MS&E-212 Combinatorial Optimization
Instructor: Professor Amin Saberi (saberi@stanford.edu)

Lecture 10: Marriage, Stability and Honesty

1 Stable Matching

This lecture is on stable matching problem. Consider a community with a set of n men, M, and a set of n women, W. Each man, m, has a ranking of women representing his preferences i.e., if in m's list, woman w comes before woman w', it means that m prefers to marry w rather than w'. Similarly each woman w has a ranking of her preferred men. The *stable marriage* problem asks to pair (match) the men and women in such a way that no two persons prefer each other over their matched partners. More formally:

Definition: A **matching**, P_M, is a one-to-one mapping from M to W (or equivalently we can define a matching as a one-to-one mapping from W to M).

Definition: A pair (m, w) is a **rogue pair** iff

1. m prefers w to his matched partner $P_M(m)$.

2. w prefers m to to her matched partner m' $(w = P_M(m'))$,

A matching is **stable** if it doesn't have a rogue pair.

Example: Assume that we have $M = \{A, B, C\}$, and $W = \{1, 2, 3\}$ with preferences (rankings) given by,

$$
\begin{array}{ll}
A: & 123 \\
B: & 213 \\
C: & 321
\end{array}
\qquad
\begin{array}{ll}
1: & BAC \\
2: & ABC \\
3: & CBA
\end{array}
$$

Let matching P_M be as follows:

$$P_M(C) = 1 \quad P_M(B) = 2 \quad P_M(A) = 3$$

It is easy to see that the matching above is not stable since $(A, 1)$ is a rogue pair. But the following matching has no rogue pairs $(1, 3, A, C$ get their first choices), and so is stable.

$$P_M(A) = 1 \quad P_M(B) = 2 \quad P_M(C) = 3$$

Question: How can we find a stable matching in general?

Gale and Shapley, in 1962, proposed the Deferred Acceptance Algorithm (1). Here, we assume that each m (w) ranks all the possible women (men), i.e. m's (w's) list is complete.

Claim 1 *The GS algorithm terminates.*

Proof: A man is rejected at most m times, each time removing a woman from his preference list. There are n such lists, thus it takes at most nm steps. ∎

Claim 2 *GS algorithm produces a perfect matching whenever $m = n$.*

Algorithm 1 Defered Acceptance [Gale-Shapley] (men-proposing version)

let all men and women be unmatched.
repeat
  unmatched man m proposes to the most preferred woman w in his list
  if w is unmatched **then**
    match w and m (they become engaged)
  else if w is matched to m', but she prefers m to m' **then**
    match (engage) w to m and leave m' unmatched.
  else
    m removes w from his preference list
  end if
until each m has been matched or has reached the end of his list (i.e., has an empty list)
return matching M of all engaged pairs $(m, P_M(m))$

Proof: Suppose there is some unmatched man m. Since $m = n$ there exists some unmatched woman w. Since m is unmatched he has been rejected by *all* women. Since w is unmatched she has never rejected a proposal. This is a contradiction since w is in some position in m's preference list. ∎

Claim 3 *The matching found by the GS algorithm is stable.*

Proof: Consider pair (m, w) where $P_M(m) \neq w$. According to GS algorithm, two scenarios are possible:

  1. m has proposed to w and was rejected; this means w prefers her current partner to m.

  2. m has not proposed to w, then m prefers his current partner to w

neither of the above scenario results in (m, w) being a rogue pair. So there cannot be a rogue pair. ∎

Theorem 1 *The men-proposing algorithm is* man-optimal*: every man will be matched to the* best *partner he could be matched to in any stable matching.*

Proof: By contradiction.
Denote M as the matching produced by Algorithm 1. Consider the **first** event, E^*, that a man is rejected by a woman in the algorithm. Label this man m, and the woman w. m is rejected because w prefers m' to m. Suppose there exists some other stable matching $\widetilde{M}$ such that m and w are matched i.e. $m = P_{\widetilde{M}}(w)$. Suppose m' is matched to w' in $\widetilde{M}$ ($w' = P_{\widetilde{M}}(m')$). Consider two cases:

Case I: m' **prefers** w **to** w',

$\widetilde{M}$ contains the pairs (m, w) and (m', w'), however (m', w) is a rogue pair. This is contradicting with the assumption that $\widetilde{M}$ is stable.

Case II: m' **prefers** w' **to** w,

Going back to the algorithm; m' prefers w' to w but it is matched to w at the time that event E^* happens. This means that m' was rejected by w' before proposing to w. This is contradicting with the assumption that m is the first man that gets rejected in the run of the algorithm.

∎

Theorem 2 *The men-proposing algorithm is* female-pessimal*: every woman will be matched to the* worst *partner she could be matched to in any stable matching.*

Proof: Again by contradiction; let M be the output of Algorithm 1. w is matched to m in M. Suppose w prefers m to m' and there exists another stable matching $\widetilde{M}$ in which w is matched to m' and m is matched to another woman w'.

By man-optimality of M (Theorem 1), m prefers w to w'. Matching $\widetilde{M}$ contains (m, w') and (m', w), however, (m, w) is a rogue pair. This is contradicting with the assumption that $\widetilde{M}$ is stable. ∎

2   Extensions and Applications

Generalization to stable marriage presents some difficulties. In more realistic scenarios we can expects ranking lists to be incomplete (one may rather be alone than with someone insufficiently qualified), and to model indifference between subsets of partners. While each of these are solvable in polynomial time separately, solving incomplete lists with indifferent preferences together makes the problem NP-hard.

Another issue of concern is whether there are any incentives to change preference lists. In game theory, a game is called *strategyproof (thruthful)*, if players have no incentive to hide information from each other. Could someone state false preferences to gain a benefit from the match? For men, it is trivially truthful, but there is no incentive for women to be truthful! One can design cases where changing the list of preferences will increase the happiness of a given woman with the final matching results.

Furthermore, in the real world matching with preferences is usually not one to one. Many to one marriage problems, commonly referred to as the college admissions model is famously employed in several entry level professional labor markets such as the National Residency Matching Program (NRMP). The college admission model assumes some population of students, a number of colleges with certain student capacities and preferences stated by the colleges and the students.

NRMP has been in effect since the 1950's and has experimented with different matching models to match medical school graduates to hospitals as they enter residency programs. For a long time the system was run as *hospital optimal*. For this reason, the system was sued for being anti-competitive; the system held up, but was switched from being hospital optimal to resident optimal. Interestingly, this switch didn't change the rankings much, because rankings are highly correlated.

Rules for the algorithm used to conduct the matching for NRMP have been updated many times through the years as complications arose. One of the major ones was the introduction of couples, that is an application to a hospital coming not from a single student but from a married couple. Economist, Al Roth, proved that introducing couples into the match leaves a possibility that no stable match exists.