CMPT 307

Data Structures & Algorithms

Valentine Kabanets

kabanets@cs.sfu.ca

(No class this Wed, May 8.)

What is this course about?

Problem

Given: Instance

Want: Solution

Algorithm

Given: Input instance

Finds: Solution

Various Methods for Designing Algorithms
Example: Stable Marriage Problem

Amy \{ n \text{ women} \}
Bertha
Clare

Xavier \{ n \text{ men} \}
Yancey
Zeus

- Each person ranks the people of the opposite sex
- Want a perfect matching
- Want total happiness
- Stability
Stable matching problem

**Goal.** Given a set of \( n \) men and a set of \( n \) women, find a "suitable" matching.
- Participants rank 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</th>
<th>Women</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
</tr>
<tr>
<td>Yancey</td>
<td>Bertha</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Men</th>
<th>Women</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Yancey</td>
</tr>
<tr>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Zeus</td>
<td>Yancey</td>
</tr>
</tbody>
</table>

Matching: set of disjoint couples

Perfect matching: everyone is matched

Stability?
Unstable pair\( (m, w) \):

- Not a matched couple \textit{BUT}\( \quad \)
- \( m \) prefers \( w \) to his match, \&
- \( w \) prefers \( m \) to her match.

**Def.** Given a perfect matching \( S \), man \( m \) and woman \( w \) are \textit{unstable} if:
- \( m \) prefers \( w \) to his current partner.
- \( w \) prefers \( m \) to her current partner.

**Key point.** An unstable pair \( m-w \) could each improve partner by joint action.

<table>
<thead>
<tr>
<th>Xavier</th>
<th>Amy</th>
<th>Bertha</th>
<th>Clare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yancey</td>
<td>Bertha</td>
<td>Amy</td>
<td>Clare</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
<td>Bertha</td>
<td>Clare</td>
</tr>
</tbody>
</table>

| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

Bertha and Xavier are an unstable pair
Stable matching problem

**Def.** A **stable matching** is a 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).
- Natural, desirable, and self-reinforcing condition.
- Individual self-interest prevents any man–woman pair from eloping.

\[
\begin{array}{|c|c|c|}
\hline
\text{Xavier} & \text{Amy} & \text{Bertha} \\
\text{Yancey} & \text{Bertha} & \text{Amy} \\
\text{Zeus} & \text{Amy} & \text{Bertha} \\
\hline
\end{array}
\quad
\begin{array}{|c|c|c|}
\hline
\text{Amy} & \text{Yancey} & \text{Xavier} \\
\text{Bertha} & \text{Xavier} & \text{Yancey} \\
\text{Clare} & \text{Xavier} & \text{Yancey} \\
\hline
\end{array}
\]

a perfect matching \( S = \{ X-A, Y-B, Z-C \} \)

---

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.

\[
\begin{array}{|c|c|c|}
\hline
\text{Adam} & \text{B} & \text{C} \\
\text{Bob} & \text{C} & \text{A} \\
\text{Chris} & \text{A} & \text{B} \\
\text{Doofus} & \text{A} & \text{B} \\
\hline
\end{array}
\]

\[\text{no perfect matching is stable}\]

\[A-B, C-D \Rightarrow B-C\] unstable
\[A-C, B-D \Rightarrow A-B\] unstable
\[A-D, B-C \Rightarrow A-C\] unstable

**Observation.** Stable matchings need not exist for stable roommate problem.
In each round:
- a free man $M$ proposes to the top woman $W$ on his list not proposed to before
- woman $W$ evaluates the proposal:
  - if $W$ is free, she says yes
  - if $W$ is engaged to $M'$, then
    - if $M$ is preferable to $M'$, then $W$ accepts $M$ & frees $M'$

---

**Gale-Shapley deferred acceptance algorithm**

An intuitive method that guarantees to find a stable matching.

**Gale-Shapley (preference lists for men and women)**

1. **Initialize** $S$ to empty matching.
2. **While** (some man $m$ is unmatched and hasn't proposed to every woman)
   - $m \leftarrow$ first woman on $m$'s list to whom $m$ has not yet proposed.
   - **If** ($w$ is unmatched)
     - Add pair $m$–$w$ to matching $S$.
   - **Else if** ($w$ prefers $m$ to her current partner $m'$)
     - Remove pair $m$–$w$ from matching $S$.
     - Add pair $m$–$w$ to matching $S$.
   - **Else**
     - $w$ rejects $m$.

---

```
// pseudocode
for m := all men do
    m := empty
for w := all women do
    w := empty
for m := men do
    while (m has not proposed to all women) do
        w := first woman on m's list not yet proposed
        if w is unmatched then
            add (m, w) to S
        else if w prefers m to her current partner m' then
            remove (m', w) from S
            add (m, w) to S
        else
            w rejects m
    return S
```
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.

Pf. Each time through the while loop a man proposes to a new woman. There are only $n^2$ possible proposals. •

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victor</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>Wyatt</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>Xavier</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Yancy</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Zeus</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Bertha</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
</tr>
<tr>
<td>Clare</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>W</td>
</tr>
<tr>
<td>Diane</td>
<td>Z</td>
<td>V</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td>Erika</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
</tr>
</tbody>
</table>

$n(n-1) + 1$ proposals required

---

Proof of correctness: perfection

Claim. In Gale-Shapley matching, all men and women get matched.

Pf. [by contradiction]

- Suppose, for sake of contradiction, that Zeus is not matched upon termination of GS algorithm.
- Then some woman, say Amy, is not matched upon termination.
- By Observation 2, Amy was never proposed to.
- But, Zeus proposes to everyone, since he ends up unmatched. •
Proof of correctness: stability

**Claim.** In Gale-Shapley matching, there are no unstable pairs.

**Pf.** Suppose the GS matching \( S^* \) does not contain the pair \( A-Z \).

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

- **Case 2:** \( Z \) proposed to \( A \).
  - \( A \) rejected \( Z \) (right away or later)
  - \( A \) prefers her GS partner \( Y \) to \( Z \).
  - \( A-Z \) is stable.
- In either case, the pair \( A-Z \) is stable.

---

**Summary**

**Stable matching problem.** Given \( n \) men and \( n \) women, and their preferences, find a stable matching if one exists.


**Q.** How to implement GS algorithm efficiently?

**Q.** If there are multiple stable matchings, which one does GS find?
Efficient implementation

Efficient implementation. We describe an \(O(n^2)\) time implementation.

Representing men and women.
- Assume men are named \(1, \ldots, n\).
- Assume women are named \(1', \ldots, n'\).

Representing the matching.
- Maintain a list of free men (in a stack or queue).
- Maintain two arrays \(\text{wife}[m]\) and \(\text{husband}[w]\).
  - if \(m\) matched to \(w\), then \(\text{wife}[m] = w\) and \(\text{husband}[w] = m\)
  - set entry to 0 if unmatched

Men proposing.
- For each man, maintain a list of women, ordered by preference.
- For each man, maintain a pointer to woman in list for next proposal.

---

Efficient implementation (continued)

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>pre[i]</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></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>inverse</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></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

For \(i = 1\) to \(n\)

\[
\text{inverse}[\text{pre[i]]} = i
\]
Understanding the solution

For a given problem instance, there may be several stable matchings.

- Do all executions of GS algorithm yield the same stable matching?
- If so, which one?

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
<tr>
<td>Yancey</td>
<td>Bertha</td>
<td>Amy</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
</tbody>
</table>

an instance with two stable matching: \( M = \{ A \rightarrow X, B \rightarrow Y, C \rightarrow Z \} \) and \( M' = \{ A \rightarrow Y, B \rightarrow X, C \rightarrow Z \} \)

Understanding the solution

**Def.** Woman \( w \) is a **valid partner** of man \( m \) if there exists some stable matching in which \( m \) and \( w \) are matched.

**Ex.**

- Both Amy and Bertha are valid partners for Xavier.
- Both Amy and Bertha are valid partners for Yancey.
- Clare is the only valid partner for Zeus.

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
<tr>
<td>Yancey</td>
<td>Bertha</td>
<td>Amy</td>
</tr>
<tr>
<td>Zeus</td>
<td>Amy</td>
<td>Bertha</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yancey</td>
<td>Xavier</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
</tbody>
</table>

an instance with two stable matching: \( M = \{ A \rightarrow X, B \rightarrow Y, C \rightarrow Z \} \) and \( M' = \{ A \rightarrow Y, B \rightarrow X, C \rightarrow Z \} \)
Understanding the solution

**Def.** Woman \( w \) is a **valid partner** of man \( m \) if there exists some stable matching in which \( m \) and \( w \) are matched.

**Man-optimal assignment.** Each man receives best valid partner.
- Is it perfect?
- Is it stable?

**Claim.** All executions of GS yield **man-optimal** assignment.

**Corollary.** Man-optimal assignment is a stable matching!

---

Man optimality

**Claim.** GS matching \( S^o \) is man-optimal.

**Pf.** [by contradiction]
- Suppose a man is matched with someone other than best valid partner.
- Men propose in decreasing order of preference
  \( \Rightarrow \) some man is rejected by valid partner during GS.
- 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.
- When \( Y \) is rejected by \( A \) in GS, \( A \) forms (or reaffirms) engagement with a man, say \( Z \).
  \( \Rightarrow A \) **prefers** \( Z \) to \( Y \).
- Let \( B \) be partner of \( Z \) in \( S \).
- \( Z \) has not been rejected by any valid partner (including \( B \)) at the point when \( Y \) is rejected by \( A \).
  \( \leftarrow \) rejection by a valid partner
- Thus, \( Z \) has not yet proposed to \( B \) when he proposes to \( A \).
  \( \Rightarrow Z \) **prefers** \( A \) to \( B \).
- Thus \( A-Z \) is unstable in \( S \), a contradiction. 

---

[Note: Diagram of stable matching with notes on top of it is included.]
**Woman pessimality**

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

**A.** Yes.

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

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

**Pf.** [by contradiction]

- Suppose $A-Z$ matched in $S^m$ 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 $Z$.
  
  $\Rightarrow A \text{ prefers } Z \text{ to } Y.$

- Let $B$ be the partner of $Z$ in $S$. By man-optimality, $A$ is the best valid partner for $Z$.
  
  $\Rightarrow Z \text{ prefers } A \text{ to } B.$

- Thus, $A-Z$ is an unstable pair in $S$, a contradiction. $\blacksquare$

---

**Deceit: Machiavelli meets Gale-Shapley**

**Q.** Can there be an incentive to misrepresent your preference list?

- Assume you know men's propose-and-reject algorithm will be run.
- Assume preference lists of all other participants are known.

**Fact.** No, for any man; yes, for some women.

<table>
<thead>
<tr>
<th>men's preference list</th>
<th>women's preference list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2nd</td>
</tr>
<tr>
<td>X</td>
<td>A</td>
</tr>
<tr>
<td>Y</td>
<td>B</td>
</tr>
<tr>
<td>Z</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Y</td>
<td>X</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>Y</td>
</tr>
</tbody>
</table>

Amy lies

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>Y</td>
</tr>
</tbody>
</table>
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. ← hospital X wants to hire 3 residents

Def. Matching is $S$ 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.

Historical context

National resident matching program (NRMP).

- Centralized clearinghouse to match med-school students to hospitals.
- Began in 1952 to fix unraveling of offer dates.
- Originally used the "Boston Pool" algorithm.
- Algorithm overhauled in 1998.
  - med-school student optimal
  - deals with various side constraints
    (e.g., allow couples to match together)
  - 38,000+ residents for 26,000+ positions.

The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design

By Alvin E. Roth and Elliott S. Eisenberg®

We report on the design of the new clearinghouse adopted by the National Resident Matching Program, which randomized $50%$ approximately 36,000 positions. Because the market has complements between applicants and between positions, the theory of stable matching markets does not apply directly. However, computational experiments show the limited potential pool approximations. For instance, the role of stable matching, and the opportunities for strategic manipulation, are not negligible small. A new kind of “core concept” reveals insights. While each applicant interviews only a small fraction of available positions, its impact. We also discuss engineering aspects of the design process. (JPE, 1978; 861-1841.)

stable matching is no longer guaranteed to exist

hospitals began making offers earlier and earlier, up to 2 years in advance
2012 Nobel Prize in Economics

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

Alvin Roth. Applied Gale-Shapley to matching new doctors with hospitals, students with schools, and organ donors with patients.

Social Implications: Fairness

Is GS algo is unfair to women (or to men)?

Q: How to define “fairness”?

Q: Is there an efficient algo to find a “fair” stable matching?

Algorithms influence our everyday lives! We need to take care.

4 Kinds of Algorithms
4 Kinds of Algorithms

**Obvious:** by problem definition

**Methodical:** by a method good for a class of problems

**Clever:** specific to a problem

**Miraculous:** how did anyone think of this?