南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

03 A First Problem: Stable Matching

CS216 Algorithm Design and Analysis (H)

Instructor: Shan Chen

chens3@sustech.edu.cn

Stable Matching

- **Motivation:** [Gale and Shapley 1962]
  - Can one design a college admissions process, or job recruiting process, that is self-enforcing (stable)?
- **Example:** A “bare-bones” version of job recruiting
  - *n* applicants and *n* companies
  - Each applicant ranks companies and each company ranks applicants
  - Each company may hire one or multiple applicants.
- Let’s first look at a **simpler setting**: **one-to-one matching** (e.g., **marriage**)
  - *n* men: $A = \{m_1, m_2, \ldots, m_n\}$ and *n* women: $B = \{w_1, w_2, \ldots, w_n\}$
  - Each man can be married to at most one woman and vice versa.
  - A matching M is a subset of the Cartesian product $A \times B$.

Some Definitions

- **Perfect matching:** everyone is matched monogamously (一夫一妻)
  - Each man gets exactly one woman.
  - Each woman gets exactly one man.
- **Stability:** no pair of participants has incentive to undermine the current matching by joint action

在稳定匹配中，没有任何一对男人和女人能通过交换彼此的配对来使自己变得更好；交换后，新形成的这一对男女都变得更好

  - In a matching *M*, an unmatched pair *m* - *w* is unstable if man *m* and woman *w* prefer each other to their current partners.
  - An unstable pair *m* - *w* could each improve by joint action (e.g., eloping).
- **Stable matching:** perfect matching with no unstable pairs

The Stable Marriage/Matching Problem

- **The stable marriage/matching problem.** Given the preference lists of n men and n women, find a stable matching if one exists.

- **Example 1** $(n = 2)$: $[m_1: w_1 > w_2;\ m_2: w_1 > w_2;\ w_1: m_1 > m_2;\ w_2: m_1 > m_2]$
  - ➢ The stable matching $\{m_1 - w_1,\ m_2 - w_2\}$ is unique.
    - ✓The other perfect matching $\{m_1 - w_2,\ m_2 - w_1\}$ has an unstable pair $m_1 - w_1$.
- **Example 2** $(n = 2)$: $[m_1: w_1 > w_2;\ m_2: w_2 > w_1;\ w_1: m_2 > m_1;\ w_2: m_1 > m_2]$
  - ➢ Both perfect matchings are stable:
    - ✓$\{m_1 - w_1,\ m_2 - w_2\}$: both men are happy
    - ✓$\{m_1 - w_2,\ m_2 - w_1\}$: both women are happy

Questions

- **Q.** Do stable matchings always exist in general?
- **A.** No. See the counterexample (not a marriage problem) below.

	1st	2nd	3rd
A	B	C	D
B	C	A	D
C	A	B	D
D	A	B	C

no perfect matching is stable:

{A – B, C – D} => B – C unstable
{A – C, B – D} => A – B unstable
{A – D, B – C} => A – C unstable

- **Q.** Does there exist stable matchings for the marriage problem? 稳定婚姻问题总是存在稳定匹配
- **Q.** How can we find such a stable matching?

The Gale-Shapley Algorithm

- **The Gale-Shapley algorithm.** [Gale-Shapley 1962] An intuitive method that guarantees to find a stable matching.
  - also known as the propose-and-reject or delayed-acceptance algorithm
  - Idea: men propose to preferred women (but may get rejected) until all matched

```
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
}
```

Proof of Correctness: Termination

男人按照递减的偏好顺序向女人提出求婚

- **Observation 1.** Men propose to women in decreasing preference order.
- **Observation 2.** Women only "trade up": once a woman is matched, she never becomes unmatched. 女人只会交换的更好

- **Claim.** Algorithm terminates after at most n^2 iterations of the while loop.
- **Pf.** Each time through the while loop a man proposes to a new woman. There are only n^2 possible proposals. ▪

- **Q.** Can you think of a scenario that requires $\Theta(n^2)$ steps for GS?
- **A.** E.g., all men ranks women in the same order, and all women ranks men in the opposite order.

Proof of Correctness: Perfection

- **Claim.** All men and women are uniquely matched.
- **Pf.** (by contradiction)
  - Suppose that Zeus is not matched upon termination of algorithm.
  - Then some woman, say Amy, is not matched upon termination.
  - By Observation 2, Amy was never proposed to.
  - But Zeus proposed to everyone, since he ends up unmatched. Contradiction! ▪

Proof of Correctness: Stability

- **Claim.** No unstable pairs.
- **Pf.** (by contradiction)
  - Suppose *Z* - *A* is an unstable pair (see the bottom-right figure), i.e., each prefers each other to their current partner in the Gale-Shapley matching *S**.
  - Case 1: *Z* never proposed to *A*.
    - ✓*Z* prefers *B* to *A*. ⟵ men propose in decreasing order of preference
    - ✓So, *Z* - *A* is stable.
  - Case 2: *Z* proposed to *A* but got rejected (right away or later)
    - ✓*A* prefers *Y* to *Z*. ⟵ women only trade up
    - ✓So, *Z* - *A* is stable.
  - In either case, *Z* - *A* is stable. Contradiction! ■

*S**
Zeus - Bertha
Yancey - **Amy**
. . .

Quick Summary and Questions

- **The stable marriage problem.** Given n men and n women, and their preferences, find a stable matching if one exists.
- **The Gale-Shapley algorithm.** Guarantees to find a stable matching for any problem instance.

- **Q.** How to implement the GS algorithm efficiently?
- **Q.** If there are multiple stable matchings, which one does GS find?

Efficient $O(n^2)$ Implementation

- **Representing men and women.**
  - Assume men and women are each named *1, …, n*.
- **Recording the matching.**
  - Maintain a list of free men, e.g., in a queue or stack.
  - Maintain two arrays wife[m], and husband[w].
    - ✓Set entries to 0 if unmatched.
    - ✓If *m* matches *w*, then wife[m] = w and husband[w] = m.
- **Men proposing.**
  - For each man, maintain a list of women, ordered by preference.
  - Keep an array count[m] that counts the number of proposals made by man *m*.

Efficient $O(n^2)$ Implementation

- **Women accepting/rejecting.**
  - How can we *efficiently* check if woman w prefers man m to man m' ?
  - For each woman, create an inverse mapping from men to preference orders.
    ✓ $O(1)$ access for each query after $O(n)$ preprocessing

Amy	1st	2nd	3rd	4th	5th	6th	7th	8th
pref	8	3	7	1	4	5	6	2

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

Amy	1	2	3	4	5	6	7	8
inverse	4th	8th	2nd	5th	6th	7th	3rd	1st

E.g., Amy prefers man 3 to 6 since
inverse[3] = 2 < 7 = inverse[6]

- **Memory.** It is not hard to see that GS consumes $O(n^2)$ memory.
  - Input and output also take memory!

Understanding the Solution

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

- **Def.** Man *m* is a valid partner of woman *w* if there exists some stable matching in which they are matched.
- **Def.** The man-optimal matching: every man receives best valid partner.
- **Claim.** All GS executions yield the man-optimal matching, which is also a stable matching! Very surprising, isn't it?
  - ➢ The man-optimal matching is simultaneously best for all men.
  - ➢ No reason to believe that the man-optimal matching exists, let alone stable!

Man Optimality

- **Claim.** The GS matching S^* is man-optimal.
- **Pf.** (by contradiction)
  - Suppose S^* is not man-optimal, i.e., some man is not paired with his best valid partner. Since men proposed in decreasing order of preference, some man is rejected by his valid partner.
  - Let Y be the first such man and let A be the first valid partner of Y that rejects Y.
  - When Y is rejected, A (re)affirms matching with a man, say Z, whom A prefers to Y. We know Z was not rejected by any valid partner at this point, so Z prefers A to any other valid partners. since A is a valid partner of Y
  - There exists a stable matching S where Y and A are matched. Let B be Z's valid partner in S. From above, Z prefers A to B.
  - Also, A prefers Z to Y, so Z - A is unstable in S. Contradiction! ∎

S
Yancey-Amy
Zeus-Bertha
. . .

Woman Pessimality

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

- **Def.** Woman-pessimal assignment: every woman gets worst valid partner.
- **Claim.** The GS matching S^* is woman-pessimal.
- **Pf.** (by contradiction)
  - Suppose Z - A is matched in S^*, but Z is not the worst valid partner for A.
  - There exists a stable matching S in which A is paired with a man, say Y, whom A likes less than Z. Let B be Z's valid partner in S.
  - From man-optimality of S^*, we have Z prefers A to B.
  - Recall A prefers Z to Y, so Z - A is unstable in S. Contradiction! ∎

S
Yancey-Amy
Zeus-Bertha
. . .

Gale-Shapley Algorithm: Summary

- **The Gale-Shapley algorithm.** Finds a stable matching in $O(n^2)$ time.
- **Man optimality.** In the version of the GS algorithm where men propose, each man receives best valid partner.
- **Woman pessimality.** In the version of the GS algorithm where men propose, each woman receives worst valid partner.

- **Q.** If you want a best mate, would you propose or wait to be proposed?

Extension: Matching Students to Hospitals

- **Extension:** hospitals hire medical students
  - Variant 1. Participants declare others as unacceptable.
  - Variant 2. Unequal number of positions and students.
  - Variant 3. Limited polygamy.

some student is unwilling to work in some hospitals, or the other way.

some hospital could hire multiple students, e.g., ≤ 3

- In Assignment 1, you are asked to prove that GS can be adapted to find stable matchings in the above generalized setting.
  - To prove it, you need to first define stable matching in this setting.

Men/Women ≠ Hospitals/Students

- **Men/Women marriage:** one-to-one matching
- **Hospitals/Students recruitment:** one-to-many matching

- For around 20 years, most people thought the above problems had very similar properties. However, this is wrong.
  - [Roth 1982] Any algorithm for men/women marriage (e.g., man-proposing GS) that yields a man-optimal stable matching implies that truth telling is the dominant strategy for men.
  - [Roth 1985] No stable matching algorithm for hospitals/students recruitment exists such that truth-telling is the dominant strategy for hospitals.

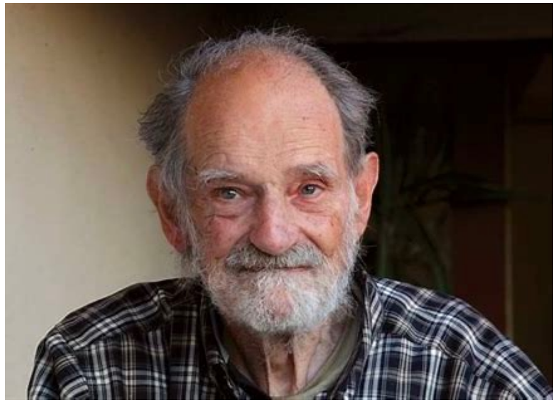
Real-World Application: NRMP

- **National Resident Matching Program (NRMP):**
  - The algorithm is an extension to GS but was in practical use before GS!
  - Original use in 1950s, just after WWII. ← predates computer usage
  - Initial version does not handle couples and other special cases.
  - The full algorithm was adopted and used since late 1990s.

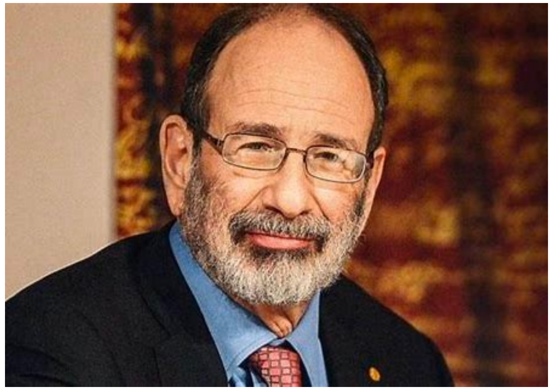
- **Rural hospital dilemma.** Certain hospitals (mainly in rural areas) are unpopular and declared unacceptable by many students.
  - How can we find stable matchings that benefit "rural hospitals"?
- **Rural hospital theorem.** [Roth 1986] Rural hospitals get exactly the same students in every stable matching!

2012 Nobel Prize in Economics

- **Lloyd Shapley.** Stable matching theory and Gale-Shapley algorithm.



- **Alvin Roth.** Applied Gale-Shapley to matching med-school students with hospitals, students with schools, and organ donors with patients.



More on Stable Matching

- **The stable roommate problem:**
  - Matching is defined on general graphs (may be non-bipartite)
  - Stable matchings may not exist!

- **Q.** Can we find a polynomial-time algorithm that does the following?
  - either finds a stable matching
  - or reports non-existence
- **A.** Irving's algorithm [Irving 1985]
  - builds on GS ideas and work by [McVitie and Wilson 1971].

Lessons Learned

- **Powerful ideas of algorithm design and analysis:**
  - Isolate underlying structure of the problem.
  - Design useful and efficient algorithms.
  - Prove correctness and bound time and memory.

- **Caveat.** Potentially deep social ramifications. [legal disclaimer]

Announcements

- **Assignment 1 has been released and the deadline is Mar 11.**
- **Lab 2 will be released today and the deadline is Mar 12 at 2pm.**