Stable Matching

CS31005: Algorithms-II
Autumn 2020
IIT Kharagpur

Stable Matching

- A type of perfect matching in a complete bipartite graph
- Posed usually as the “Stable Marriage Problem”
- Well known for application to the problem of assigning doctors to hospitals for residencies (internships)
  - Doctors list the hospitals in order of preference
  - Hospitals list the students in order of preference
  - Find stable matching between the doctors and the hospitals
    - Unstable pair: Suppose doctor D1 is assigned to H1 and D2 to H2, but D1has higher preference for H2 than H1 and H2 has higher preference for D1 over D2
    - Can make D1 move to H2, who will welcome the student

The Stable Marriage Problem

- There are n men and n women, all unmarried
- Each has a preference list giving a relative preference of each person of the opposite sex
- Find a matching between the men and the women such that
  - Each man is matched to exactly one woman and each woman is matched to exactly one man (perfect matching)
  - There is no unstable pair (an unmatched pair of a man and a woman who both prefer each other over whoever they are assigned to in the matching)
- Stable matching – perfect matching with no unstable pair

Example

- 3 men m1, m2, m3 and 3 women w1, w2, w3
- Preference List (highest to lowest):

m1: w2, w1, w3	w1: m2, m1, m3
m2: w1, w3, w2	w2: m1, m3, m2
m3: w1, w3, w2	w3: m2, m1, m3

- Stable matching: (m1, w2), (m2, w1), (m3, w3)
- An unstable matching: (m1, w1), (m2, w2), (m3, w3)
  - The unmatched pair (m1, w2) is unstable, as both m1 and w2 prefer each other over their current matchings

Gale-Shapely Algorithm

- Proposed by Gale and Shapely in 1962
  - Lloyd Shapely was awarded the Nobel Prize in Economics in 2012 partly for this
- Showed that a stable matching always exists and gave an algorithm to find it
- We will denote the status of each man and woman as free or matched
- A matched pair of a man m and woman w will be denoted by (m, w)
- Let M denote the set of matched pairs

The Algorithm

```
Set initial status of all men and women as free
while (some man m is free)
    w = first woman on m's list       /* m proposes to w */
    if  (w is free)
        add (m, w) to M               /* w accepts m’s proposal */
        set status of m and w to matched
    else if  ( (m’, w) is in M and w has higher preference for m to m')
            /* m is better match for w, so w breaks engagement with m’ and
                gets engaged with m. m’ becomes free again */
            add (m, w) to M and remove (m‘, w) from M
            set status of m to  matched and status of m' to  free
        else     /* w rejects m, nothing else to do */
    remove w from m’s list   /* each man proposes to a woman only once */
```

Example

Preferences:

m1: w1, w3, w2	w1: m2, m1, m3
m2: w1, w2, w3	w2: m3, m1, m2
m3: w1, w3, w2	w3: m1, m2, m3

We will use **green** to indicate status free and **blue** to indicate status matched

Initial status: **m1, m2, m3, w1, w2, w3**

Action	M	Description	Status
m1 → w1	(m1, w1)	w1 accepts m1	m1, m2, m3, w1, w2, w3
m2 → w1	(m2, w1)	w1 breaks from m1 and accepts m2	m1, m2, m3, w1, w2, w3
m3 → w1	(m2, w1)	w1 rejects m3	m1, m2, m3, w1, w2, w3
m3 → w3	(m2, w1), (m3, w3)	w3 accepts m3	m1, m2, m3, w1, w2, w3
m1 → w3	(m2, w1), (m1, w3)	w3 breaks from m3 and accepts m1	m1, m2, m3, w1, w2, w3
m3 → w2	(m2, w1), (m1, w3), (m3, w2)	w2 accepts m3	m1, m2, m3, w1, w2, w3

Some Observations

- Men propose to their highest preference woman first
- Once a woman is matched, she never becomes free, can only change from a lower preference partner to a higher preference partner
- All men and women are eventually matched
  - Because if not, suppose man m is not matched. Then there must be a woman w also not matched. But then w has never received a proposal (or it would have matched with the first one and then never get unmatched). This is a contradiction as m has proposed to everyone since he is unmatched.

Proof of Correctness

Theorem: When the algorithm terminates, the set M contains a stable matching

Proof: Suppose that there is an unstable pair (m,w). Let their current matchings be (m, w') and (m', w). Then there are two possibilities:

- m has never proposed to w: But then, m must have higher preference for w' than w, so (m,w) is not an unstable pair.
- m has proposed to w: But then, w must have rejected m (either at the time of the proposal, or later when she got a proposal from a higher preference man). So w has higher preference for m' than m. So (m, w) is not an unstable pair.

- Will it terminate?
  - Yes, because a man proposes to a woman at most once
  - Time complexity = $O(n^2)$
- The is fine, but Gale-Shapely algorithm requires
  - No. of men = No. of women
  - All men to give preference to all women and vice-versa.
- What if not satisfied (ex. In the resident matching problem, a doctor may not give preference for all hospitals)
  - Not a big problem
    - Add dummy nodes to make both sides same as before
    - If m has not given preference for w (or vice-versa), add a dummy preference of m for w lower than any other preference m has actually given
    - In the final matching, throw away any edge that uses either a dummy node or a dummy preference

- An interesting observation
  - The stable matching found by the Gayle-Shapely algorithm when men propose first is man-optimal
    - Each man gets his highest preference partner subject to the stability constraint
    - More precisely: Consider any man m. Then a woman w is a valid partner of m if there exists at least one stable matching in which m is matched with w. Let $w_{highest}$ be the highest ranked valid partner of m (highest in m's preference list for women). Then in the stable matching produced by the Gayle-Shapely algorithm when men propose first, m is paired with $w_{highest}$

- Note that "man" and "woman" are just placeholders here, you get a woman-optimal matching if women propose to men first
  - Important it practice, as you can make the algorithm favor one set over the other by choosing where to start from
    - For example, if the hospitals "propose" first, they are benefitted more over the students
    - Get their highest choice possible subject to stability constraint

A Related Problem

- Stable Roommate problem
  - Set of 2n people, each of whom rank everyone else in order of preference. Find a perfect matching (a disjoint set of n pairs) such that there is no unstable pair.
  - An example is assigning roommates in allocating hostel rooms (double bed rooms)
  - We will do not do this, not in syllabus