CSE 421: Introduction to Algorithms

Stable Matching

Shayan Oveis Gharan

Propose-And-Reject Algorithm [Gale-Shapley'62]

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman) {
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged, and m' to be free
    else
        w rejects m
}
```

Summary

Stable matching problem: Given n men and n women, and their preferences, find a stable matching if one exists.

- Gale-Shapley algorithm: Guarantees to find a stable matching for any problem instance.

- Q: How to implement GS algorithm efficiently?

- Q: If there are multiple stable matchings, which one does GS find?

- Q: How many stable matchings are there?

Implementation of GS Algorithm

Problem size

- $N=2n^2$ words
  - $2n$ people each with a preference list of length n
- $2n^2 \log n$ bits
  - specifying an ordering for each preference list takes $n \log n$ bits

Brute force algorithm

- Try all $n!$ possible matchings
- Do any of them work?

Gale-Shapley Algorithm

- n^2 iterations, each costing constant time as follows:

Efficient Implementation

We describe $O(n^2)$ time implementation.

Representing men and women:

Assume men are named 1, …, n.
Assume women are named n+1, …, 2n.

Engagements.

Maintain a list of free men, e.g., in a queue.
Maintain two arrays **wife[m]**, and **husband[w]**.

- set entry to **0** if unmatched
- if **m** matched to **w** then **wife[m]=w** and **husband[w]=m**

Men proposing:

For each man, maintain a list of women, ordered by preference.
Maintain an array **count[m]** that counts the number of proposals made by man **m**.

Efficient Implementation

Women rejecting/accepting.

Does woman **w** prefer man **m** to man **m'**?

For each woman, create inverse of preference list of men.

Constant time access for each query after **O(n)** preprocessing per woman. **O(n²)** total reprocessing cost.

w_i	1st	2nd	3rd	4th	5th	6th	7th	8th
Pref	8	3	7	1	4	5	6	2

w_i	1	2	3	4	5	6	7	8
Inverse	4th	8th	2nd	5th	6th	7th	3rd	1st

```
for i = 1 to n
   for j = 1 to n
      inverse[i][pref[i][j]] = j
```

w_i prefers man **3** to **6**

since `inverse[i][3]`=2 < 7=`inverse[i][6]`

Summary

- Stable matching problem: Given **n** men and **n** women, and their preferences, find a stable matching if one exists.

- Gale-Shapley algorithm guarantees to find a stable matching for any problem instance.

- GS algorithm finds a stable matching in $\mathbf{O(n^2)}$ time. ✓
- Q: If there are multiple stable matchings, which one does GS find?
- Q: How many stable matchings are there?

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

An instance with two stable matchings:

- (m_1, w_1), (m_2, w_2).
- (m_1, w_2), (m_2, w_1).

	1st	2nd
m_1	w_1	w_2
m_2	w_2	w_1

	1st	2nd
w_1	m_2	m_1
w_2	m_1	m_2

Man Optimal Assignments

Definition: Man m is a valid partner of woman w if there exists some stable matching in which they are matched.

Man-optimal matching: Each man receives the best valid partner (according to his preferences).

- Not that each man receives his most favorite woman.

Claim: All executions of GS yield a man-optimal matching, which is a stable matching!

- So, output of GS is unique!!
- No reason a priori to believe that man-optimal matching is perfect, let alone stable.

Man Optimality

S

(m, w)
(m', w')
$\ldots$

Claim: GS matching **S*** is man-optimal.

Proof: (by contradiction)

Suppose some man is paired with someone other than his best partner. Men propose in decreasing order of preference $\Rightarrow$ some man is rejected by a valid partner.

Let m be the man who is the first such rejection, and let w be the women who is first valid partner that rejects him.

Let **S** be a stable matching where m and w are matched.

In building **S***, when m is rejected, w forms (or reaffirms) engagement with a man, say m' whom she prefers to m.

Let w' be m' partner in **S**.

In building **S***, m' is not rejected by any valid partner at the point when m is rejected by w. Thus, m' prefers w to w'.

since this is the first rejection by a valid partner

But w prefers m' to m.

Thus (m', w) is unstable in **S**. ■

Man Optimality Summary

Man-optimality: In version of GS where men propose, each man receives the best valid partner.

> w is a valid partner of m if there exist some stable matching where m and w are paired

Q: Does man-optimality come at the expense of the women?

Woman Pessimality

Woman-pessimal assignment: Each woman receives the worst valid partner.

Claim. GS finds woman-pessimal stable matching **S***.

Proof.

Suppose (m, w) matched in **S***, but m is not worst valid partner for w.

There exists stable matching **S** in which w is paired with a man, say m', whom she likes less than m.

- Let w' be m' partner in **S**.
- m prefers w to w'. ← man-optimality of **S***
- Thus, (m, w) is an unstable in **S**.

■

Summary

- Stable matching problem: Given **n** men and **n** women, and their preferences, find a stable matching if one exists.
- Gale-Shapley algorithm guarantees to find a stable matching for any problem instance.
- GS algorithm finds a stable matching in **O**(n^2) time.
- GS algorithm finds man-optimal woman pessimal matching
- Q: How many stable matching are there?

How many stable Matchings?

We already show every instance has at least 1 stable matchings.

There are instances with about b^n stable matchings for $b > 2$

[Karlin-O-Weber'17]: Every instance has at most c^n stable matchings for some $c > 2$

[Research-Question]:

Is there an "efficient" algorithm that chooses a uniformly random stable matching of a given instance.