Intro to Gale-Shapley Matching

Leonard Tang & Jeremy Hsu

Fall 2021

Contents

1 Classic Problem Setting: Stable Matching

The stable matching problem, in its most basic form, takes as input equal numbers of two types of participants (n men and n women, or n medical students and n internships, for example), and an *ordering* for each participant giving their preference for whom to be matched to among the participants of the other type.

In Datamatch's case, ordering is implied from the output of our scoring function.

A stable matching always exists, and the algorithmic problem solved by the Gale–Shapley algorithm is how to find one.

A matching (i.e. set of matches) is *not* stable if:

1. There is an element A of the first matched set which prefers some given element B of the second matched set over the element to which A is already matched, and
2. The same element B also prefers A over the element to which B is already matched

In other words, a matching is unstable when there is an *unmatched* pair (A,B) where both A and B can improve their outcome by *joint action* (i.e. eloping).

A *stable matching* is defined to be a set of monogamous matches such that there are no unstable pairs.

Example 1.1. Let's say we have three men, Xavier, Ye, and Zion; and three women, Amy, Betty, and Claire. Our matches are (Xavier, Claire), (Ye, Betty), and (Zion, Amy). Suppose that Xavier's original preference list was [Amy, Betty, Claire], in that order. Also suppose that Claire's original preference list was [Xavier, Ye, Zion]. Given this information, is the proposed matching stable?

2 The Existence of Stable Matches

In the traditional problem definition (n men and n women), a stable matching is guaranteed.

Alter the problem statement ever-so-slightly, and this is no longer true.

Indeed, consider the Roommates Problem. Here we have one unified set of $2n$ people who must be paired up to be roommates instead of two groups of n. That is, each person can be paired with $2n-1$ potential "partners".

Example 2.1. Consider 4 roommates: A, B, C, D. Their preferences are A: (B, C, D); B: (C, A, D); C: (A, B, D); and D has no preference. There exists *no* stable matching in this scenario. Let's work out why.

Suppose we had the matching (A,B) and (C,D). Clearly C prefers B over D, and B also prefers C over A $\rightarrow$ so this is an unstable matching. Then we might group B and C together to get (A,C) and (B,D). But here, A prefers B over C and B prefers A over D, so this is unstable as well. Work out the third (and last combination, since we have $\frac{4!}{2!2!2!}$ choices) for yourself and convince yourself that it's unstable as well.

3 Gale-Shapley Algorithm (Propose and Reject)

Algorithm 1 Propose-and-Reject Algorithm

Require: Initialize each person to be free (unmatched)
  while Some man is free and hasn't proposed to every woman **do**
    Choose such a man m
    $w \leftarrow$ 1st woman on m's list to whom m has not yet proposed
    if w is free **then**
      Match m and w
    else if w prefers m to her current match m' **then**
      Match m and w, and free up m'
    else
      w rejects m
    end if
  end while

That's literally it!

4 Proof(s) of Correctness

Observation 1: Men propose to women in decreasing order of preference.

Observation 2: Once a woman is matched, she never becomes unmatched. In fact, she only becomes matched with better partners ("trades up").

4.1 Termination (and Runtime)

How do we know that this algorithm will always terminate?

Proposition 4.1. *Algorithm terminates after at most n^2 iterations of while loop.*

Proof. Each pass through the **while** loop, a man proposes to a *new* woman (see the second line in the **while** loop). There are only n^2 possible proposals (n men can each propose to n women on their ranking list). □

4.2 No Single Left Behind

How do we guarantee that there are no stragglers in after perform our matching? That is, how do we know all women and men are matched?

Proposition 4.2. *All women and men are matched.*

Proof. Suppose, for the sake of contradiction, that male X is not matched upon termination of the algorithm.

Then some woman, say Y, is not matched upon termination either (since we have n equal men and women in *monogamous* relationships).

By Observation 2, woman Y was never matched with in the first place (as in no circumstance could she become unmatched).

But X must have proposed to everyone, since he is a free man (the algorithm terminates since the second condition of the while statement, that he hasn't proposed to every woman, becomes false). So he must have proposed to woman Y, hence a contradiction. □

4.3 Stability

How do we know that there are no unstable pairs?

Proposition 4.3. *There are no unstable pairs after Gale-Shapley terminates.*

Proof. Suppose the man-woman pair $M - W$ is an unstable (unmatched) pair: each prefers each other to their current partners in the given Gale-Shapley matching S^*.

- Case 1: M never proposed to W. Since men propose in decreasing order of preference, this means that M's current partner in the matching S^* is more preferable. Hence $M - W$ is actually stable, since M does *not* prefer W over his current partner.
- Case 2: M did in fact propose to W. Since they are not a match, W must have rejected M at some point (either immediately during the time of proposal, or swaps him for a better man). Since W could have only "traded up", or W did not think M was better than her partner at the time of proposal, W must prefer her current partner over M.

In both cases, the unstable pair M-W actually turns out to be stable, hence a contradiction.

□