# Matchings in Graphs

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

We will be considering finite bipartite grahs, denoted as $G = (M, W, E)$. Every vertex in $G$ will have a strict set of preferences corresponding to the vertices from the other partition.

## 1 Some preliminary definitions

**Definition 1** *Let $G$ = $(M,W,E)$ be a bipartite graph. Let $v, v_1, v_2 \in M \cup W$ 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 of graph $G = (M, W, E)$. Let two edges $(m, w), (m', w') \in N$. The pair $(m, w')$ is called an* ***blocking pair*** *if $w' >_m w$ and $m >_{w'} m'$.*

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

**Definition 4** *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.*

## 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)

Choose such a man $m$

$w$ = 1st woman in $m$'s list to whom $m$ has not proposed yet

**if** $w$ is free

$w$ accepts $m$

**else if** $w$ is engaged to $m$'

**if** $m >_w m'$

$w$ accepts $m$. $m$' becomes free.

**else** $w$ rejects $m$

**Remark 5** *Once a woman is engaged, she never becomes free.*

**Claim 6** *GS algorithm will terminate in n²* *iterations.*

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

**Lemma 7** *GS algorithm always outputs a perfect matching.*

**Proof:** Assume we doesn't get a perfect matching after the completion of GS algorithm. $\exists m \in M$, $m$ is free. Then $\exists w \in W$, $w$ is free. By remark earlier, $w$ was never proposed. But $m$ proposes to every woman and so we get a contradiction. ■

**Theorem 8** *GS algoritm always output a stable matching.*

**Proof:** Assume GS algorithm ouputs a matching $N$, which is unstable. ie $\exists (m,w),(m',w') \in N, s.t.(m,w')$ is a $blockingpair$. But according to our algorithm, $m$ must have proposed $w'$ before $w$. If $m$ had proposed $w'$, $w'$ must have retained $m$ as the partner ($m >_{w'} m'$). This contradicts the fact that $(m,w')$ is a $blockingpair$. Hence N will not contain any blocking pair and it will be a stable matching. ■

**Lemma 9** *GS algorithm outputs a stable matching which is men-optimal and women-pessimal.*

**Proof:** Men will start proposing women in the decreasing order of their preferences. In this stable matching, every man will get the best partner in any stable matcing. This comes at the expense of women and each woman will have the worst partner she can have in any stable matching. ■

# 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

$mr(m, w)$ = Position of $w$ on $m$'s preference list.

$wr(m, w)$ = Position of $m$ on $w$'s preference list.

The weight of a stable matching is defined as,

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

An *Egalitarian stable matching* is the stable matching which **minimizes w(M)**.

## 3.2 Minimum regret stable matching

Regret of a person is defined as,

$$Regret(m) = mr(m, w) \quad and \quad Regret(w) = wr(m, w)$$

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

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

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

## 3.3 Stable Matchings with ties

We will consider a special case of the stable matchings problem which allows ties 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 and it will give rise to different notions of stability of the matcing 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 a **weak stable matching** if it does not contain a blocking pair $(m, w), (m', w')$ such that,

$$w' >_m w \quad and \quad m >_{w'} m'$$

2. **Super Stability:** A matching $N$ is said to be a **super stable matching** if it does not contain a blocking pair $(m, w), (m', w')$ such that,

$$w' \geq_m w \quad and \quad m \geq_{w'} m'$$

3. **Strong Stability:** A matching $N$ is said to be a **strong stable matching** if it does not contain a blocking pair $(m, w), (m', w')$ such that,

$$w' \geq_m w \quad and \quad m >_{w'} m'$$
$$OR \quad w' >_m w \quad and \quad m \geq_{w'} m'$$

# 4 Stable Roommates problem (SR)

In 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.

**Remark 10** *It is possible that a stable matching does not exist for an instance of stable roommmates problem. It occurs when a person is rejected by everyone else.*

## 4.1 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 it already holds a proposal from someone higher than $y$ in $x$'s preference list

   (b) Otherwise, $x$ holds it for consideration and rejects any other lower proposal he currently holds.

2. An individual $x$ proposes to others in the order in which they appear in his preference list, stoping when a promise of consideration is received. Any subsequent rejection of $x$ causes $x$ to continue his sequenxce 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)