MATH 340 - Discrete Structures 2

McGill University - Winter 2013

Last Updated: February 24, 2013

Contents

Information

- Instructor: Bruce Shepherd
- LaTeX: Ehsan Kia
- Notes: Catherine Hilgers

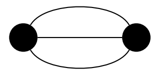
1 Summary of Graph Theory Terms

A (simple) graph G is an ordered pair $(V(G), E(G))$, sometimes written (V, E), where $V(G)$ is a finite set of *vertices* (aka nodes), and $E(G)$ is a finite set of edges.

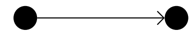
Each edge is of the form $\{u, v\}$ sometimes written uv, where $u \neq v$ are two vertices that are the end points of the edge.

NOTE: simple, undirected graph mean that we have no:

(a) Loops

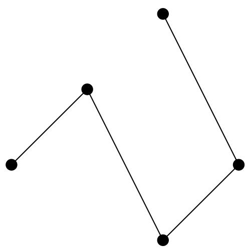


(b) Multiple edges

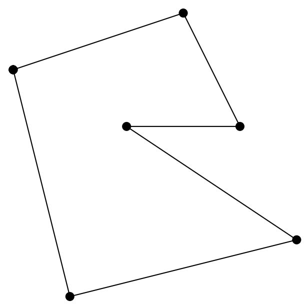


(c) Direction

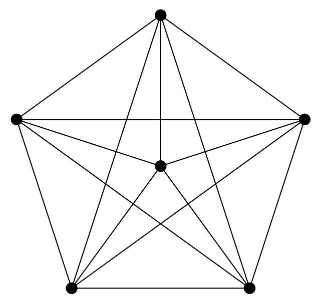
1.1 Special Graphs



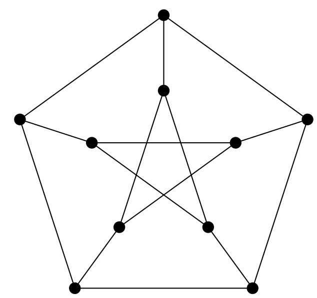
(a) Path



(b) Cycle



(c) Complete graph



(d) Petersen graph

Figure 1: Examples of simple graphs

A graph is *connected* if $\forall\, i, j \in V, \exists$ a path between them.

A *tree* is a connected graph with no cycles (Figure 2).

A *component* of G is a maximal connected subgraph.

The *degree* of a vertex v is the number of edges of which it is an endpoint, denoted by $deg_G(v)$ or $d_G(v)$.

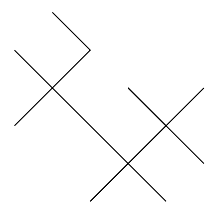


Figure 2: A tree

2 Stable Marriages

We have n boys and n girls. Each boy has an ordered list of girls and vice versa.

A set M of marriages is *stable* if there is no boy-girl pair who prefer each other to their current pairings in M. We call this situation an unstable (unblocking) pair [Figure 3].

2.1 Example

In the following example [Figure 4], we have 3 boys and 3 girls, each with their own preference list, but the given matching isn't a stable marriage.

But when trying again, we can easily find two stable configuations [Figure 5]

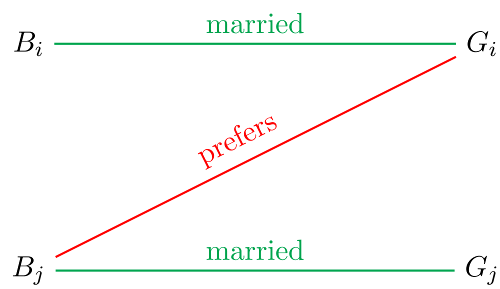


Figure 3: **Unstable pair** B_j prefers G_i to G_j and G_i prefers B_j to B_i

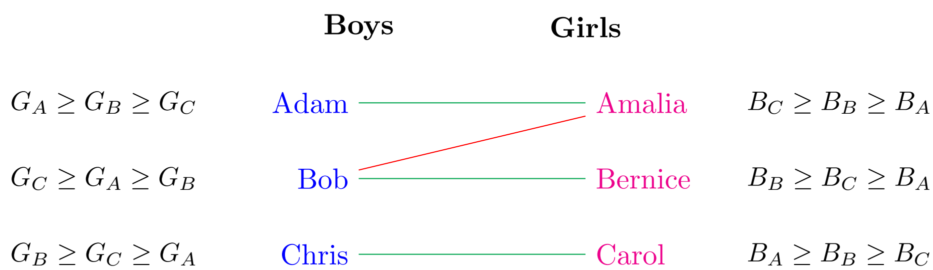


Figure 4: Unstable because Amalia and Bob prefer each other over their current partner

2.2 Gale-Shapley

Do stable matchings exist in general?

Theorem (Gale & Shapley): A stable matching always exists

Proof (by algorithm): While there is some "single" boy B, B proposes to the next girl on his list, call her G. Girl G accepts if she is single or prefers B to her current fiancé. Claim is that the algorithm terminates for any set of lists with a stable matching.

NOTE: as the algorithm proceeds, girls' choices only get better and mens' only get worse. Each time a girl changes fiancè, she trades up. A boy only changes if he gets dumped by G and he then proposes to the next girl on his list.

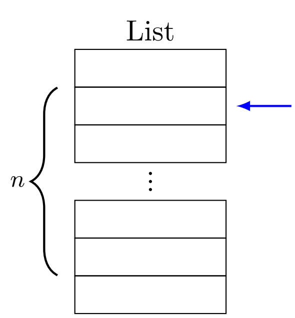


Figure 6: Preference list

Corollary: The algorithm terminates. Say boy B has his list. The pointer aims at his current match. There are n boys and n possible pointers into their lists [Figure 6]. each dumping moves the pointer down the list by one. We have $\leq n^2$ total dumpings. The algorithm terminates after $\mathcal{O}(n^2)$.

The matching returned by the algorithm is stable. Suppose M is the output matching, and has unstable pair (B_i, G_j), for a contradiction:

- B_i prefers G_j to current match G_i
- G_j prefers B_i to current match B_j

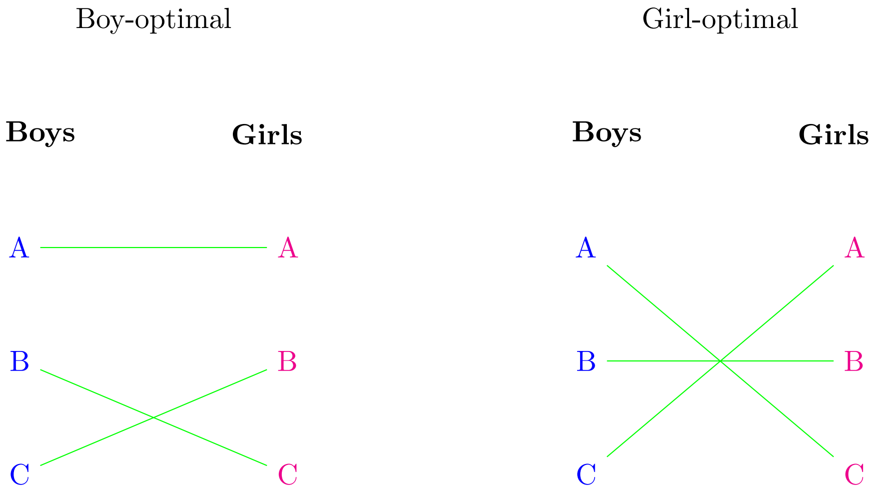


Figure 5: These work because each boy prefers a different girl, and each girl prefers a different boy.

Since B_i prefers G_j to G_i, he proposed to her earlier and she either rejected him, or accepted and dumped him later. In either case, she was at some point matched to some B_k she preferred to B_i. By observation, her partners only improved from that point on. Thus, she prefers B_j to B_k and B_k to B_i $\Rightarrow$ prefers B_j to B_i and (B_i, G_j) is not unstable. $\Rightarrow\Leftarrow$ (contradiction)

There can be many stable matchings. Let:

$$\mathcal{S} = \{M_1, M_2, ..., M_k\}$$

be the set of all stable matchings. Call G_j a *valid partner* for B_i if (B_i, G_j) are matched in some $M_i \in \mathcal{S}$. For each B, let $G^+(B)$ be his most preferred valid partner.

Remarkably, the boy-proposal algorithm matches each boy B to $G^+(B)$. To show this, we require a lemma:

Lemma: a girl never rejects a valid partner

Proof (by contradiction): Suppose not. Consider the first time G_j rejects a valid partner B_i. Say (B_i, G_j) were matched in $M_t \in \mathcal{S}$. Say G_j dumps B_i for B_j at that time. Say (B_j, G_k) is a match in M_t [Figure 7].

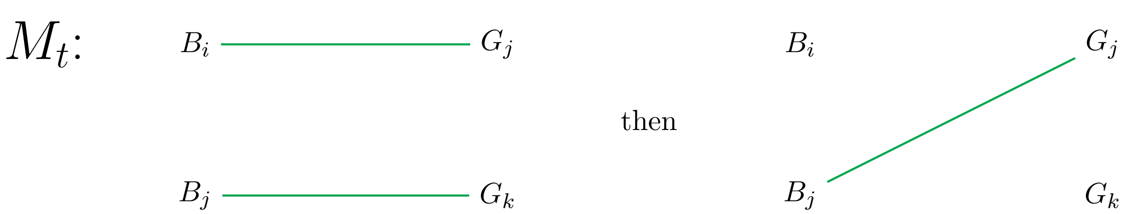


Figure 7: A valid partner being dumped by a girl in boy-proposal

Since B_i is the first valid partner to be dumped, we claim B_j prefers G_j to G_k. Why? Supposed

B_j prefers G_k to G_j. Thus he proposes first to G_k. But $(G_k, B_j) \in M_t$, and therefore G_k is valid for B_j. But B_j was as we supposed in the beginning the first valid person to be dumped, which means B_j did not get dumped and $B - j$ is not free to propose to G_j. $\Rightarrow\Leftarrow$

So B_j prefers G_j to G_k and G_j prefers B_j to B_i, therefore (B_i, G_j) is unstable in M_t. But $M_t \in \mathcal{S}$ and in thus stable. $\Rightarrow\Leftarrow$. Hence a girl never rejects a valid partner.

$\blacksquare$

Now we will show that the boy-proposal algorithm matches each boy B with $G^+(B)$.

Proof: If B_i is matched by algorithm to G_j, who he doesn't like as much as $G^+(B_i)$, then he proposed to $G^+(B_i)$ first. But $G^+(B_i)$ and B_i are valid, hence $G^+(B_i)$couldn't have rejected him. $\Rightarrow\Leftarrow$

Let $B^-(G_j)$ be the worst partner for G_j amongst all stable matchings.

Lemma: The boy-proposal algorithm matches each G_j to $B^-(G_j)$.

Proof: Supposed B_j and G_j are matched, whom she prefers to $B^-(G_j)$. Say $(G_j, B^-(G_j)) \in M_r$ and $(G_i, B_j) \in M_r$ [Figure 8].

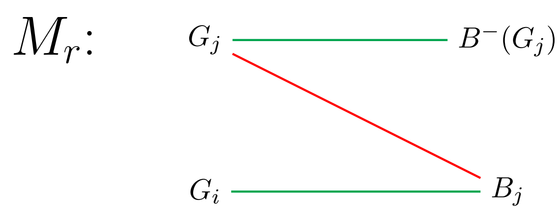


Figure 8: By the previous, B_j gets $G^+(B_j)$, so $G_j = G^+(B_j)$

Thus, B_j prefers G_j to G_i and G_j prefers B_j to $B^-(G_j)$, therefore the valid pair (B_i, G_j) is unstable in M_r. $\Rightarrow\Leftarrow$. It follows that G_j gets $B^-(G_j)$ with boy proposal.

$\blacksquare$

2.3 Matching

A *matching* in a graph $G(V, E)$ is a set $M \subseteq E$ of vertex-disjoint edges, i.e., each vertex of G is the endpoint of at most one edge in M.

we say $v \in V$ is *matched* (or *saturated*) by M if it is the endpoint of some edges in M. Otherwise, it is *unmatched*. A path P is *M-alternating* if its edges are alternatively in M and not in M.

An alternating path is *M-augmenting* if its endpoints are unmatched.

Theorem: A matching in G is of maximum cardinality $\iff$ there is no M-augmenting path.

Proof:($\Rightarrow$) Suppose P is an M-augmenting path, then switching the edges in P produces a larger matching. Let $M' = M \oplus E(P)$ (Symmetric difference of M and the edges in the path P).

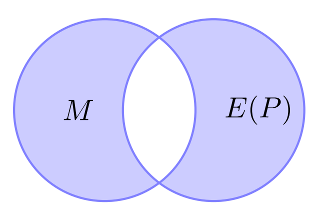
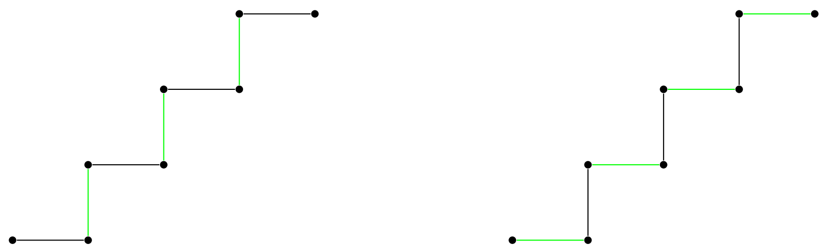


Figure 9: Symmetric difference of M and $E(P)$

$$
\begin{aligned}
M \oplus E(P) &= (M \cup E(P)) - (M \cap E(P)) \\
&= (M - E(P)) \cup (E(P) - M)
\end{aligned}
$$



(a) M has matching of size 3 (b) M' has matching of size 4

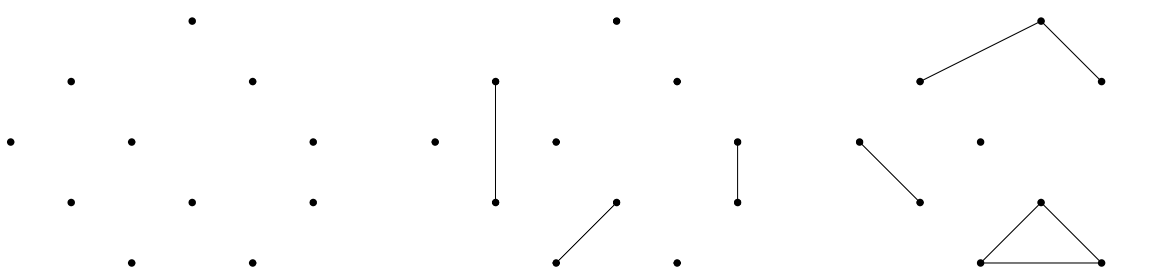
($\Leftarrow$) Suppose M has no augmenting path. Claim that it is a maximum matching. Suppose not, and that M^* is a maximum matching where $|M^*| > |M|$. Consider $M \oplus M^*$. Let H be the subgraph induced by the edges.

Claim:

$$
\begin{aligned}
|M| &= \# \text{ of } M \text{ -edges } \in H + |M \cap M^*| \\
&= \# \text{ of } M^*\text{-edges} \in H + |M \cap M^*|
\end{aligned}
$$

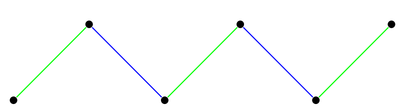
What is the degree of any vertex in H? It's at most two, since each vertex is incident to at most one edge in M and at most one edge in M^*. $deg_H(v) \in \{0, 1, 2\}$.

What does H look like?

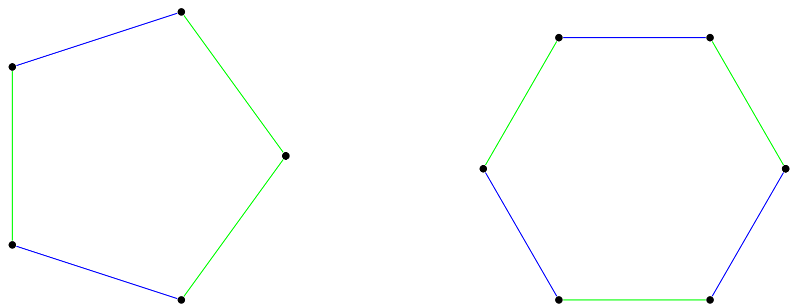


(a) $d_H(v) = 0 \ \forall \ v$ (b) $d_H(v) \in \{0, 1\} \ \forall \ v$ (c) $d_H(v) \in \{0, 1\} \ \forall \ v$

Say blue is M and green is M^*. They alternate:



This means that a cycle must be even:



(a) No (b) Yes

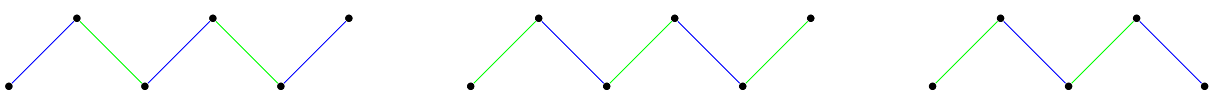
Each component is either

- an even cycle
- a path

Since alternating an even cycle doesn't change the size of M nor M^*, we will focus on paths.

Consider the 3 following types of paths:

1. M^*-augmenting
2. M-augmenting
3. augments nothing



(a) Type 1 (b) Type 2 (c) Type 3

There are no type 1 paths since they are M^*-augmenting and we assumed M^* was maximum! (See $\Rightarrow$ path of the proof). Each type 3 path, similarly to the cycle components, have the same number M and M^* edges. But, by the claim, H must have more M^* edges than M-edges. Therefore there is a type 2 component, and thus is an M-augmenting path. $\Rightarrow\Leftarrow$

■

Note: This theorem holds for all graphs.

2.4 Matching in Bipartite Graph

G is bipartite if there is a partition $V(G) = X \cup Y$, such that each edge has one endpoint in Z and the other in Y (figure 10).

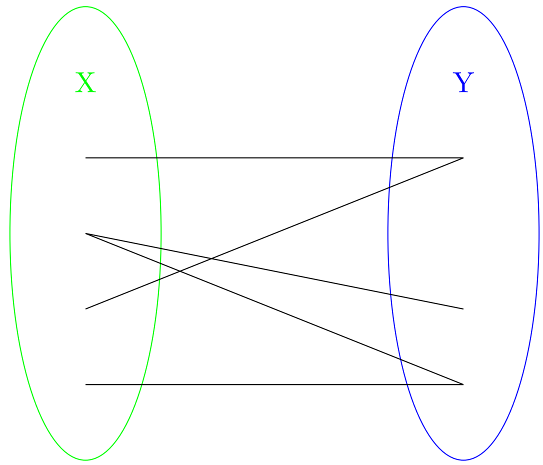
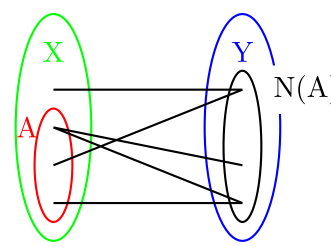


Figure 10: Bipartite graph partitioned into vertex set X and Y

Theorem: G is bipartite $\iff$ it has no odd cycles.



Definition: A matching is perfect if it matches each vertex of G (we can only have degree 1 matching here).

Fundamental question: "When does a graph have a perfect matching?"

Definition: For $A \in V$, denote by $N(A)$ the set of neighbors of A, i.e., $N(A) = \{v \notin A : \exists\, uv \in E, u \in A\}$

Hall's Theorem: A bipartite graph G with $|X| = |Y|$ has a perfect matching $\iff$ $|N(A) \geq |A| \; \forall \, A \subseteq X$. (Known as Hall's condition)

Proof: ($\Rightarrow$) Trivially holds since we can't have this:

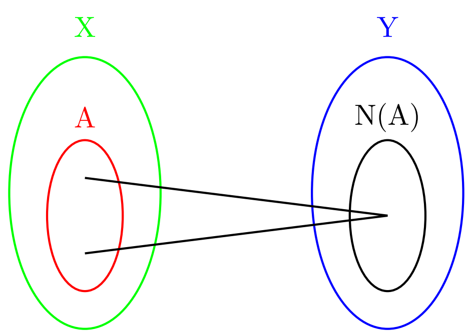


Figure 11: The two vertices in A have only one possible vertex they can match with, therefore there is no perfect matching that would match both.

($\Leftarrow$) Supposed G is satisfying Hall's condition. If M is some matching with an unmatched vertex $u \in X$, we show how to make it bigger.