CSE 521: Design and Analysis of Algorithms I

Stable Matching

Paul Beame
Matching Residents to Hospitals

- **Goal:** Given a set of preferences among hospitals and medical school 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 students.

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

<table>
<thead>
<tr>
<th>Men's Preference Profile</th>
<th>Women's Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>favorite</strong></td>
<td><strong>favorite</strong></td>
</tr>
<tr>
<td><strong>least favorite</strong></td>
<td><strong>least favorite</strong></td>
</tr>
<tr>
<td>1st</td>
<td>1st</td>
</tr>
<tr>
<td>Amy</td>
<td>Yuri</td>
</tr>
<tr>
<td>Brenda</td>
<td>Xavier</td>
</tr>
<tr>
<td>Claire</td>
<td>Zoran</td>
</tr>
<tr>
<td>2nd</td>
<td>2nd</td>
</tr>
<tr>
<td>Yuri</td>
<td>Brenda</td>
</tr>
<tr>
<td>Amy</td>
<td>Claire</td>
</tr>
<tr>
<td>Claire</td>
<td>3rd</td>
</tr>
<tr>
<td>3rd</td>
<td>3rd</td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Brenda</td>
<td>Yuri</td>
</tr>
<tr>
<td>Claire</td>
<td>Xavier</td>
</tr>
<tr>
<td>Zoran</td>
<td>1st</td>
</tr>
</tbody>
</table>

Xavier, Yuri, Zoran, Amy, Brenda, Claire
Stable Matching Problem

- **Perfect matching**: everyone is matched monogamously.
  - Each man gets exactly one woman.
  - Each woman gets exactly one man.

- **Stability**: no incentive for some pair of participants to undermine assignment by joint action.
  - In matching $M$, an unmatched pair $m-w$ is unstable if man $m$ and woman $w$ prefer each other to current partners.
  - Unstable pair $m-w$ could each improve by eloping.

- **Stable matching**: perfect matching with no unstable pairs.

- **Stable matching problem**: Given the preference lists of $n$ men and $n$ women, find a stable matching if one exists.
Stable Matching Problem

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

<table>
<thead>
<tr>
<th>Men’s Preference Profile</th>
<th>Women’s Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>favorite</strong></td>
<td><strong>least favorite</strong></td>
</tr>
<tr>
<td><strong>1st</strong></td>
<td><strong>1st</strong></td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yuri</td>
<td>Brenda</td>
</tr>
<tr>
<td>Zoran</td>
<td>Amy</td>
</tr>
<tr>
<td><strong>2nd</strong></td>
<td><strong>2nd</strong></td>
</tr>
<tr>
<td>Amy</td>
<td>Brenda</td>
</tr>
<tr>
<td>Yuri</td>
<td>Claire</td>
</tr>
<tr>
<td>Claire</td>
<td>Amy</td>
</tr>
<tr>
<td><strong>3rd</strong></td>
<td><strong>3rd</strong></td>
</tr>
<tr>
<td>Claire</td>
<td>Xavier</td>
</tr>
<tr>
<td>Yuri</td>
<td>Yuri</td>
</tr>
<tr>
<td>Zoran</td>
<td>Xavier</td>
</tr>
</tbody>
</table>

Is assignment X-C, Y-B, Z-A stable?
Stable Matching Problem

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

A. No. Brenda and Xavier will hook up.
Stable Matching Problem

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

<table>
<thead>
<tr>
<th>Men’s Preference Profile</th>
<th>Women’s Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1st</strong></td>
<td><strong>2nd</strong></td>
</tr>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yuri</td>
<td>Brenda</td>
</tr>
<tr>
<td>Zoran</td>
<td>Amy</td>
</tr>
</tbody>
</table>

**favorite** ↓ **least favorite** ↓ **favorite** ↓ **least favorite** ↓
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.

<table>
<thead>
<tr>
<th>Adam</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>C</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>Chris</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>David</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

A-B, C-D  ⇒  B-C unstable
A-C, B-D  ⇒  A-B unstable
A-D, B-C  ⇒  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.

Initialize each person to be free.
while (some man is free and hasn't proposed to every woman) {
  Choose such a man \( m \)
  \( w = 1^{\text{st}} \) 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

- **Observation 1.** Men propose to women in decreasing order of preference.

- **Observation 2.** Once a woman is matched, she never becomes unmatched; she only "trades up."

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

$$n(n-1) + 1$$ proposals required
Proof of Correctness: Perfection

- **Claim.** All men and women get matched.
- **Proof.** (by contradiction)
  - Suppose, for sake of contradiction, that Zoran is not matched upon termination of algorithm.
  - Then some woman, say Amy, is not matched upon termination.
  - By Observation 2 (only trading up, never becoming unmatched), Amy was never proposed to.
  - But, Zoran proposes to everyone, since he ends up unmatched. ✷
Proof of Correctness: Stability

- **Claim.** No unstable pairs.
- **Proof.** (by contradiction)
  - Suppose \textbf{A-Z} is an unstable pair: each prefers each other to partner in Gale-Shapley matching \( S^* \).

  - **Case 1:** \textbf{Z} never proposed to \textbf{A}.
    - \( \Rightarrow \) \textbf{Z} prefers his GS partner to \textbf{A}.
    - \( \Rightarrow \) \textbf{A-Z} is stable.

  - **Case 2:** \textbf{Z} proposed to \textbf{A}.
    - \( \Rightarrow \) \textbf{A} rejected \textbf{Z} (right away or later)
    - \( \Rightarrow \) \textbf{A} prefers her GS partner to \textbf{Z}.
    - \( \Rightarrow \) \textbf{A-Z} is stable.

- In either case \textbf{A-Z} is stable, a contradiction. □
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?
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 men and women.
  - Assume men are named $1, \ldots, n$.
  - Assume women are named $1', \ldots, n'$.

- Engagements.
  - Maintain a list of free men, e.g., in a queue.
  - Maintain two arrays $\text{wife}[m]$, and $\text{husband}[w]$.
    - set entry to 0 if unmatched
    - if $m$ matched to $w$ then $\text{wife}[m] = w$ and $\text{husband}[w] = m$

- Men proposing.
  - For each man, maintain a list of women, ordered by preference.
  - Maintain an array $\text{count}[m]$ 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.

<table>
<thead>
<tr>
<th>Amy</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pref</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amy</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverse</td>
<td>4th</td>
<td>8th</td>
<td>2nd</td>
<td>5th</td>
<td>6th</td>
<td>7th</td>
<td>3rd</td>
<td>1st</td>
</tr>
</tbody>
</table>

\[
\text{for } i = 1 \text{ to } n \\
\text{inverse[pref[i]]} = i
\]

Amy prefers man 3 to 6 since \( \text{inverse[3]}=2 < 7=\text{inverse[6]} \)
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.

- A-X, B-Y, C-Z.
- A-Y, B-X, C-Z.
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. Man $m$ is a valid partner of woman $w$ if there exists some stable matching in which they are matched.

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

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

- No reason a priori to believe that man-optimal assignment is perfect, let alone stable.
- Simultaneously best for each and every man.
Man Optimality

- 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 ⇒ some man is rejected by a valid partner.
  - Let $Y$ be first such man, and let $A$ be the first valid woman 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 man, say $Z$, whom she prefers 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 this is the first rejection by a valid partner
Stable Matching Summary

- **Stable matching problem.** Given preference profiles of \( n \) men and \( n \) women, find a stable matching. 
  
  no man and woman prefer to be with each other than with their assigned partner

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

- **Man-optimality.** In version of GS where men propose, each man 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 man-optimality come at the expense of the women?
Woman Pessimality

- **Woman-pessimal assignment.** Each woman receives worst valid partner.

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

- **Proof.**  
  - 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 man, say $Y$, whom she likes less than $Z$.  
  - Let $B$ be $Z$'s partner in $S$.  
  - $Z$ prefers $A$ to $B$.  
  - Thus, $A-Z$ is an unstable in $S$. □
Extensions: Matching Residents to Hospitals

- **Ex:** Men ≈ hospitals, Women ≈ med school residents.

- **Variant 1.** Some participants declare others as unacceptable.

- **Variant 2.** Unequal number of men and women.

- **Variant 3.** Limited polygamy.

**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.

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

- **e.g. hospital X wants to hire 3 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!
Q. Can there be an incentive to misrepresent your preference profile?
- Assume you know men’s propose-and-reject algorithm will be run.
- Assume that you know the preference profiles of all other participants.

Fact. No, for any man. Yes, for some women. No mechanism can guarantee a stable matching and be cheatproof.

### Men’s Preference List

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Yuri</td>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Zoran</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

### Women’s True Preference Profile

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Y</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td>Brenda</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Claire</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

Amy Lies