CSCI 570 - Fall 2021 Homework 1

Rajat Kumar Singh (rajatsin@usc.edu)

3rd September, 2021

Q 1. Reading Assignment: Kleinberg and Tardos, Chapter 1

Done. All sections read.

Q 2. Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.
True or false? In every instance of the Stable Matching Problem, there is a stable matching containing a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

False. The statement is incorrect since there are stable matchings in with the above property does not hold true. Consider the following example.

m prefers w to w'
m' prefers w' to w
w prefers m' to m
w' prefers m to m'

For the given priority preferences there exists a stable matching consisting of (m,w) and (m',w'). In such a matching, the men are paired with women who are their first preferences. Women are paired with their worst preferences. The matching (m,w') and (m',w) also forms a stable matching. The women are matched with their first preferences but the men are paired with their worst preferences. Since no other perfect matching without instabilities exist, we have provided an example where the statement does not hold true.

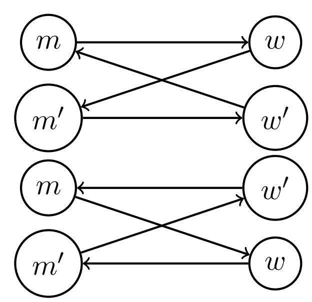

Figure 1: Stable matchings

Q 3. Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.
True or false? Consider an instance of the Stable Matching Problem in which there exists a man m and a woman w such that m is ranked first on the preference list of w and w is ranked first on the preference list of m. Then in every stable matching S for this instance, the pair (m, w) belongs to S.

True. The statement is true and we can prove that using contradiction. We can infer the preference sequences of a man m and a woman w on the basis of the given statement.

m prefers w to w'

w prefers m to m',

where $m' \in M(listofallmen)$ $m' \neq m$

and $w' \in W(listofallwomen)$ $w' \neq w$

Let us assume that the the statement is false and we have a stable matching with (m,w') and (m',w). Since m prefers w to w' and w prefers m to m', there is an inconsistency in the matching. Since m' and w' are generic to all other men and women respectively, we can come to the conclusion that there is no stable matching which does not have the pair (m,w), otherwise there would be an inconsistency always between m and w.

Q 4. State True/False: An instance of the stable marriage problem has a unique stable matching if and only if the version of the Gale-Shapely algorithm where the male proposes and the version where the female proposes both yield the exact same matching.

True.

First, we prove that the Gale-Shapley algorithm leads to the same matching S* regardless of the order of men. In this matching, the men are paired with women who are their best valid partner. A partnership is said to be valid, if the pairing does not form instabilities in the resultant set. Let us assume that our original matching S' consists of the pairs (m,w) and (m',w') and is a stable matching. Let us assume that we get a resultant matching in which a man m is paired with w' who is not his best valid partner. Preference for w' is lower than that of a woman w in m's preference list. Since in the GS algorithm, men propose based on their preference, it means that m already proposed to w. w must have either rejected m's proposal for a higher preference man m' or broke off her engagement to m during the execution on algorithm. Let us assume (m,w) is the first engagement to break. In this matching ,we get (m',w) as a pair. w prefers m' over m since it decided to marry m'. m' prefers w over w' since it proposes to w first. Otherwise, m' would have married w' rather that w. But since we know w prefers m' over m, there is an instability in our original matching S'. Hence we only get one solution for all GS executions where men propose first.

Secondly, same can be proved for GS executions where women propose first. Women end up with their best valid partners using the same proof as shown above(since it is symmetric). Finally, we can prove that if GS executions for men proposals and women proposals lead to the same matching, we have a unique stable matching. Let us assume that there is another matching M' which is different that the matching M we get from GS algorithm(both men proposal and women proposal). Since we get the best case matching for men as the same as the best case matching for women, we can infer that the stable matching solution is unique. If there is a pair(m,w) in this matching M', we cannot change m's pairing to a weaker preference, since it violates the best match property(men proposing first). If women propose first then (m',w) can only be formed if w prefers m' over m. But this is also a contradiction since we know (m,w) is the best preference from w's point of view and (m',w) violates the best match property. Hence stable matching must be unique since other pairings are not possible.

Q 5. A stable roommate problem with 4 students a,b,c,d is defined as follows. Each student ranks the other three in strict order of preference. A matching is defined as the separation of the students into two disjoint pairs. A matching is stable if no two separated students prefer each other to their current roommates. Does a stable matching always exist? If yes, give a proof. Otherwise give an example roommate preference where no stable matching exists.

The statement is false. Take for example preference lists for a,b,c,d as follows:-

a:-b,c,d
b:-c,a,d
c:-a,b,d
d:-c,a,b

In this case we cannot have the following matchings.
1. (a,b)(d,c) :- Instability between b and c
2. (a,c)(b,d) :- Instability between a and b
3. (a,d)(b,c) :- instability between a and c

Q 6. There are many other settings in which we can ask questions related to some type of "stability" principle. Here's one, involving competition between two enterprises.
Suppose we have two television networks, whom we'll call A and B. There are n prime-time programming slots, and each network has n TV shows. Each network wants to devise a schedule—an assignment of each show to a distinct slot—so as to attract as much market share as possible. Here is the way we determine how well the two networks perform relative to each other, given their schedules. Each show has a fixed rating, which is based on the number of people who watched it last year; we'll assume that no two shows have exactly the same rating. A network wins a given time slot if the show that it schedules for the time slot has a larger rating than the show the other network schedules for that time slot. The goal of each network is to win as many time slots as possible.
Suppose in the opening week of the fall season, Network A reveals a schedule S and Network B reveals a schedule T. On the basis of this pair of schedules, each network wins certain time slots, according to the rule above. We'll say that the pair of schedules (S, T) is stable if neither network can unilaterally change its own schedule and win more time slots.
That is, there is no schedule S' such that Network A wins more slots with the pair (S',T) than it did with the pair (S,T) and symmetrically, there is no schedule T' such that Network B wins more slots with the pair (S,T') than it did with the pair (S,T).
The analogue of Gale and Shapley's question for this kind of stability is the following: For every set of TV shows and ratings, is there always a stable pair of schedules? Resolve this question by doing one of the following two things:
(a) give an algorithm that, for any set of TV shows and associated ratings, produces a stable pair of schedules; or
(b) give an example of a set of TV shows and associated ratings for which there is no stable pair of schedules.

We are going to show that stable matching might not exist for the given scenario. Take an example of n=2 and ratings A = 50,100 and B = 20,70. Network A can release its list of schedules as 100->50 and B can release its schedule as 20->70. Since A wants to win more time slots, it will swap its shows as 50->100. At this point of time, A wins both slots. B swaps its shows to 70->20 to win 1 slot. A can swap its shows again to get more shows(since it only has one). Equilibrium is not achieved and this entire cycle continues and there is no stable pair of schedules.
B will always want to win 1 slot and A will want to win both slots. This will lead to instability in the schedules.

Q 7. Gale and Shapley published their paper on the Stable Matching Problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals. Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals. The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.)
We say that an assignment of students to hospitals is stable if neither of the following situations arises.
First type of instability: There are students s and s', and a hospital h, so that
– s is assigned to h, and
– s' is assigned to no hospital, and
– h prefers s' to s
Second type of instability: There are students s and s', and hospitals h and h', so that
– s is assigned to h, and
– s' is assigned to h', and
– h prefers s' to s, and
– s' prefers h to h'.
So we basically have the Stable Matching Problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students. Show that there is always a stable assignment of students to hospitals, and give an algorithm to find one.

We can provide the following algorithm for stable assignment of students to hospitals. The algorithm is similar to Gale Shapley wherein a student is either engaged to a hospital or is free. The following is the pseudo code for the proposed algorithm

While any hospital exists with empty positions
  Pick a hospital h_i with empty positions
  h_i offers the next student s_j in its preference list the job whom it has not yet offered
  If s_j is free then
    s_j accepts the offer
  Else
    If s_j prefers h_k over h_i where s_j is currently filling h_k's job offer
      s_j remains committed to h_k
    Else
      Increase positions at h_k with 1 and decrease positions at h_i by 1
      Job position at h_i is now assigned to student s_j

This is only possible if any of the hospital h_i has already offered all students a job offer and there are no students left. Let the total number of students be *sum(S)* and total number of jobs be *sum(J)*. If there are no students left to offer the job, it means *sum(J)* $\geq$ *sum(S)*, which is a contradiction since we have assumed that there are more students than jobs. Once a job position is offered to a student, the student remains employed throughout the execution, although it might be to a preferred hospital. One student cannot be offered multiple jobs by this logic, and hence the matching is perfect.
We can prove that a stable solution always exists via contradiction. Let us assume a stable matching is not possible because of the following conditions of instability:-
1. *Student s' preferred to s by hospital h but (h,s) present*
Since h prefers s' to s, it would have offered s' the job earlier. If s' broke the engagement, then it would have ended up with a better engagement rather than no engagement, hence a contradiction. If h proposed to s first, that is a contradiction in itself since s' is ranked higher that s by hospital h.

2. *Instability exists in any of the assignments:-* Let us assume the pair (h,s) and (h',s') in a stable matching M'. Then h must offered the position to s based on its preference list, but this is a contradiction since h prefers s' over s. If h offered the position to s' and is paired, then at some point h' would have offered the position to s and s would have accepted it. But since we know that s' prefers h to h', this would cause an instability in M' which contradicts our assumption. Hence this instability does not exist.

Q 8. N men and N women were participating in a stable matching process in a small town named Walnut Grove. A stable matching was found after the matching process finished and everyone got engaged. However, a man named Almazo Wilder, who is engaged with a woman named Nelly Oleson, suddenly changes his mind by preferring another woman named LauraIngles, who was originally ranked right below Nelly in his preference list, therefore Laura and Nelly swapped their positions in Almanzo's preference list. Your job now is to find a new matching for all of these people and to take into account the new preference of Almanzo, but you don't want to run the whole process from the beginning again, and want to take advantage of the results you currently have from the previous matching.Describe your algorithm for this problem.Assume that no woman gets offended if she got refused and then gets proposed by the same person again

---

Let us assume the following pairs are present in the original matching (Almanzo,Nelly) and (X,Laura). Since Almanzo has changed his preference, we break the engagement between Almanzo and Nelly and make Almanzo a free man and assume that neither Nelly nor Laura has been proposed by Almanzo yet. We can propose the following algorithm for the refined stable matching process.

While any man m remains free(Initially starts with Almanzo)
  Choose a woman w next in the man's preference list who he has not proposed to yet
  If w prefers her current partner to m
    Don't change anything
  If w prefers m over her current partner m'
    Break engagement with m' and form new pair (m,w). m' is now free

The algorithm starts with Almanzo proposing to Laura since she is higher in the preference list. If Laura is prefers X to Almanzo, the engagement is not broken and Almanzo moves on to propose to Nelly. Since Nelly is not married to anyone(Almanzo was her initial partner), Almanzo forms a pair with Nelly and our algorithm stops since there is no free man. On the other hand, if Laura prefers Almanzo to X, she breaks her original engagement and forms a pair with Almanzo. X is now added to free men list and Almanzo is removed.The matching is carried on till there is no free man. We can prove that the algorithm always ends with a stable matching. If we assume that a man remains free at the end of our algorithm, it means that he has proposed to all the women and is rejected by them. This means there are already n men paired with the n women, which means there would be n+1 men, which is a contradiction. HEnce we prove that the algorithm always ends with no free men.