a solution to P3

Given $P2 \supseteq P1$, and $P2 \Leftrightarrow P_3$, we choose to solve <u>only</u> P3. But we still need solution to P1 as this is used as a subroutine in the algorithm to solve P3. So we consider some results for P1 in a Primal-Dual algorithm.

Definitions:

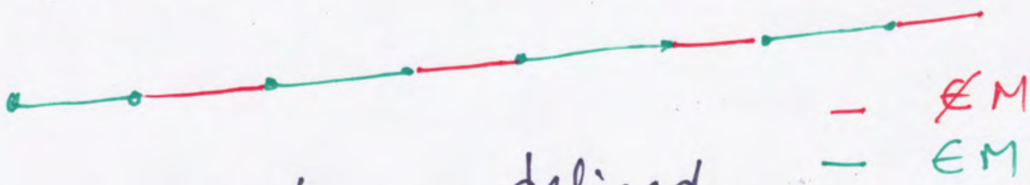
A node v is free (exposed) relative to a matching M if no edge in M is incident at v. ~~in~~

A perfect matching leaves no exposed node.

In order for G to have PM, a necessary (but not sufficient) condition is that

$|V| =$ even.

A path is said to be alternating with respect to M if edges in the path are alternatingly in M and not in M. See picture below



Alternating cycles are defined in a similar manner.

An alternating path whose terminal nodes are both exposed (free) is called an augmenting path. Since $M' = M -$ {Green edges} + {red edges} is also a matching and $|M'| > |M|$.

[Notice Similarity to augmenting paths in flows]

(C. Berge, N. Rabin)

<u>Theorem 1</u>: A matching is of maximum Cardinality if and only if $\nexists$ no augmenting path relative to it.

<u>Pf</u>: "Only if" is clear from the definition of augmenting path.

<u>To prove the if part</u>:

Let ~~M~~ M and P be two matchings in G.

Consider the graph $G^{\Delta} = [V; P\Delta M]$

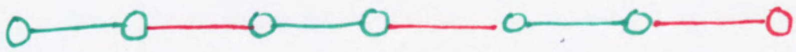
where $P \Delta M = \left\{ e \in E : \begin{array}{l} e \in P - M \text{ or} \\ e \in M - P \end{array} \right\}$

Since each of P and M have at most one edge at each vertex of G, G^{Δ} has at most two ~~vertices~~ edges at each vertex. Hence, each ~~conn~~ Connected component of G^{Δ} is a path, a cycle or an isolated node.

Moreover, paths and cycles of G^{Δ} are alternating with respect to M and P.

In the next page we show <u>all</u> possibilities:

— ∈ M
— ∈ P

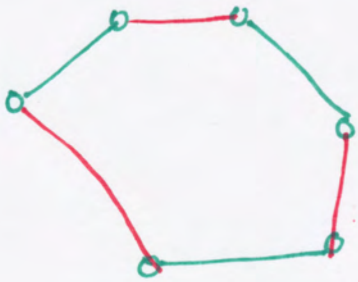
(i) 

(ii) 

(iii) 

(iv) 

(v)



In cases (i), (iv) and (v) : $|P-M| = |M-P|$

In case (ii) $|M-P| = |P-M| + 1$

In case (iii) $|P-M| = |M-P| + 1$

Hence if $|P| > |M|$, we must have a component of type (iii) which is an <u>augmenting path</u> relative to M.

Although this is a characterization of max. card. matching, it is <u>far from being useful</u> in a constructive algorithm (which is polynomial)

J. Edmonds showed how to develop an algorithm

Integer Programming Formulation of PMP

$$x_e = \begin{cases} 1 & \text{if } e \in M \\ 0 & \text{else} \end{cases}$$

P

$$y_i : \sum_{e:\text{incident at } i} X_e = 1 \quad \forall i \in V \leftarrow \text{PM}$$

Unrestd

$$\text{Max} \sum_{e \in E} w_e x_e$$

Dual of P (treating it as if it is a linear program)

(Integer Programs do not have a dual in the same way as linear programs)

$$x_e \geq 0 : \quad y_i + y_j \geq w_e \quad ; \; e = (i,j) \in E$$

$$\text{M}\cancel{\text{ax}}\text{in} \sum_{i \in V} y_i$$

Lemma (Weak duality): For any y feasible to D and any PM, x for P

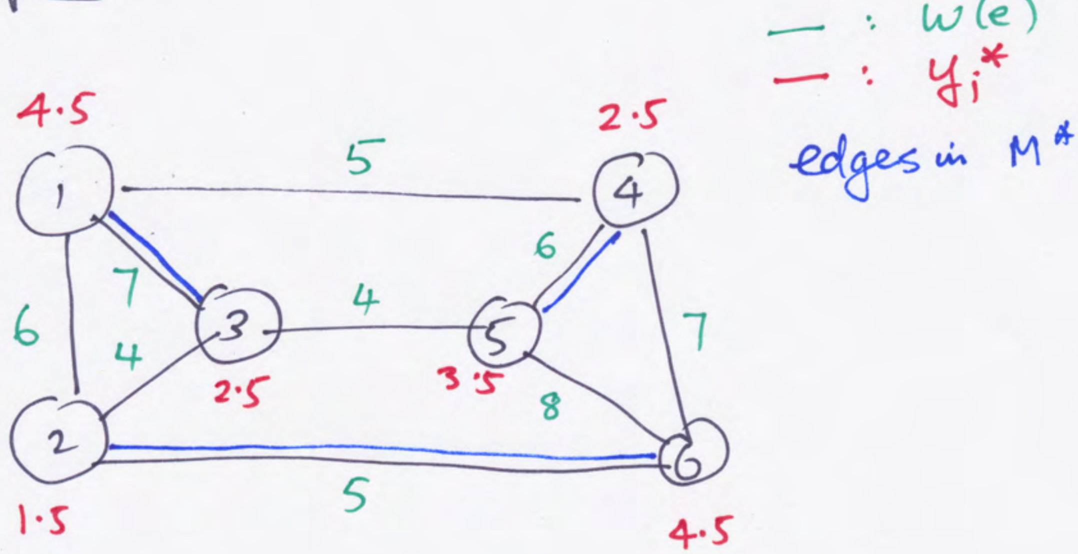
integer feas. to P as LP

$$\sum_{e \in PM} w_e \leq \sum_{i \in V} y_i \quad . \quad (*)$$

If equality holds in $(*)$, these are optimal.

Unfortunately, we may not have equality in some instances.

Example



$\sum_{e \in M^*} w_e = 18$; $\sum y_i^* = 19$.

The difference is known as Integrality gap

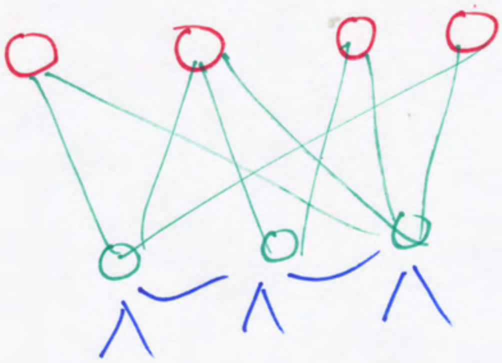
For bipartite graphs, integrality gap = 0 and equality holds in (*)

To see which graphs may not have a PM we develop the concept of a Hungarian set
[The first person to consider matching problem was Egervary who is a Hungarian ; hence the name].

A subset H of nodes of G is called a Hungarian Set of G if

(i) No two nodes in H are connected by an edge of G.

(ii) the set $N_G(H)$, of neighbors of H in G (i.e nodes connected to some node in H by an edge in G) satisfies the relation $|N_G(H)| < |H|$



— nodes in H

— nodes in $N_G(H)$

Clearly, if G contains a Hungarian Set H, then G can not have a PM (regardless of the type of graph G)

An algorithm shows that the converse is true if G is bipartite because it contains no odd cycles.