CSE 421: Introduction to Algorithms

Stable Matching

Paul Beame

Matching Residents to Hospitals

- Goal: Given a set of preferences among hospitals and medical school residents (graduating medical students), design a self-reinforcing admissions process.

- Unstable pair: applicant **x** and hospital **y** are unstable if:
  - **x** prefers **y** to their assigned hospital.
  - **y** prefers **x** to one of its admitted residents.

- Stable assignment. Assignment with no unstable pairs.
  - Natural and desirable condition.
  - Individual self-interest will prevent any applicant/hospital side deal from being made.

Simpler:Stable Matching Problem

- Goal. Given two groups of **n** people each, find a "suitable" matching.
  - Participants rate members from opposite group.
  - Each person lists members from the other group in order of preference from best to worst.

favorite ↓ … least favorite ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

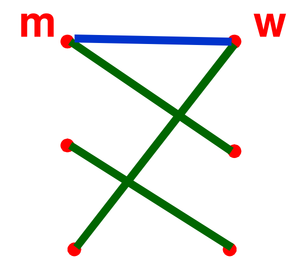
Group 0 Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Group 1 Preference Profile

Stable Matching Problem



- Perfect matching: everyone is matched to precisely one person from the other group

- Stability: self-reinforcing, i.e. no incentive for some pair of participants to undermine assignment by joint action.
  - In matching **M**, an unmatched pair **m**-**w** from different groups is unstable if **m** and **w** prefer each other to current partners.
  - Unstable pair **m**-**w** could each improve by ignoring the assignment.

- Stable matching: perfect matching with no unstable pairs.

- Stable matching problem. Given the preference lists of **n** people from each of two groups, find a stable matching between the two groups if one exists.

Stable Matching Problem

- Q. Is assignment X-C, Y-B, Z-A stable?

favorite ↓ … least favorite ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Group 0 Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Group 1 Preference Profile

Stable Matching Problem

- Q. Is assignment X-C, Y-B, Z-A stable?
- A. No. B and X prefer each other.

favorite ↓ … least favorite ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Group 0 Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Group 1 Preference Profile

Stable Matching Problem

- Q. Is assignment X-A, Y-B, Z-C stable?
- A. Yes.

	favorite ↓ 1st	2nd	least favorite ↓ 3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Group 0 Preference Profile

	favorite ↓ 1st	2nd	least favorite ↓ 3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Group 1 Preference Profile

Stable Roommate Problem

- Q. Do stable matchings always exist?
- A. Not obvious a priori.

- Stable roommate problem.
  - **2n** people; each person ranks others from **1** to **2n-1**.
  - Assign roommate pairs so that no unstable pairs.

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

A-B, C-D $\Rightarrow$ B-C unstable
A-C, B-D $\Rightarrow$ A-B unstable
A-D, B-C $\Rightarrow$ A-C unstable

- Observation. Stable matchings do not always exist for stable roommate problem.

Propose-And-Reject Algorithm

- Propose-and-reject algorithm. [Gale-Shapley 1962]
  Intuitive method that guarantees to find a stable matching.
- One group is designated *proposers*, the other *receivers*

```
Initialize each person to be free.
while (some proposer is free and hasn't proposed to every
        receiver) {
     Choose such a proposer m
     w = 1st receiver 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 current tentative match m')
          assign m and w to be engaged, and m' to be free
     else
          w rejects m
}
```

Proof of Correctness: Termination

- Observation 1. Proposers propose to receivers in decreasing order of preference.

- Observation 2. Once a receiver is matched, they never become unmatched; they only "trade up."

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

	1st	2nd	3rd	4th	5th
V	A	B	C	D	E
W	B	C	D	A	E
X	C	D	A	B	E
Y	D	A	B	C	E
Z	A	B	C	D	E

Proposers' Preference Profile

	1st	2nd	3rd	4th	5th
A	W	X	Y	Z	V
B	X	Y	Z	V	W
C	Y	Z	V	W	X
D	Z	V	W	X	Y
E	V	W	X	Y	Z

Receivers' Preference Profile

$n(n-1) + 1$ proposals required in the worst case

Proof of Correctness: Perfection

- Claim. Everyone gets matched.
- Proof. (by contradiction)
  - Suppose, for sake of contradiction, that some proposer Z is not matched upon termination of algorithm.
  - Then some receiver, say A, is not matched upon termination.
  - By Observation 2 (only trading up, never becoming unmatched), A was never proposed to.
  - But, Z proposes to everyone, since Z ends up unmatched. Contradiction ▪

Proof of Correctness: Stability

- Claim. No unstable pairs.
- Proof. (by contradiction)
  - Suppose **A**-**Z** is an unstable pair: each prefers each other to partner in Gale-Shapley matching **S***.
  - Case 1: **Z** never proposed to **A**.
    ⇒ **Z** prefers GS partner to **A**.
    ⇒ **A**-**Z** is stable.
  - Case 2: **Z** proposed to **A**.
    ⇒ **A** rejected **Z** (right away or later)
    ⇒ **A** prefers GS partner to **Z**.
    ⇒ **A**-**Z** is stable.
  - In either case **A**-**Z** is stable, a contradiction. ▪

proposers propose in decreasing order of preference

receivers only trade up

S*

A-Y
B-Z
. . .

Summary

- Stable matching problem. Given **n** people in each of two groups, 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?

Implementation for Stable Matching Algorithms

- 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

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

- Representing proposers and receivers.
  - Assume proposers are named **1**, …, **n**.
  - Assume receivers are named **1'**, …, **n'**.

- Engagements.
  - Maintain a list of free proposers, e.g., in a queue.
  - Maintain two arrays **match[m]**, and **match'[w]**.
    - set entry to **0** if unmatched
    - if **m** matched to **w** then **match[m]=w** and **match'[w]=m**

- Proposals.
  - For each proposers, maintain a list of receivers, ordered by preference.
  - Maintain an array **count[m]** that counts the number of proposals made by proposer **m**.

Efficient Implementation

- Receivers rejecting/accepting.
  - Does receiver **w** prefer proposer **m** to proposer **m'**?
  - For each receiver, create inverse of preference list of proposers.
  - Constant time access for each query after $O(n)$ preprocessing per receiver. $O(n^2)$ total reprocessing cost.

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

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

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

A prefers proposer 3 to 6
since `inverse[3]=2` < `7=inverse[6]`

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?

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

- An instance with two stable matchings.
  - A-X, B-Y, C-Z.
  - A-Y, B-X, C-Z.

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?

- Def. Proposer **m** is a valid partner of receiver **w** if there exists some stable matching in which they are matched.

- Proposer-optimal assignment. Each proposer receives **best** valid partner (according to their preferences).

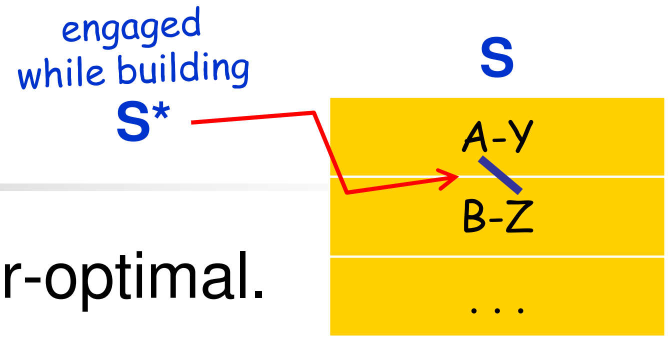
- Claim. All executions of GS yield a **proposer-optimal** assignment, which is a stable matching!
  - No reason a priori to believe that proposer-optimal assignment is perfect, let alone stable.
  - Simultaneously best for each and every proposer.

Proposer Optimality

- Claim. GS matching **S*** is proposer-optimal.
- Proof. (by contradiction)
  - Suppose some proposer is paired with someone other than their best partner. Proposers propose in decreasing order of preference ⇒ some proposer is rejected by a valid partner.
  - Let **Y** be the proposer who is the first such rejection, and let **A** be the receiver who is first valid partner that rejects him.
  - Let **S** be a stable matching where **A** and **Y** are matched.

Must exist since **Y** and **A** are valid partners

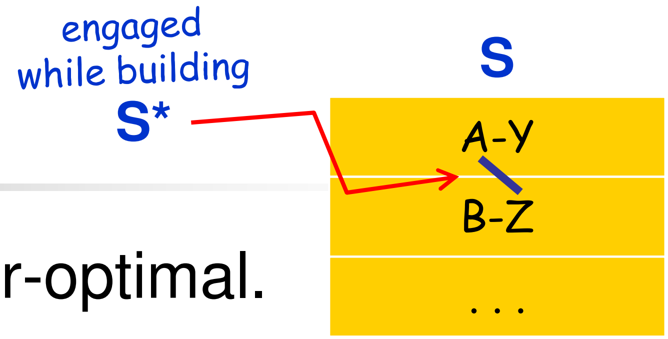
Proposer Optimality



- Claim. GS matching **S*** is proposer-optimal.
- Proof. (by contradiction)
  - Suppose some proposer is paired with someone other than their best partner. Proposers propose in decreasing order of preference $\Rightarrow$ some proposer is rejected by a valid partner.
  - Let **Y** be the proposer who is the first such rejection, and let **A** be the receiver who is first valid partner that rejects him.
  - Let **S** be a stable matching where **A** and **Y** are matched.
  - In building **S***, when **Y** is rejected, **A** forms (or reaffirms) engagement with a proposer, say **Z**, whom they prefer to **Y**.
  - Let **B** be **Z**'s partner in **S**.

Must exist since **Y** and **A** are valid partners

Proposer Optimality



- Claim. GS matching **S*** is proposer-optimal.
- Proof. (by contradiction)
  - Suppose some proposer is paired with someone other than their best partner. Proposers propose in decreasing order of preference ⇒ some proposer is rejected by a valid partner.
  - Let **Y** be the proposer who is the first such rejection, and let **A** be the receiver who is first valid partner that rejects **Y**.
  - Let **S** be a stable matching where **A** and **Y** are matched.
  - In building **S***, when **Y** is rejected, **A** forms (or reaffirms) engagement with a proposer, say **Z**, whom they prefer to **Y**.
  - Let **B** be **Z**'s partner in **S**.
  - In building **S***, **Z** is not rejected by any valid partner at the point when **Y** is rejected by **A**.
  - Thus, **Z** prefers **A** to **B**.
  - But **A** prefers **Z** to **Y**.
  - Thus **A**-**Z** is unstable in **S**. ▪

since Y was the first to be rejected by a valid partner

Stable Matching Summary

- Stable matching problem. Given preference profiles of two groups of n people, find a stable matching.

> Nobody prefer to be with each other than with their assigned partner

- Gale-Shapley algorithm. Finds a stable matching in $O(n^2)$ time.

- Proposer-optimality. In GS, each proposer receives best valid partner.

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

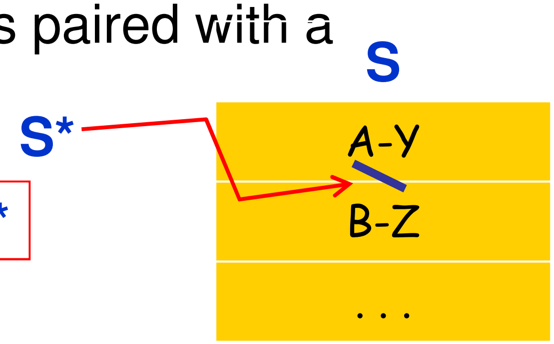
- Q. Does proposer-optimality come at the expense of the receivers?

Receiver Pessimality

- Receiver-pessimal assignment. Each receiver receives worst valid partner.

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

- Proof. (Contradiction again).
  - Suppose **A**-**Z** matched in **S***, but **Z** is not worst valid partner for **A**.
  - There exists stable matching **S** in which **A** is paired with a proposer, say **Y**, whom **A** likes less than **Z**.
  - Let **B** be **Z**'s partner in **S**.
  - **Z** prefers **A** to **B**. ← proposer-optimality of S*
  - Thus, **A**-**Z** is an unstable in **S**. ▪



Extensions: Matching Residents to Hospitals

- Original: Proposers ≈ hospitals, Receivers ≈ med school residents.
- Variant 1. Some participants declare others as unacceptable.
- Variant 2. Unequal number of proposers and receivers.
- Variant 3. Limited polygamy.

e.g. resident **A** unwilling to work in Cleveland

e.g. hospital **X** wants to hire **3** residents

- Def. Matching **S** is unstable if there is a hospital **h** and resident **r** such that:
  - **h** and **r** are acceptable to each other; and
  - either **r** is unmatched, or **r** prefers **h** to her assigned hospital; and
  - either **h** does not have all its places filled, or **h** prefers **r** to at least one of its assigned residents.

Application: Matching Residents to Hospitals

- NRMP. (National Resident Matching Program)
  - Original use just after WWII. ← predates computer usage
  - Ides of March, 23,000+ residents.

- Rural hospital dilemma.
  - Certain hospitals (mainly in rural areas) were unpopular and declared unacceptable by many residents.
  - Rural hospitals were under-subscribed in NRMP matching.
  - How can we find stable matching that benefits "rural hospitals"?

- Rural Hospital Theorem. Rural hospitals get exactly same residents in every stable matching!

- Note: Pre-1995 NRMP favored hospitals (they proposed). Changed in 1995 to favor residents (after a lawsuit).

Lessons Learned

- Powerful ideas learned in course.
  - Isolate underlying structure of problem.
  - Create useful and efficient algorithms.

- Potentially deep social ramifications.
  [legal disclaimer]

Deceit: Machiavelli Meets Gale-Shapley

- Q. Can there be an incentive to misrepresent your preference profile?
  - Assume you know propose-and-reject algorithm will be run and who will be proposers.
  - Assume that you know the preference profiles of all other participants.
- Fact. No, for proposers. Yes, for some receivers. No mechanism can guarantee a stable matching and be cheatproof.

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Group 0 Preference List

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Group 1 True Preference Profile

	1st	2nd	3rd
A	Y	Z	X
B	X	Y	Z
C	X	Y	Z

A Lies

Extra Slides

Stable Matching Problem

- Goal: Given **n** men and **n** women, find a "suitable" matching.
  - Participants rate members of opposite sex.
  - Each man lists women in order of preference from best to worst.
  - Each woman lists men in order of preference from best to worst.

favorite ↓ (1st) … least favorite ↓ (5th)

	1st	2nd	3rd	4th	5th
Victor	Brenda	Amy	Diane	Erika	Claire
Walter	Diane	Brenda	Amy	Claire	Erika
Xavier	Brenda	Erika	Claire	Diane	Amy
Yuri	Amy	Diane	Claire	Brenda	Erika
Zoran	Brenda	Diane	Amy	Erika	Claire

Men's Preference List

Stable Matching Problem

- Goal: Given **n** men and **n** women, find a "suitable" matching.
  - Participants rate members of opposite sex.
  - Each man lists women in order of preference from best to worst.
  - Each woman lists men in order of preference from best to worst.

favorite ↓ (1st) … least favorite ↓ (5th)

	1st	2nd	3rd	4th	5th
Amy	Zoran	Victor	Walter	Yuri	Xavier
Brenda	Xavier	Walter	Yuri	Victor	Zoran
Claire	Walter	Xavier	Yuri	Zoran	Victor
Diane	Victor	Zoran	Yuri	Xavier	Walter
Erika	Yuri	Walter	Zoran	Xavier	Victor

Women's Preference List