CSCI270 WEEK 1

Jacob Ma

September 28, 2022

1 Stable Matching

Example 1.1. We have universities and student. Each student has a ranking (total order) over universities; Each university has a ranking over students. (Different individuals may have different rankings.)
Goal: Find a "good" assignment of students to universities.
Assumption: Algorithm has access to all the information. (All the rankings, and number of slots at each university).
Other applications:

(1) Jobs and applicants.

(2) Man and woman.

(3) Owners to pets.

(4) medical residents to med school grads.

(5) student and elementary/high/middle schools

Common themes:

(1) two types of entities

(2) match them up

(3) "good" with respect to the rankings

(You can also define this problem for just one type of entity: for example, roommates, homosexual dating)

Definition 1.2: What should "good" mean?

Assumption: The total number of slots is equal to the total number of applicants.
Potential definitions:

(1) Minimize the sum of positions of partners in everyone's ranking?

(2) Stability: there is no pair that would prefer to deviate from the prescribed assignment together.

Definition 1.3: Stability

A matching is **stable** if for each pair (u, s) such that u and s are not assigned to each other, at least one of the following two is true:

(1) s prefers their assigned university over u

(2) u prefers all of their assigned students over s

If the algorithm's assignment were not stabler, then "backroom deals" might happen, and thy might set off other backrooms deals.

Implication we will make: everyone gets one darner only. And there is same number of both. $\implies$ standard nomenclature: men and women.

Definition 1.4: Perfect Matching

Matching: Graph in which each node is incident on at most one edge.
Perfect Matching: Graph in which each node is incident on exactly one edge.
$\implies$ We are looking for a stable perfect matching in a given bipartite instance with ranking.
Bipartite: Nodes are divided into two sets - edges only exist between the two sets, but not inside either of them.

Example 1.5: Man and Women (Simplified). n men, n women. Each has a preference order over the other set. Find an assignment where each man is assigned exactly one woman, woman exactly one man, and the assignment is stable.
Did we make the problem too easy? How much did the assumption of only one partner (and same number of men/women) simplify the problem.

Example 1.6. Not a lot:

(1) To deal with universities having multiple slots, for a university with k slots, we can generate k copies USC(1), USC(2), ..., USC(k). Each of the USC(i) has the same preference ranking as the original USC. Students all copies of USC in arbitrary order, in the position where they originally had USC. $\implies$ Can prove that this reduction works.

(2) What if the number of men and women is not the same?

If there are k more women then men, create k "dummy men" who everyone ranks at the bottom. $\implies$ Can prove this works.

We just saw our first two reductions. They showed that if we can solve the problem with same number of men/-women, we can solve the more general version.
We were able to give these reductions even though we don't know how to solve either problem (yet).
Here, it justifies focusing only on the version with men/women and the same number of both.

Other modeling simplifications we had made:

(1) everyone prefers being matched over unmatched $\implies$ can also be solved with dummy nodes.

(2) ties?

(3) strength of preferences? $\implies$ very different problem.

(4) does everyone really know their ranking?

(5) might people want to lie?

Problem 1: Uniqueness of stable matching

Is the stable matching always unique? Or could there be different stable matching for the same input?

There are men A, B, and women $1, 2$.

A: 1 > 2

B: 2 > 1

1: $B > A$

2: $A > B$

Both $\{(A, 1), (B, 2)\}$ and $\{(B, 1), (A, 2)\}$ are stable.

Theorem 1.7: Gale-Shapely Proposal Algorithm

- Maintain temporary assignments of "engagements"
- Start with the empty assignment.
- Until there is no single man:
  - Let m be any single man.
  - Let w be the highest-ranked woman (according to his list) to whom he has never "proposed".
  - If w is a single (unmatched) or prefers m over her current partner, then
    - match m and w (temporarily, engagement)
    - if w was previously matched, her old partner becomes single
  - Otherwise, do nothing (but m will never propose to w again)
- Finalize the temporary assignment.

Lemma:

(1) Once a woman is matched for the first time, she will never become single again, and her matches can only improve(according to her ranking overtime).

(2) The algorithm terminates.

Proof: If not, then there would still be a single man forever. Because number of men = number of women, there would be a single woman. This means she was never proposed to. But then, the single man would propose to her $\implies$ Contradiction.

(3) We get a perfect matching at termination.

Proof: At termination, there is no single man. Therefore, there is no single woman (Same number of both).

Proposition: The final matching is stable.