Chapter 1

Introduction: Some Representative Problems
Algorithm design overview

• Problem definition and formulation
• Algorithm design
• Proof of correctness
• Efficient implementation
  – choosing efficient data structure
• Optimality analysis
1.1 A First Problem: Stable Matching
**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></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>Amy</td>
<td>Bertha</td>
<td>Clare</td>
</tr>
<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>

**Men’s Preference Profile**

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

**Women’s Preference Profile**
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>Favorite</th>
<th>Least Favorite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2nd</td>
</tr>
<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>

Men's Preference Profile

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

Women's Preference Profile
**Stable Matching Problem**

Q. Is assignment X-C, Y-B, Z-A stable?
A. No. Bertha and Xavier will hook up.

### Men's Preference Profile

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

### Women's Preference Profile

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

*Men's Preference Profile*  
*Women's Preference Profile*
Stable Matching Problem

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

**A.** Yes.

---

**Men's Preference Profile**

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

**Women's Preference Profile**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amy</td>
<td>Yancey</td>
<td>Xavier</td>
<td>Zeus</td>
</tr>
<tr>
<td>Bertha</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
<td>Yancey</td>
<td>Zeus</td>
</tr>
</tbody>
</table>
Propose-And-Reject Algorithm


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

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></th>
<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>
<td>E</td>
</tr>
<tr>
<td>Wyatt</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>E</td>
</tr>
<tr>
<td>Xavier</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>Yancey</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>Zeus</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<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>
<td>V</td>
</tr>
<tr>
<td>Bertha</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>W</td>
</tr>
<tr>
<td>Clare</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td>Diane</td>
<td>Z</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Erika</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

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

Claim. All men and women get matched.
Pf. (by contradiction)
  - Suppose, for sake of contradiction, 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 proposes to everyone, since he ends up unmatched.
Proof of Correctness: Stability

Claim. No unstable pairs.

Pf. (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.
    - $\Rightarrow$ Z prefers his GS partner to A.
    - $\Rightarrow$ A-Z is stable.

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

- In either case 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?
Efficient Implementation

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

Representing men and women.
- Assume men are named 1, ..., n.
- Assume women are named 1', ..., 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>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>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>

Amy prefers man 3 to 6 since $\text{inverse}[3] < \text{inverse}[6]$

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

Claim. All executions of GS yield 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.
Pf. (by contradiction)
  - Suppose some man is paired with someone other than best partner.
    Men propose in decreasing order of preference $\Rightarrow$ some man is
    rejected by valid partner.
  - Let $Y$ be first such man, and let $A$ be first valid
    woman that rejects him.
  - Let $S$ be a stable matching where $A$ and $Y$ are matched.
  - 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 $S$.
  - $Z$ 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 $S$.  

\[ s \]

<table>
<thead>
<tr>
<th>Amy-Yancey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bertha-Zeus</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

since this is 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 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 $S^*$. 

**Pf.**
- 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 $B$. $\leftarrow$ man-optimality
- Thus, $A-Z$ is an unstable in $S$. $\blacksquare$
1.2 Five Representative Problems
**Interval Scheduling**

**Input.** Set of jobs with start times and finish times.

**Goal.** Find maximum cardinality subset of mutually compatible jobs.

jobs don't overlap
**Weighted Interval Scheduling**

**Input.** Set of jobs with start times, finish times, and weights.

**Goal.** Find maximum weight subset of mutually compatible jobs.
Bipartite Matching

Input. Bipartite graph.
Goal. Find maximum cardinality matching.
Input. Graph.

Goal. Find maximum cardinality independent set.

subset of nodes such that no two joined by an edge
Competitive Facility Location

Input. Graph with weight on each each node.

Game. Two competing players alternate in selecting nodes. Not allowed to select a node if any of its neighbors have been selected.

Goal. Select a maximum weight subset of nodes.

Second player can guarantee 20, but not 25.

Is there a strategy for the second player so that no matter
Five Representative Problems

Variations on a theme: independent set.

Interval scheduling: \( n \log n \) greedy algorithm.
Weighted interval scheduling: \( n \log n \) dynamic programming algorithm.
Bipartite matching: \( n^k \) max-flow based algorithm.
Independent set: NP-complete.
Competitive facility location: PSPACE-complete.

PSPACE-complete: the problem can be solved using an amount of memory that is polynomial in the input length (polynomial space) and every other problem that can be solved in polynomial space can be transformed to it in polynomial time.