Chapter 1

Introduction: Some Representative Problems

Mostly Adopted from

Slides by Kevin Wayne
Copyright © 2005 Pearson-Addison Wesley
All rights reserved
Stable Matching

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 its assigned hospital.
- y prefers x to one of its admitted students.

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

Consider a set $M = \{m_1, \ldots, m_n\}$ of n men, and a set $W = \{w_1, \ldots, w_n\}$ of n women. Let $M \times W$ denote the set of all possible ordered pairs of the form $(m, w)$, where $m \in M$ and $w \in W$. A matching $S$ is a set of ordered pairs, each from $M \times W$, with the property that each member of $M$ and each member of $W$ appears in at most one pair in $S$. A perfect matching $S$ is a matching with the property that each member of $M$ and each member of $W$ appears in exactly one pair in $S$. Note: man and woman sets are just examples; the problem is to match members of two disjoint sets of equal cardinality given ranking lists (preferences) [no ties are allowed].
Stable Matching Problem

**Goal.** Given $n$ men and $n$ women, find a "suitable" matching.

- Participants rate each member 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>least favorite</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1$^{st}$</td>
<td>1$^{st}$</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>
<tr>
<td></td>
<td>Bertha</td>
</tr>
<tr>
<td></td>
<td>Clare</td>
</tr>
<tr>
<td></td>
<td>Yancey</td>
</tr>
<tr>
<td></td>
<td>Xavier</td>
</tr>
<tr>
<td></td>
<td>Zeus</td>
</tr>
<tr>
<td></td>
<td>Bertha</td>
</tr>
<tr>
<td></td>
<td>Amy</td>
</tr>
<tr>
<td></td>
<td>Yancey</td>
</tr>
<tr>
<td></td>
<td>Xavier</td>
</tr>
<tr>
<td></td>
<td>Zeus</td>
</tr>
</tbody>
</table>

**Note:** In practice, things may be a bit more complicated; e.g., in college admissions, each college looking for several applicants, but each student looking for one college; there may be more (fewer) applicants than available slots for colleges; each student does not apply to every college. More examples: job applicants and multiple offers, etc.
Stable Matching Problem

Perfect matching:
- Each man is matched with exactly one woman.
- Each woman is matched with 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 create problems in many scenarios.

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?

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

Women's Preference Profile

<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><strong>Bertha</strong></td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
<tr>
<td>Clare</td>
<td>Xavier</td>
<td>Yancey</td>
</tr>
<tr>
<td>Zeus</td>
<td>Clare</td>
<td>Yancey</td>
</tr>
</tbody>
</table>
Stable Matching Problem

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

Men's Preference Profile

<table>
<thead>
<tr>
<th>Men's Preference Profile</th>
<th>Women's Preference Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorite</td>
<td>least favorite</td>
</tr>
<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>

| Amy                      | Yancey                    | Xavier                    | Zeus                      |
| Bertha                  | Xavier                    | Yancey                    | Zeus                      |
| Clare                   | Xavier                    | Yancey                    | Zeus                      |
An Interesting Observation

Consider two scenarios of two men and two women:

- Two men (m1, m2) and two women (w1, w2): m1 prefers w1 to w2, m2 prefers w1 to w2, w1 prefers m1 to m2 and w2 prefers m1 to m2.
  - Only one stable matching {(m1, w1), (m2, w2)}; why? Because, the other possible matching {(m1, w2), (m2, w1)} is not stable since the pair (m1, w1) form an instability.

- Two men (m1, m2) and two women (w1, w2): m1 prefers w1 to w2, m2 prefers w2 to w1, w1 prefers m2 to m1, w2 prefers m1 to m2.
  - Observe that two men’s preferences mesh perfectly and so do women’s; men’s and women’s preferences clash.
  - The matching {(m1, w1), (m2, w2)} is stable (both men are as happy as possible); so is the matching {(m2, w1), (m1, w2)} (both women are as happy as possible);

**Lesson:** In some instances (depending on specific preferences), multiple stable matchings are possible.

**Question:** Does stable matching always exists? If one exists, how to find it?
Stable Roommate Selection

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></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<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>Doofus</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

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 (we need to prove this) 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 = 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
}

Randomly?

Temporary assignment

One can easily adjust the algorithm having women to propose in stead of men. Run two versions of the algorithm on the same set of preferences. See if you can make any interesting observations looking at the outputs.
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 $O(n^2)$ iterations of while loop.

Proof. Each time through the while loop a man proposes to a new woman. There are only $O(n^2)$ possible proposals.

\[ n(n-1) + 1 \text{ proposals required} \]
Proof of Correctness: Perfection

Claim. All men and women get matched.

Proof. (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.

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 his GS partner to A.
    - ⇒ A-Z is stable.

  - Case 2: Z proposed to A.
    - ⇒ A rejected Z (right away or later)
    - ⇒ A prefers her GS partner to Z.
    - ⇒ 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>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>

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

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

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. 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.

Proof. (by contradiction)

- Suppose some man is paired with someone other than best partner. Men propose in decreasing order of preference ⇒ 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. □

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.

\[ \text{no man and woman prefer to be with each other than assigned partner} \]

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.

\[ w \text{ is a valid partner of } m \text{ if there exist some stable matching where } m \text{ and } w \text{ are paired} \]

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.
- Thus, A-Z is an unstable in S.  

$\leftarrow$ man-optimality

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

Def. Matching $S$ 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.
Application: Matching Residents to Hospitals

NRMP. (National Resident Matching Program)
- Original use just after WWII.
- 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!
Lessons Learned

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

Potentially deep social ramifications.  [legal disclaimer]
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.

![Graph showing jobs and their compatibility](image)

- Jobs b, e, and h do not overlap.

The diagram illustrates how to find the maximum cardinality subset of mutually compatible jobs.
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.
Independent Set

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 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.
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.
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 List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
</tr>
<tr>
<td>Victor</td>
</tr>
<tr>
<td>Wyatt</td>
</tr>
<tr>
<td>Xavier</td>
</tr>
<tr>
<td>Yancey</td>
</tr>
<tr>
<td>Zeus</td>
</tr>
</tbody>
</table>

*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.

<table>
<thead>
<tr>
<th>Women's Preference List</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>favorite</strong> ↓</td>
</tr>
<tr>
<td><strong>1st</strong></td>
</tr>
<tr>
<td>Amy</td>
</tr>
<tr>
<td>Bertha</td>
</tr>
<tr>
<td>Clare</td>
</tr>
<tr>
<td>Diane</td>
</tr>
<tr>
<td>Erika</td>
</tr>
</tbody>
</table>
Understanding the Solution*

Claim. The man-optimal stable matching is weakly Pareto optimal.

Pf. Let A be last woman in some execution of GS algorithm to receive a proposal.
No man is rejected by A since algorithm terminates when last woman receives first proposal.
No man matched to A will be strictly better off than in man-optimal 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 cheat proof.
Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Potentially deep social ramifications.  [legal disclaimer]

- Historically, men propose to women. Why not vice versa?
- Men: propose early and often.
- Men: be more honest.
- Women: ask out the guys.
- Theory can be socially enriching and fun!
- CS majors get the best partners!