CSE 421: Introduction to Algorithms

Stable Matching

Shayan Oveis Gharan

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

```
Initialize each side to be free.
while (some company is free and hasn't proposed to every applicant) {
    Choose such a c
    a = 1st applicant on c's list to whom c has not yet proposed
    if (a is free)
        assign c and a
    else if (a prefers c to her current c’)
        assign c and a, and c' to be free
    else
        a rejects c
}
```

First step: Properties of Algorithm

Observation 1: Companies propose to Applicants in decreasing order of preference.

Observation 2: Each company proposes to each applicant at most once

Observation 3: Once an applicant is matched, she never becomes unmatched; she only "trades up."

2) Correctness: Output is Perfect matching

Claim. All Companies and Applicants get matched.

Proof. (by contradiction) First, notice each company/applicant is matched to at most one other agent.

Suppose, for sake of contradiction, that c is not matched upon termination of algorithm.

Then some applicant, say a, is not matched upon termination.

By Observation 3 (only trading up, never becoming unmatched), a was never proposed to.

But, c proposes to everyone, since it ends up unmatched.

■

2) Correctness: Stability

Claim. No unstable pairs.

Proof. (by contradiction)

Suppose c, a is an unstable pair: each prefers each other to the partner in Gale-Shapley matching **S***.

Obs1: companies propose in decreasing order of preference

Case 1: c never proposed to a.

- $\Rightarrow$ c prefers its **S*** partner to a.
- $\Rightarrow$ c, a is stable.

Case 2: c proposed to a.

- $\Rightarrow$ a rejected c (right away or later)
- $\Rightarrow$ a prefers her **S*** partner to c.
- $\Rightarrow$ c, a is stable.

Obs3: applicants only trade up

In either case c, a is stable, a contradiction. ■

Summary

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

- Gale-Shapley algorithm: Guarantees to find a stable matching for any problem instance.
- Q: If there are multiple stable matchings, which one does GS find?
- Q: How to implement GS algorithm efficiently?
- 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:

- $(c_1, a_1), (c_2, a_2).$
- $(c_1, a_2), (c_2, a_1).$

	1st	2nd
c_1	a_1	a_2
c_2	a_2	a_1

	1st	2nd
a_1	c_2	c_1
a_2	c_1	c_2

Company Optimal Assignments

Definition: Company c is a valid partner of applicant a if there exists some stable matching in which they are matched.

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

- Not that each company receives its most favorite applicant.

Example

Here

Valid-partner(c_1) $= \{a_1, a_2\}$

Valid-partner(c_2) $= \{a_1, a_2\}$

Valid-partner(c_3) $= \{a_3\}$.

Company-optimal matching $\{c_1, a_1\}, \{c_2, a_2\}, \{c_3, a_3\}$

	favorite ↓		least favorite ↓
	1st	2nd	3rd
c_1	a_1	a_2	a_3
c_2	a_2	a_1	a_3
c_3	a_1	a_2	a_3

	favorite ↓		least favorite ↓
	1st	2nd	3rd
a_1	c_2	c_1	c_3
a_2	c_1	c_2	c_3
a_3	c_1	c_2	c_3

Company Optimal Assignments

Definition: Company c is a valid partner of applicant a if there exists some stable matching in which they are matched.

Company-optimal matching: Each company receives the best valid partner (according to its preferences).

- Not that each company receives its most favorite applicant.

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

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

Applicant Pessimality

Applicant-pessimal assignment: Each applicant receives the worst valid partner.

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

Proof.

Suppose (c, a) matched in **S***, but c is not the worst valid partner for a.

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

Let a' be c partner in **S**.

c prefers a to a'. ← company-optimality of **S***

Thus, (c, a) is an unstable in **S**.

■

Company Optimality

S
(c, a)
(c', a')
$\ldots$

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

Proof: (by contradiction)

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

Let c be the first such rejection, and let a be its best valid partner.

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

In building **S***, when c is rejected, a is assigned to a company, say c' whom she prefers to c.

Let a' be c' partner in **S**.

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

But a prefers c' to c.

Thus (c', a) is unstable in **S**.

since this is the first rejection by a valid partner

■

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 man-optimal woman pessimal matching ✓
- GS algorithm finds a stable matching in **O**(**n**^{2}) time.

- Q: How many stable matching are there?

Induction: Intro 1

Prove that for all $n \geq 1$,

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Def $P(n) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$

Base Case: $P(1)$ holds: $1 = 1(1+1)/2$

IH: $P(n-1)$ holds for some $n \geq 2$

IS: Goal to prove $P(n)$.

$$\begin{aligned} 1 + \cdots + n &= (1 + \cdots + n - 1) + n \\ &= \left(\frac{(n-1)n}{2}\right) + n \qquad \text{By IH} \\ &= \frac{n(n+1)}{2} \end{aligned}$$

Induction: Intro 2

Prove that if n+1 balls are placed into n bins then one bin has at least two balls.

Def: P(n): For all placements of n+1 balls into n bins there exists a bin with at least two balls.

Base Case: P(1) holds. Two balls into one bin

IH: P(n-1) holds for some $n \geq 2$

IS: Goal is to prove P(n). Suppose n+1 balls are placed into n bins arbitrarily. Need to show a bin has ≥ 2 balls. Look at bin 1.

Case 1: Bin 1 has at least two balls. Then we are done.

Case 2: Bin 1 has 1 ball. Then. we have placed n balls into bins 2,..,n. So, by IH one bin has at least two balls.

Case 3: Bin 1 has 0 balls. Remove an arbitrary ball. Then, we have n balls in bins 2,..,n. So, by IH a bin has ≥ 2 balls