CSC304 Lecture 13

Mechanism Design w/o Money 2: Stable Matching
Gale-Shapley Algorithm
Stable Matching

• Recap Graph Theory:

• In graph \( G = (V, E) \), a matching \( M \subseteq E \) is a set of edges with no common vertices
  ➢ That is, each vertex should have at most one incident edge
  ➢ A matching is perfect if no vertex is left unmatched.

• \( G \) is a bipartite graph if there exist \( V_1, V_2 \) such that \( V = V_1 \cup V_2 \) and \( E \subseteq V_1 \times V_2 \)
Stable Marriage Problem

• Bipartite graph, two sides with equal vertices
  ➢ n men and n women

  (old school terminology 😞)

• Each man has a ranking over women & vice versa
  ➢ E.g., Eden might prefer Alice > Tina > Maya
  ➢ And Tina might prefer Tony > Alan > Eden

• Want: a perfect, stable matching
  ➢ Match each man to a unique woman such that no pair of
    man \(m\) and woman \(w\) prefer each other to their current
    matches (such a pair is called a “blocking pair”)

CSC304 - Nisarg Shah
Why ranked preferences?

• Until now, we dealt with cardinal values.
  ➢ Our goal was welfare maximization.
  ➢ This was sensitive to the exact numerical values.

• Our goal here is stability.
  ➢ Stability is a property of the ranked preference.
  ➢ That is, you can check whether a matching is stable or not using only the ranked preferences.
  ➢ So ranked information suffices.
# Example: Preferences

<table>
<thead>
<tr>
<th></th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diane</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
</tr>
</tbody>
</table>

≥ ≥ ≥
Example: Matching 1

<table>
<thead>
<tr>
<th></th>
<th>Albert</th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
<td></td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Diane</th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
</tbody>
</table>

Question: Is this a stable matching?
Example: Matching 1

<table>
<thead>
<tr>
<th></th>
<th>Albert</th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
<td></td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Diane</th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
</tbody>
</table>

No, Albert and Emily form a **blocking pair**.
## Example: Matching 2

<table>
<thead>
<tr>
<th></th>
<th>Albert</th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
<td></td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Diane</th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
</tbody>
</table>
Example: Matching 2

<table>
<thead>
<tr>
<th></th>
<th>Albert</th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albert</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
<td></td>
</tr>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
<td></td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Diane</th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diane</td>
<td>Bradley</td>
<td>Albert</td>
<td>Charles</td>
<td></td>
</tr>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
<td></td>
</tr>
</tbody>
</table>

**Yes!** (Charles and Fergie are unhappy, but helpless.)
Does a stable matching always exist in the marriage problem?

Can we compute it in a strategyproof way?

Can we compute it efficiently?
Gale-Shapley 1962

• Men-Proposing Deferred Acceptance (MPDA):

1. Initially, no one has proposed, no one is engaged, and no one is matched.

2. While some man $m$ is unengaged:
   - $w \leftarrow m$’s most preferred woman to whom $m$ has not proposed yet
   - $m$ proposes to $w$
   - If $w$ is unengaged:
     - $m$ and $w$ are engaged
   - Else if $w$ prefers $m$ to her current partner $m'$
     - $m$ and $w$ are engaged, $m'$ becomes unengaged
   - Else: $w$ rejects $m$

3. Match all engaged pairs.
Example: MPDA

<table>
<thead>
<tr>
<th>Albert</th>
<th>Diane</th>
<th>Emily</th>
<th>Fergie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bradley</td>
<td>Emily</td>
<td>Diane</td>
<td>Fergie</td>
</tr>
<tr>
<td>Charles</td>
<td>Diane</td>
<td>Emily</td>
<td>Fergie</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diane</th>
<th>Bradley</th>
<th>Albert</th>
<th>Charles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emily</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
</tr>
<tr>
<td>Fergie</td>
<td>Albert</td>
<td>Bradley</td>
<td>Charles</td>
</tr>
</tbody>
</table>

- **= proposed**
- **= engaged**
- **= rejected**
Running Time

• **Theorem:** DA terminates in polynomial time (at most $n^2$ iterations of the outer loop)

• **Proof:**
  - In each iteration, a man proposes to someone to whom he has never proposed before.
  - $n$ men, $n$ women → at most $n^2$ proposals

• At termination, it must return a perfect matching.
Stable Matching

• **Theorem:** DA always returns a stable matching.

• **Proof by contradiction:**
  ➢ Assume \((m, w)\) is a blocking pair.
  
  ➢ **Case 1:** \(m\) never proposed to \(w\)
    
    o \(m\) cannot be unmatched o/w algorithm would not terminate.
    
    o Men propose in the order of preference.
    
    o Hence, \(m\) must be matched with a woman he prefers to \(w\)
    
    o \((m, w)\) is not a blocking pair
Stable Matching

• **Theorem**: DA always returns a stable matching.

• **Proof by contradiction:**
  - Assume \((m, w)\) is a blocking pair.
  - **Case 2**: \(m\) proposed to \(w\)
    - \(w\) must have rejected \(m\) at some point
    - Women only reject to get better partners
    - \(w\) must be matched at the end, with a partner she prefers to \(m\)
    - \((m, w)\) is not a blocking pair
Men-Optimal Stable Matching

• The stable matching found by MPDA is special.

• **Valid partner:** For a man $m$, call a woman $w$ a valid partner if $(m, w)$ is in some stable matching.

• **Best valid partner:** For a man $m$, a woman $w$ is the best valid partner if she is a valid partner, and $m$ prefers her to every other valid partner.
  ➢ Denote the best valid partner of $m$ by $best(m)$. 
Men-Optimal Stable Matching

• **Theorem:** Every execution of MPDA returns the men-optimal stable matching in which every man is matched to his best valid partner $\text{best}(m)$.

  ➢ Surprising that this is even a matching. E.g., why can’t two men have the same best valid partner?

  ➢ Every man is simultaneously matched with his best possible partner across all stable matchings

• **Theorem:** Every execution of MPDA produces the women-pessimal stable matching in which every woman is matched to her worst valid partner.
Men-Optimal Stable Matching

- **Theorem:** Every execution of MPDA returns the men-optimal stable matching.
- **Proof by contradiction:**
  - Let $S$ = matching returned by MPDA.
  - $m \leftarrow$ first man rejected by $\text{best}(m) = w$
  - $m' \leftarrow$ the man $w$ preferred more and thus rejected $m$
  - $w$ is valid for $m$, so $(m, w)$ part of stable matching $S'$
  - $w' \leftarrow$ woman $m'$ is matched to in $S'$
  - **Mic drop:** $S'$ cannot be stable because $(m', w)$ is a blocking pair.
Men-Optimal Stable Matching

- **Theorem:** Every execution of MPDA returns the men-optimal stable matching.

- **Proof by contradiction:**

  - Assume there is a blocking pair $(m, w)$ in $S$ such that $m$ prefers $w$ to $w'$ and $w$ hasn't proposed to $w'$.

  - Then, $w'$ is not yet rejected by a valid partner.

  - If $w'$ was rejected by a valid partner, then $w$ prefers $w'$ to $m$.

  - This implies $m$ prefers $w'$ to $w$.

  -矛盾，因为$S$是男人最优的匹配，所以不可能有$S'$的结构。
Strategyproofness

• **Theorem**: MPDA is strategyproof for men, i.e., reporting the true ranking is a weakly dominant strategy for every man.
  ➢ We’ll skip the proof of this.
  ➢ Actually, it is group-strategyproof.

• But the women might want to misreport.

• **Theorem**: No algorithm for the stable matching problem is strategyproof for both men and women.
Women-Proposing Version

• Women-Proposing Deferred Acceptance (WPDA)
  ➢ Just flip the roles of men and women

• Strategyproof for women, not strategyproof for men

• Returns the women-optimal and men-pessimal stable matching
Extensions

• Unacceptable matches

➢ Allow every agent to report a partial ranking
➢ If woman $w$ does not include man $m$ in her preference list, it means she would rather be unmatched than matched with $m$. And vice versa.
➢ $(m, w)$ is blocking if each prefers the other over their current state (matched with another partner or unmatched)
➢ Just $m$ (or just $w$) can also be blocking if they prefer being unmatched than be matched to their current partner

• Magically, DA still produces a stable matching.
Extensions

• Resident Matching (or College Admission)
  ➢ Men $\rightarrow$ residents (or students)
  ➢ Women $\rightarrow$ hospitals (or colleges)
  ➢ Each side has a ranked preference over the other side
  ➢ But each hospital (or college) $q$ can accept $c_q > 1$ residents (or students)
  ➢ Many-to-one matching

• An extension of Deferred Acceptance works
  ➢ Resident-proposing (resp. hospital-proposing) results in resident-optimal (resp. hospital-optimal) stable matching
Extensions

• For ~20 years, most people thought that these problems are very similar to the stable marriage problem

• Roth [1985]:
  - No stable matching algorithm exists such that truth-telling is a weakly dominant strategy for hospitals (or colleges).
Extensions

• Roommate Matching
  ➢ Still one-to-one matching
  ➢ But no partition into men and women
    o “Generalizing from bipartite graphs to general graphs”
  ➢ Each of $n$ agents submits a ranking over the other $n - 1$ agents

• Unfortunately, there are instances where no stable matching exist.
  ➢ A variant of DA can still find a stable matching if it exists.
  ➢ Due to Irving [1985]
NRMP: Matching in Practice

• 1940s: Decentralized resident-hospital matching
  ➢ Markets “unraveled”, offers came earlier and earlier, quality of matches decreased

• 1950s: NRMP introduces centralized “clearinghouse”

• 1960s: Gale-Shapley introduce DA

• 1984: Al Roth studies NRMP algorithm, finds it is really a version of DA!

• 1970s: Couples increasingly don’t use NRMP

• 1998: NRMP implements matching with couple constraints (stable matchings may not exist anymore…)

• More recently, DA applied to college admissions