Matchings in Graphs

Lecturer: *Prajakta* Meeting: 6
Scribe: *Jose Mathew* 11th February 2010

We will be considering finite bipartite graphs. Think of one part of the vertex partition as representing men M, and the other part as women W. So the graph is of the form $G = (M, W, E)$ with $M \cap W = \emptyset$, $E \subseteq M \times W$. Every vertex in G will have a strict set of preferences corresponding to the vertices from the other partition. The idea is to find a pairing that is stable or "divorce-free", as formalised below.

1 Some preliminaries

Definition 1 *Let G = (M,W,E) be a bipartite graph. In addition there is a set of preferences: for each $v \in M \cup W$, $>_v$ is a linear order on the set $N(v)$ of neighbours of v. Let $v \in M \cup W$ and $v_1, v_2 \in N(v)$ be vertices of G. The vertex v* ***prefers*** *v_1 over v_2 if $v_1 >_v v_2$.*

Definition 2 *Let N be a matching in the graph $G = (M, W, E)$, with edges $(m, w), (m', w') \in N$. The pair (m, w') is called a* ***blocking pair*** *if $w' >_m w$ and $m >_{w'} m'$.*

A blocking pair is likely to abandon their respective mates and match up with each other; thus a matching containing a blocking pair is not stable.

Definition 3 *A matching N in G is said to be* ***stable*** *if it does not contain a blocking pair.*

The **stable marriage problem** is the problem of finding a stable matching M of G. This problem involves a set of men and a set of women, each of whom have ranked the members of the other set in an order of preference. In an instance of the classical problem, we assume $|M| = |W| = n$ and the each person has a strictly ordered preference list containing all the members of the other set.

First, let us consider the decision question. We are looking for a perfect matching with no blocking pair; what if no such matching exists? The algorithm we describe below actually cosntructs a stable matching, thus also proving that the answer to the decision question is always Yes.

2 Gale Shapley algorithm for Stable Marriage problem

In 1962, David Gale and Lloyd Shapley proposed an algorithm for solving Stable Marriage problem. The Gale-Shapley (GS) algorithm is as follows.

$\forall m \in M$, m is free. $\forall w \in W, w$ is free.
while some man is free and hasn't proposed to every woman **do**
  Choose such a man m
  w = 1st woman in m's list to whom m has not proposed yet
  if w is free **then**
    w accepts m
  else
    w is engaged to m'
    if $m >_w m'$ **then**
      w accepts m, and m' becomes free.
    else
      w rejects m.
    end if
  end if
end while

Note that the first time a woman receives a proposal, she accepts it. And then she stays engaged: she may move to a more-preferred partner, but she never becomes free. So if at some stage a woman is free, it means that no one has proposed to her so far.

The same is not true for a man. Once engaged, he may find himself jilted and free again. Thus this algorithm is asymmetric in the way it treats men and woman.

Claim 4 *The algorithm terminates in at most n^2 iterations.*

Proof: In each iteration, a man proposes to a woman to whom he has not proposed before. So, there are only n^2 proposals possible. ■

Lemma 5 *The algorithm always outputs a perfect matching.*

Proof: Assume, to the contrary, that when the algorithm terminates, we do not have a perfect matching. Thus $\exists m \in M$, m is free. Since $|M| = |W|$, $\exists w \in W$, w is free. By the discussion following the algorithm, no one has proposed to w during the algorithm. But m has proposed to every woman, including w, a contradiction. ■

Theorem 6 *The algoritm always output a stable matching.*

Proof: Assume to the contrary that the algorithm ouputs a matching N which is unstable. ie $\exists (m, w), (m', w') \in N$, such that (m, w') is a **blocking pair**. But according to our algorithm, m must have proposed to w' before w. Let m'' be the partner of w' immediately after that proposal; either $m'' = m$ or $m'' >_{w'} m$. Since women never move to less preferred partners, either $m'' = m'$ or $m' >_{w'} m$". In all these cases, it cannot be that $m >_{w'} m'$. Thus (m, w') cannot be a blocking pair in N. ■

The asymmetry with respect to men and women in this algorithm translates to the best possible deals for men and the worst possible deals for women. Say that a pairing (m, w) is feasible if it is contained in some stable matching. The Gale-Shapley algorithm matches each man to the first woman on his list with whom the pairing is feasible: it is **men-optimal**. It matches a woman with the last man on her list with whom the pairing is feasible; it is **women-pessimal**. We can prove this formally in two steps.

Lemma 7 *The GS algorithm outputs a stable matching which is men-optimal and women-pessimal.*

Proof: Suppose the resulting matching is not men-optimal. Consider the first iteration where a man is rejected by his optimal partner. Let m be the man, w the woman. Since w rejected m, w is already paired with a man m' such that $m' >_w m$. So m' has proposed to w. Since this is the first bad iteration, m' has not yet been rejected by his optimal partner (say w'), and so $w' <_{m'} w$. Since (m, w) is feasible, there is a stable matching M containing (m, w). And let N be the matching that the GS algorithm ends with.

We can now show that M is in fact not stable. Let w_1 be the partner of m' in M. Since M is stable, the pair (m', w_1) is feasible, and so $w_1 \leq_{m'} w' <_{m'} w$. And we know that $m' >_w m$. Thus (m', w) is a blocking pair. ∎

Using the above Lemma, we can now show the following:

Corollary 8 *The GS algorithm outputs a stable matching which is women-pessimal.*

3 Different variations of Stable Matching

We will now consider different variations of the stable matching problem. Let N be a stable matching of graph $G = (M, W, E)$ and $(m, w) \in M$.

3.1 Egalitarian stable matching

Here we evaluate how good a stable matching is by weighting pairs with their preference positions. We define **men-rank** mr and **women-rank** wr as follows:

$$\begin{aligned} mr(m,w) &= \text{Position of } w \text{ on } m\text{'s preference list.} \\ wr(m,w) &= \text{Position of } m \text{ on } w\text{'s preference list.} \end{aligned}$$

For $(m, w) \in M$, m is happier with smaller $mr(m, w)$, and w is happier with smaller $wr(m, w)$. So to maximize happiness all around, we want these numbers to be small. With this in mind, we define the weight of a stable matching as

$$w(N) = \sum_{(m,w)\in N} mr(m,w) + \sum_{(m,w)\in N} wr(m,w)$$

An **Egalitarian stable matching** is a stable matching that minimizes $\mathbf{w(M)}$.

An egalitarian matching seeks to minimize total (or average) unhappiness.

3.2 Minimum regret stable matching

Another way of evaluating the goodness of a stable matching is by seeing how unhappy anyone is. Let $M(x)$ denote the person paired with x in a matching M, for $x \in M \cup W$. The **Regret** of a person in a matching M is defined as

$$\text{Regret}(m) = mr(m, M(m)) \quad \text{and} \quad \text{Regret}(w) = wr(M(w), w)$$

The **Regret** of matching N is defined as

$$\text{Regret}(N) = \max_{x \in M \cup W} \text{Regret}(x).$$

The **minimum regret stable matching problem** is to find a stable matching N which minimizes **Regret(N)**.

A minimum regret matching seeks to minimize the maximum (or worst-case) unhappiness.

3.3 Stable Matchings with ties

A special case of the stable matchings problem is one in which ties are allowed in the preference list of each vertex. A set W of k women forms a tie of length k in the preference list of man m, if m does not prefer w_i to w_j for any $w_i, w_j \in W$.

Different types of blocking pairs are possible in this case, giving rise to different notions of stability of a matching N. We will define these forms of stability by defining a blocking pair for each case.

1. **Weak Stability:** A matching N is said to be **weakly stable** if it does not contain a blocking pair $(m, w), (m', w')$ such that
$$w' >_m w \quad \textit{and} \quad m >_{w'} m'.$$
2. **Super Stability:** A matching N is said to be **super-stable** if it does not contain a blocking pair $(m, w), (m', w')$ such that
$$w' \geq_m w \quad \textit{and} \quad m \geq_{w'} m'.$$
3. **Strong Stability:** A matching N is said to be **strongly stable** if it does not contain a blocking pair $(m, w), (m', w')$ such that
$$w' \geq_m w \quad \textit{and} \quad m >_{w'} m'$$
$$\textit{OR} \quad w' >_m w \quad \textit{and} \quad m \geq_{w'} m'.$$

4 Stable Roommates problem (SR)

In the stable roommates problem, the graph $G = (V, E)$ is a non-bipartite graph and each vertex $v \in V$ ranks every other vertex in V in strict order of preference. We want to pair up the vertices so that there is no blocking pair.

It is possible that a stable matching does not exist for an instance of stable roommmates problem. Such a situation can occur when a person is rejected by everyone else.

For example, let $V = \{A, B, C, D\}$ and let the preference lists be $L(A) = \langle B, C, D\rangle$, $L(B) = \langle C, A, D\rangle$, $L(C) = \langle A, B, D\rangle$, and $L(D) = \langle A, B, C\rangle$. Consider the matching $\{(A, D), (B, C)\}$; the other cases are symmetric. (A, C) forms a blocking pair.

Algorithm for finding stable matching in SR

Let $x, y \in V$ be two vertices in graph G. The algorithm is as follows:

1. If x receives a proposal from y, then
   (a) x rejects if x already holds a proposal from someone higher than y in x's preference list
   (b) Otherwise, x holds the proposal for consideration and rejects any other lower proposal currently held.
2. An individual x proposes to others in the order in which they appear in his/her preference list, stopping when a promise of consideration is received. Any subsequent rejection of x causes x to continue with his/her sequence of proposals.

The above mentioned algorithm will terminate either

1. with every individual holding a proposal, OR
2. with one individual rejected by everyone (No stable matching in this case).