CSCI270 WEEK 2

Jacob Ma

September 28, 2022

1 Matching

Theorem 1.1: 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 1.1.1

The algorithm terminates.

Proof. Each proposal is made only (at most) once, because men go down the list. There are at most n^2 proposals total (each man for each woman). So there can be at most n^2 iterations.
If the algorithm didn't terminate, some man m would be single at the end.
Because number of men = number of women, there would be single woman w as well.
Because m exhausted all his proposals, he must have proposed to w. But by previous lemma, w can then not end up single $\implies$ Contradiction.

□

Lemma 1.1.2

We get a perfect matching at termination.

Proof. At termination, there is no single man. Number of men = number of women, so all women are matched. Therefore, there is no single woman (Same number of both). □

Lemma 1.1.3

Once a woman is proposed to, she will be partnered until the end. Her partners only improve over time (because she can reject ones she does not like better).

Lemma 1.1.4

The final matching is stable.

Proof. By contradiction, we assume the output is not stable.
Then, there exists a man m and a woman w when are partnered by the algorithm with w' and m', but would prefer each other.
So m must have proposed to w', which by the proposal order means that m proposed to w earlier, and either got reject immediately or dumped later.
Either way, at the time m got rejected/dumped, w had some man m'' she liked better m.
By the lemma above (things only improve for women), she likes m' at least as much as m'' (or they are equal).
So she likes m' better than m. Contradiction to the assumption that w likes m better than m'. QED □

Theorem 1.2

GS always terminates, and outputs a stable perfect matching. (It takes at most $O(n^2)$ iterations)

Problem 1

Does it matter which man m proposes next in GS?

Solution

Answer happens ton be no, and we will characterize which matching is produced.
A: 1 > 2
B: 2 > 1
1: $B > A$
2: $A > B$
Stable Matching:

$$\{(A,1),(B,2)\},\{(B,1),(A,2)\}$$

We define $P(m)$ (*possible matchings for m*) = the set of all women w such that m can end up with w in at least one stable matching.

$$P(A) = \{1,2\}, P(B) = \{1,2\}$$

We define $b(m)$ (*best possible match for m*) = the highest-ranked w in $P(m)$ according top $m's$ ranking:

$$b(A) = 1, b(B) = 2$$

Theorem 1.3

Gale-Shapely returns a matching in which each man m is matched up with $b(m)$.

Corollary 1.4

Matching each man m with his $b(m)$ is actually a matching, so no two men have the same best possible partner.

Proof. By contradiction. Assume that some man m is not matched up with his $b(m)$.
Then, at some point, he must have been rejected/dumped by his $b(m)$.
Look at the very first time that some man got dumped/rejected by a woman in $P(m)$. Let m be that man, and w the woman from $P(m)$ who rejected/dumped him.
Therefore, the man m' that w rejects/dumps m for in GS is strictly higher on her list than m. (Using our earlier lemma)
Because w was a possible choice of m, there exists some other stable matching M' in which m and w end up with each other.
Who is m' matched with in M'? Let call her w'.
Who does m' prefer between w and w'?
Since we know that w prefers m' over m, if m' preferred w over w', M' would not be stable - (m', w) would prefer being with each other.
So we infer that m' actually prefers w' over w.
In GS, m' ended up paired with w.
So he must have proposed to w' earlier and been rejected/dumped.
This must have happened before w rejected/dumped m. w' was a possible partner for m.(Because they are partnered in M')
So we have a rejection of w' rejecting m' before w rejected m. So w rejecting m was not the *first* rejection of some man by a women in his $P(m)$. $\implies$ Contradiction QED □

Theorem 1.5

The matching output by GS is simultaneously worst for all women. (Each woman is matched with her worst **possible** partner).

Proof. In the discussion section, not too hard once you've probed it's best for all men. □

Issue:

- If you implement GS for your job, which venison do you implement? Men propose? Women propose?

  $\implies$ What does it mean to be fair? There are many natural optimization objectives for stabler matching. Some can be solved efficiently, others are hard. (Papers linked under the courser webpage.)

More fundamentally, as computer scientist in a modern world, our programs/algorithms directly affect the lives of many people. Simple decisions about which algorithm we implement, how we implement it , where we get data from,. ... , may have very profound impact, making some people better off and others worse off.

$\implies$ We need to realize this is happening, think about the impacts, and talk to people who are experts on ethics/-fairness/...
Examples:

- Engagement on social media: an algorithm that maximizes engagement might easily find (by accident) that you should create addictions, and show the most divisive content. This was not by intent, but just by "coincidence". How and on what grounds would you change the goal of the algorithm, or is implementation?
- Whenever ML techniques are used, the outcome is only as good as the training data (and the algorithms). If the training data are biased, or are missing certain groups, the algorithms might perform much worse for those groups, or possibly discriminate against them.
- self-driving cars: decision on trading off driver safety vs. outsider safety

Also, this relates again to the notion of *stability* and *equilibrium*.

2 Greedy Algorithm

Definition 2.1: Greedy Algorithm

Intuitively, make a sequence of irrevocable decisions based only on local/simplified information.
Usually don't work; usually, the first think you want to try.
But when they do, it's usually for an interesting reason.

Problem 2: Interval Selection

Given n intervals starting point $s(i)$, finish point $f(i)$. You want to select as many of them as possible, but without overlaps.
Motivation: event with n presentations, which overlap in their times. You want to attend as many of them as possible.

Solution

Natural Greedy Approaches:

(1) Always pock the shortest remaining interval.

Not optimal, but can be proved to be always within a factor of $\frac{1}{2}$ of optimal.

$\implies$ $\frac{1}{2}$ approximation algorithm

(2) Pick one that starts first.

Can be far from optimal.

(3) Minimize the number of overlaps.

Also not optimal.

(4) Pick one that finishes first.

This actually works.

Greedy Algorithm:

- Sort the intervals by on-decreasing finish time $f(i)$, so now $f(1) \leqslant f(2) \leqslant \ldots \leqslant f(n)$.
- Start with $R = \{1, \ldots, n\}$ (remaining) and $A = \{\}$ (accepted intervals).
- While R not empty:
  - add to A interval in R with smallest $f(i)$
  - remove from R all intervals j which intersects i
- A

First observation: This always fins a legal solution (no overlaps), because it always explicitly removes all intervals that overlap with one we pocked.