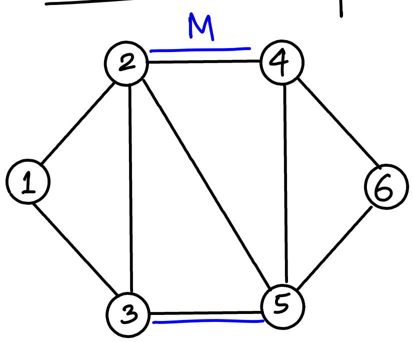
MATH 566: Lecture 28 (11/21/2024)

Today: Nonbipartite cardinality matching

Recall: non-bipartite matching

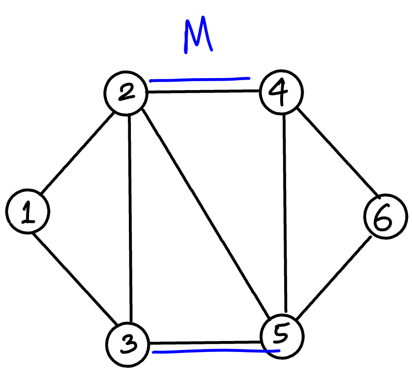


Matching M: subset of arcs of G s.t. any node is incident to at most one arc in M.

Size of matching $= \#$ arcs in M (M)
$\leq \lfloor \frac{n}{2} \rfloor$ when $|N| = n$.

Def A path $P = i_1 - \cdots - i_k$ is an **alternating path** w.r.t. a matching M if every pair of consecutive arcs in P contains one arc in M and one not in M.

Def An alternating path is an odd (even) alternating path if it has an odd (even) # arcs.



e.g.,

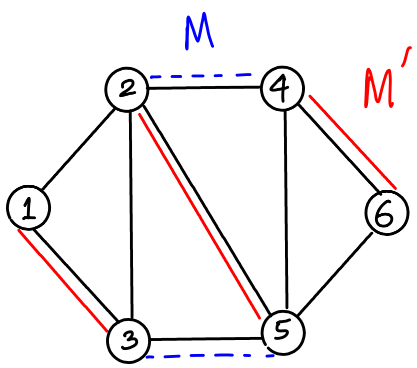
$1 - 2 = 4 - 6$
$1 - 3 = 5 - 2 = 4 - 6$ } odd alternating paths

$3 = 5 - 2 = 4 - 6$ } even alternating path

An alternating cycle is an alternating path that starts and ends at the same node.

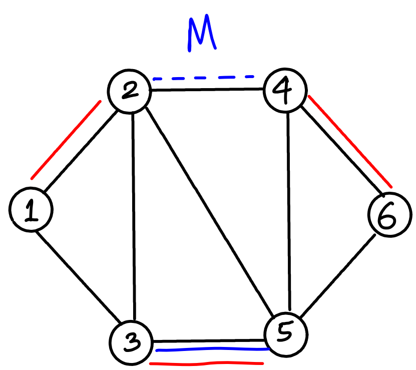
Def An odd alternating path w.r.t. matching M is an **augmenting path** if both its first and last nodes are unmatched in M.

We can swap the matched and unmatched arcs in an augmenting path to get a matching M′ with $|M'| = |M| + 1$.



$P = 1 - 3 - 5 - 2 - 4 - 6$ is an augmenting path

$|M'| = 3 = |M| + 1$.



$P' = 1 - 2 - 4 - 6$

$M' = \{(1,2), (4,6), (3,5)\}$

$|M'| = 3 = |M| + 1$.

We use set theoretic notation to describe these operations.

The **symmetric difference** of two sets S_1 and S_2 is denoted

$$S_1 \oplus S_2 = (S_1 \cup S_2) - (S_1 \cap S_2).$$

"oplus"

In words, this is the collection of elements in either set but not in both.

e.g. $\{2,3,5,6\} \oplus \{3,6,8,4\} = \{2,5,8,4\}$.

In the examples, we found $M \oplus P$ (or $M \oplus P'$), where M is a given matching and P is an augmenting path.

AMO Property 12.6 Let P be an augmenting path w.r.t. matching M. Then $M \oplus P$ is a matching with size $|M|+1$.

All nodes matched in M remain matched in $M \oplus P$, and two additional nodes are matched.

Def An **augmentation** of a matching M is to find an augmenting path P, and find $M \oplus P$.

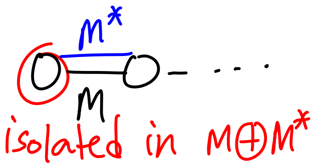
Here is an idea for an augmenting path algorithm for matching:

* Start with a matching M.
* pick a node i unmatched in M,
  - find augmenting path P starting at i,
  - find $M \oplus P$.
* Repeat.

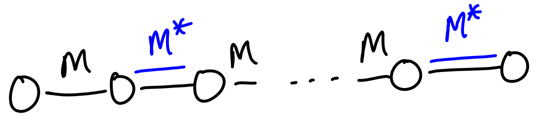
Q. What if we cannot find augmenting path starting at unmatched node i?

Way Out We can show that there exists a maximal matching M' in which node i is unmatched. So we try to find M', rather than continuing to grow current matching M.

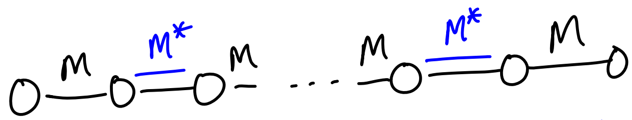
AMO Property 12.7 Let M and M^* be two matchings. Then $M \oplus M^*$ defines $G' = (N, M \oplus M^*)$, a subgraph of G, with the property that every component of G' is one of the following six types.



(a) O (degree 0; isolated node)

(b) 

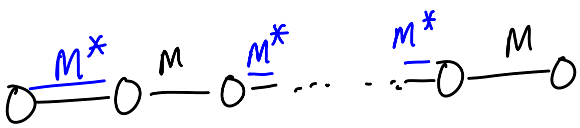
even

(c) 

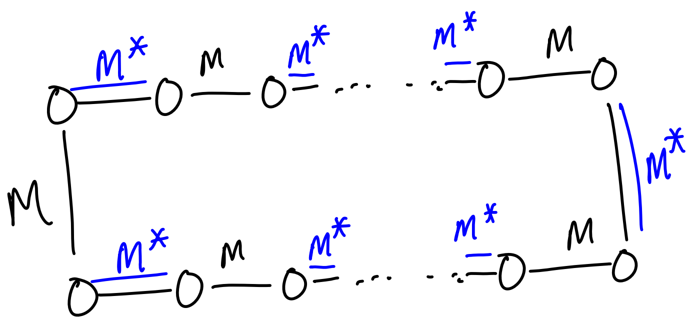
odd (one arc in M is unpaired with M^*)

(d) 

odd (one arc in M^* unpaired)

(e) 

even

(f) 

even cycle

Note that the starting node (left-most) is matched to M in (b), and in M^* in (e), both even components.

Result follows from the fact that every node in G' has degree 0, 1, or 2, and the above six types are the only possibilities.

The augmenting path algorithm depends on the following theorem.

AMO Theorem 12.8 If a node p is unmatched in matching M, and there is no augmenting path starting at node p, then node p is unmatched in some maximal matching.

Proof Let M^* be a maximum matching. If p is unmatched in M^*, we are done. So, assume p is matched in M^*.

Consider $M \oplus M^*$. This subgraph has components as specified in Property 12.7.

Since p is unmatched in M, p has to be the starting node in a component of type (d) or (e). As there is no augmenting path starting at node p in M, the only option is (e), which is even.

Then $M' = M^* \oplus P$ is also a maximum matching, as P is an even path and M^* is maximal. And p is unmatched in M'. $\square$

Note that this is an existence result ☹!

This result does not tell how to find M^* or M'...

Here is an algorithm we could consider:

* start with any matching M (e.g., zero matching)
* pick a node p that is unmatched in M
* search for augmenting path starting at node p.
  - if path found, augment
  - else delete p and all arcs incident to p.

Unfortunately, this algorithm is guaranteed to work only for bipartite networks!

Come to my Math 567 class in Spring (Integer Optimization) to learn how to solve this problem!