Lecture Notes for B503, Fall 2019

Qin Zhang

1 The Stable Matching Problem [KT 1.1]

In this problem, we have n men and n women. For each man, there is a list of preference over all women given as input. For each woman, there is a list of preference over all men is also given as input. The goal of this problem is to produce a one-to-one matching between men and women so that the matching is stable. A stable matching is one with no unstable pairs, where an unstable pair is a pair of unmatched man and woman (say X and A) so that X prefers A to his current partner and A prefers X to her current partner.

A natural idea we start with is to begin with an arbitrary matching. Then we exam all pairs of unmatched man and woman, and decide whether there is an unstable pair. If none exists, we return this matching as a stable matching. If we do find an unstable pair, we can allow this pair to "hook up" and the two dumped people forms a new pair of partners, so that we fix this particular pair and come up with a new matching. We can do the same thing for the new matching and keep doing this until a stable matching is found. When such procedure terminates, we can confidently say that the returned matching is stable. However, there are examples where this procedure can loop forever, which is problematic.

Question 1 *Give an example (input) such that the above algorithm will loop forever.*

We now turn to the propose-and-reject algorithm. In this algorithm, we start with all men and women single, and iterate the following procedure whenever there is a single man.

– Pick a single man, namely X. Among all the women X has not proposed to, he proposes to the first woman in his preference list. Call this woman A.

– If A is single or A prefers X to her current partner, she accepts X's proposal (and dumps her current partner if she is not single).

This procedure can be iterated for at most n^2 times, because each man will propose to a woman at most once.

However, when algorithm stops, does it come up with a stable matching? Does it even come up with a matching? We prove affirmative answers to these two questions. We show the following claims which will be useful.

Claim 1 *As the algorithm runs, once a woman gets matched, she will remain matched for the rest of the algorithm, and her partner only gets better and better (in terms of her preference list).*

This claim can be easily verified according to the description of the algorithm.

Claim 2 *The algorithm will end up with a matching.*

Proof: Assume the opposite, i.e. there is a single man (namely X) who has proposed to all n women. In this case, for each woman A, when X proposes to A, A becomes matched. By Claim 1, A will be matched for the rest of the algorithm. Therefore, when X has proposed to all n women, all n women are matched. Since there are only n men, X cannot be single – a contradiction. $\square$

Claim 3 *The matching returned by the algorithm is stable.*

Proof: Again we prove this claim by contradiction. Assume that the matching is not stable, i.e. there exists an unstable pair (X, A). In the matching X is matched with B, Y is matched with A, however, X prefers A to B and A prefers X to Y.

By the order of proposing, X would have proposed to A before proposing to B. Since X proposed to B (as he is matched with B), he must have proposed to A. When X proposed to A, A should have either accepted X or rejected X because she was partners with a better man than X (according to her list). However, when the algorithm terminates, A is partnered with Y – who is worse than X in her list. This is a contradiction to Claim 1. $\square$

Running time: We have shown that the algorithm terminates; and when it terminates, it correctly outputs a stable matching. We now consider the time complexity of the algorithm and implementation details. As we saw before, the iteration goes by at most n^2 times. If we can make each iteration run in $O(1)$ time, the time complexity of the algorithm would be $O(n^2)$. To achieve this, we need to take care of the following two issues.

– In order to pick a single man in $O(1)$ time, we need to maintain a queue of single men. At first, all men are in the queue. Whenever we need to pick a man, we just Dequeue an element from the head of the queue. When a woman dumps a man so that the man becomes single again, we Enqueue this man to the tail of the queue.

– In order to compare two man X and Y according to the preference of a woman A. We need to construct a rank table, namely $rank[A][]$. $rank[A][X]$ contains the rank of X in A's preference list. Now we can test whether A prefers X to Y by comparing $rank[A][X]$ and $rank[A][Y]$. Of course, we need to build the $rank[A]$ table – this takes $O(n)$ time for one woman A, and overall it take $O(n^2)$ time to build the whole $rank[][]$ table.

Fairness: We finally observe that we can also run the algorithm by letting women propose. However, we find that all men get better partners (in terms of their preference) in the matching returned by the algorithm with men proposing (compared to women proposing). Similarly, all women get better partners in matching returned by the algorithm with women proposing.

Claim 4 *If men propose, every man gets his best valid partner (according to his own preference) by the algorithm. Here A is X's valid partner if there exists a stable matching such that X is matched with A.*

Proof: Suppose this is not true. Then let X be the first man rejected by his valid partner A. Since this is the first rejection, A must be X's best valid partner (as X proposed in his preference order). When A rejects X, let Y be A's partner. So we know that A prefers Y to X.

Since A is X's valid partner, we can fix a stable matching M such that A is partnered with X in M. Let us also mention the partner of Y in M, and give her name B. Now we discuss the following 2 cases.

1. Y prefers A to B. Then A and Y forms an unstable pair in M. Contradiction to the stability of M.

2. Y prefers B to A. Y is A's partner when A rejects X. Since Y proposes in order, Y must have proposed to B before (and must have been rejected by B). This means that Y was rejected by his valid partner B before X was rejected. This is a contradiction to our assumption that X is the first man rejected by his valid partner.

In both cases we reached a contradiction. And this finishes the proof. □

Similarly we can also prove the following claim, the proof of which is left as an homework.

Homework 1 *If men propose, every woman gets her worst valid partner (according to her own preference) by the algorithm.*