Lecture 2:
• Stable Matching: the Gale-Shapley Algorithm

Sep 11, 2018
National Residency Matching Program

- National system for matching US medical school graduates to medical residencies
  - Roughly 40,000 doctors per year
  - Assignment is almost entirely algorithmic
Labor Markets

• Most labor markets are frustrating
  • Not everyone can get their favorite job
  • The market is decentralized

• Decentralized labor markets are confusing

  Nobody has all the information
  Whatever you do could lead to an untenable
Centralized Labor Markets

• What if we could just assign jobs?

• What information would we want?
  - List of doctors and hospitals
  - Preferences (ranking ordinal preferences) from each doctor and each hospital

• How would we choose the assignment?
  - Stable
Matchings

In the real world, doctors only rank ≤ 15 hospitals

- We are given the following information
  - $n$ doctors $d_1 \ldots d_n$
  - $n$ hospitals $h_1 \ldots h_n$
  - each doctor’s ranking of hospitals $d_1 : h_2 > h_3 > h_1$
  - each hospital’s ranking of doctors $h_1 : d_1 > d_3 > d_2$

<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>MGH</td>
<td>Bob</td>
<td>Alice</td>
<td>Dorit</td>
<td>Ernie</td>
<td>Clara</td>
</tr>
<tr>
<td>BW</td>
<td>Dorit</td>
<td>Bob</td>
<td>Alice</td>
<td>Clara</td>
<td>Ernie</td>
</tr>
<tr>
<td>BID</td>
<td>Bob</td>
<td>Ernie</td>
<td>Clara</td>
<td>Dorit</td>
<td>Alice</td>
</tr>
<tr>
<td>MTA</td>
<td>Alice</td>
<td>Dorit</td>
<td>Clara</td>
<td>Bob</td>
<td>Ernie</td>
</tr>
<tr>
<td>CH</td>
<td>Bob</td>
<td>Dorit</td>
<td>Alice</td>
<td>Ernie</td>
<td>Clara</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>Alice</td>
<td>CH</td>
<td>MGH</td>
<td>BW</td>
<td>MTA</td>
<td>BID</td>
</tr>
<tr>
<td>Bob</td>
<td>BID</td>
<td>BW</td>
<td>MTA</td>
<td>MGH</td>
<td>CH</td>
</tr>
<tr>
<td>Clara</td>
<td>BW</td>
<td>BID</td>
<td>MTA</td>
<td>CH</td>
<td>MGH</td>
</tr>
<tr>
<td>Dorit</td>
<td>MGH</td>
<td>CH</td>
<td>MTA</td>
<td>BID</td>
<td>BW</td>
</tr>
<tr>
<td>Ernie</td>
<td>MTA</td>
<td>BW</td>
<td>CH</td>
<td>BID</td>
<td>MGH</td>
</tr>
</tbody>
</table>
Matchings

- A **matching** $M$ is a set of doctor-hospital pairs
  - $M = \{ (d_1, h_2), (d_2, h_3) \}$
  - matching: no doctor/hospital appears twice
  - perfect matching: every doctor/hospital appears once
  - “$d$ is matched to $h$” $(d, h) \in M$

“$d$ is matched” $\exists h \text{ s.t. } (d, h) \in M$

“$d$ is unmatched”
Stable Matchings

- A matching $M$ is **unstable** if some doctor-hospital pair prefer one another to their mate in $M$

- Instabilities

  1. $d, h$ such that $d$ is matched to $h'$, $h$ is unmatched, but $d \succ h'$

  2. $d, h$ such that $h$ is matched to $d'$, $d$ is unmatched, but $h \succ d'$

  3. $d, h$ such that $d$ is matched to $h'$, $h$ is matched to $d'$, but $d \succ h'$ and $h \succ d'$
Ask the Audience

- Either find a stable matching or convince yourself that there is no stable matching

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGH</td>
<td>Alice</td>
<td>Bob</td>
<td>Clara</td>
</tr>
<tr>
<td>BW</td>
<td>Bob</td>
<td>Clara</td>
<td>Alice</td>
</tr>
<tr>
<td>BID</td>
<td>Alice</td>
<td>Clara</td>
<td>Bob</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>BW</td>
<td>BID</td>
<td>MGH</td>
</tr>
<tr>
<td>Bob</td>
<td>BID</td>
<td>MGH</td>
<td>BW</td>
</tr>
<tr>
<td>Clara</td>
<td>MGH</td>
<td>BID</td>
<td>BW</td>
</tr>
</tbody>
</table>

\[
M = \{ (\text{Alice}, \text{BU}) , (\text{Bob}, \text{MGH}) , (\text{Clara}, \text{BID}) \}
\]

\[
M' = \{ (\text{Alice}, \text{BID}) , (\text{Bob}, \text{MGH}) , (\text{Clara}, \text{BU}) \}
\]

\[
M'' = \{ (\text{Alice}, \text{BU}) , (\text{Bob}, \text{BID}) , (\text{Clara}, \text{MGH}) \}
\]
Gale-Shapley Algorithm

• Let M be empty
• While (some hospital h is unmatched):
  • If (h has offered a job to everyone): break
  • Else: let d be the highest-ranked doctor to which h has not yet offered a job
  • h makes an offer to d:
    • If (d is unmatched):
      • d accepts, add (d,h) to M
    • ElseIf (d is matched to h’ & d: h’ > h):
      • d rejects, do nothing
    • ElseIf (d is matched to h’ & d: h > h’):
      • d accepts, remove (d,h’) from M and add (d,h) to M
• Output M
### Gale-Shapley Demo

<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><strong>MGH</strong></td>
<td>Bob</td>
<td>Alice</td>
<td>Dorit</td>
<td>Ernie</td>
<td>Clara</td>
</tr>
<tr>
<td><strong>BW</strong></td>
<td>Dorit</td>
<td>Bob</td>
<td>Alice</td>
<td>Clara</td>
<td>Ernie</td>
</tr>
<tr>
<td><strong>BID</strong></td>
<td>Bob</td>
<td>Ernie</td>
<td>Clara</td>
<td>Dorit</td>
<td>Alice</td>
</tr>
<tr>
<td><strong>MTA</strong></td>
<td>Alice</td>
<td>Dorit</td>
<td>Clara</td>
<td>Bob</td>
<td>Ernie</td>
</tr>
<tr>
<td><strong>CH</strong></td>
<td>Bob</td>
<td>Dorit</td>
<td>Alice</td>
<td>Ernie</td>
<td>Clara</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><strong>Alice</strong></td>
<td>CH</td>
<td>MGH</td>
<td>BID</td>
<td>BW</td>
<td>MTA</td>
</tr>
<tr>
<td><strong>Bob</strong></td>
<td>BID</td>
<td>BW</td>
<td>MTA</td>
<td>CH</td>
<td>MGH</td>
</tr>
<tr>
<td><strong>Clara</strong></td>
<td>BW</td>
<td>BID</td>
<td>MTA</td>
<td>CH</td>
<td>MGH</td>
</tr>
<tr>
<td><strong>Dorit</strong></td>
<td>MGH</td>
<td>CH</td>
<td>MTA</td>
<td>BID</td>
<td>BW</td>
</tr>
<tr>
<td><strong>Ernie</strong></td>
<td>MTA</td>
<td>BW</td>
<td>CH</td>
<td>BID</td>
<td>MGH</td>
</tr>
</tbody>
</table>
Observations

• Hospitals make offers in descending order

  If h made offers to d, d', and d got an offer first, then h: d > d'

• Doctors that get a job never become unemployed

  If a doctor has ever had a job, they will always have a job.

• Doctors accept offers in ascending order

  If a doctor was ever matched to h, then d is never matched to a lower ranked hospital than h.
Questions about the Gale-Shapley Algorithm:

- Will this algorithm terminate?
- Does it output a perfect matching?
- Does it output a stable matching? *(Does one even exist?)*
- How do we implement this algorithm efficiently?
GS Algorithm: Termination

• **Claim:** The GS algorithm terminates after $n^2$ iterations of the main loop

  • There are only $n^2$ doctor-hospital pairs
  • Never make the same offer twice
  • Alg halts if all offers are made
**GS Algorithm: Perfect Matching**

- **Claim**: The GS algorithm returns a perfect matching (all doctors/hospitals are matched)

**Proof by Contradiction:**

- Suppose some $h$ is unmatched at the end.
- $\implies$ there is some $d$ that is unmatched
- B/c the alg terminated, $h$ has made an offer to $d$
  - $d$ accepted
    - $d$ was matched and stays matched $\therefore$ contradiction
  - $d$ rejected
    - $d$ was matched and stays matched $\therefore$ contradiction
GS Algorithm: Stable Matching

- **Stability**: GS algorithm outputs a stable matching.
- Proof by contradiction:
  - Suppose there is an instability $d, d', h, h'$
    - $d \rightarrow h'$ ($d, h') \in M$
    - $d : h > h'$
    - $d' \not\rightarrow h$ ($d', h) \in M$
    - $h : d > d'$

  We'll derive the contradiction $d : h' > h$

  - Because $h$ prefers $d$, $h$ made an offer to $d$ before $d'$
  - Case 1: $d$ accepted
    - Case 2: $d$ rejected
GS Algorithm: Stable Matching

- **Stability**: GS algorithm outputs a stable matching

- Proof by contradiction:
  - Suppose there is an instability $d, d', h, h'$

```
\begin{array}{c}
  d & \overset{d}{\longrightarrow} & h' \\
  d' & \overset{d'}{\longrightarrow} & h
\end{array}
```

- $(d, h') \in M$
- $(d', h) \in M$

Case 1: $d$ accepted

- At some point $d$ broke off the match with $h$
- Because "doctors go up" $d: h' > h$
GS Algorithm: Stable Matching

- **Stability:** GS algorithm outputs a stable matching

- **Proof by contradiction:**
  - Suppose there is an instability $d, d', h, h'$

  
  
  \[
  \begin{align*}
  d & \rightarrow h' \quad (d, h') \in M \\
  d' & \rightarrow h \quad (d', h) \in M \\
  \end{align*}
  \]

  - Case 2: $d$ rejected
    - The $d$ was matched to some $h''$ s.t. $d : h'' > h$
    - Because "doctors go up"

  Contradiction.
GS Algorithm: Running Time

- Let M be empty
- While (some hospital h is unmatched):
  - If (h has offered a job to everyone): break
  - Else: let d be the highest-ranked doctor to which h has not yet offered a job
  - h makes an offer to d:
    - If (d is unmatched):
      - d accepts, add (d,h) to M
    - ElseIf (d is matched to h' & d: h' > h):
      - d rejects, do nothing
    - ElseIf (d is matched to h' & d: h > h'):
      - d accepts, remove (d,h') from M and add (d,h) to M
- Output M
GS Algorithm: Running Time

• **Running Time:**
  - A straightforward implementation requires \( \approx n^3 \) operations, \( \approx n^2 \) space.
GS Algorithm: Running Time

• **Running Time:**
  • A careful implementation requires just \( \approx n^2 \) time and \( \approx n^2 \) space
GS Algorithm: Running Time

• **Running Time:**
  • A careful implementation requires just \( \approx n^2 \) time and \( \approx n^2 \) space

---

<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>Alice</td>
<td>CH</td>
<td>MGH</td>
<td>BW</td>
<td>MTA</td>
<td>BID</td>
</tr>
<tr>
<td>Bob</td>
<td>BID</td>
<td>BW</td>
<td>MTA</td>
<td>MGH</td>
<td>CH</td>
</tr>
<tr>
<td>Clara</td>
<td>BW</td>
<td>BID</td>
<td>MTA</td>
<td>CH</td>
<td>MGH</td>
</tr>
<tr>
<td>Dorit</td>
<td>MGH</td>
<td>CH</td>
<td>MTA</td>
<td>BID</td>
<td>BW</td>
</tr>
<tr>
<td>Ernie</td>
<td>MTA</td>
<td>BW</td>
<td>CH</td>
<td>BID</td>
<td>MGH</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th></th>
<th>MGH</th>
<th>BW</th>
<th>BID</th>
<th>MTA</th>
<th>CH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>2\text{nd}</td>
<td>3\text{rd}</td>
<td>5\text{th}</td>
<td>4\text{th}</td>
<td>1\text{st}</td>
</tr>
<tr>
<td>Bob</td>
<td>4\text{th}</td>
<td>2\text{nd}</td>
<td>1\text{st}</td>
<td>3\text{rd}</td>
<td>5\text{th}</td>
</tr>
<tr>
<td>Clara</td>
<td>5\text{th}</td>
<td>1\text{st}</td>
<td>2\text{nd}</td>
<td>3\text{rd}</td>
<td>4\text{th}</td>
</tr>
<tr>
<td>Dorit</td>
<td>1\text{st}</td>
<td>5\text{th}</td>
<td>4\text{th}</td>
<td>3\text{rd}</td>
<td>2\text{nd}</td>
</tr>
<tr>
<td>Ernie</td>
<td>5\text{th}</td>
<td>2\text{nd}</td>
<td>4\text{th}</td>
<td>1\text{st}</td>
<td>3\text{rd}</td>
</tr>
</tbody>
</table>

Can convert from doc x rank \( \Rightarrow \) doc x hosp in \( n^2 \) ops
GS Algorithm: Running Time

• **Running Time:**
  
  • A careful implementation requires just \( \approx n^2 \) time and \( \approx n^2 \) space

\begin{align*}
1 & \text{ Convert the doctors' preferences } n^2 \text{ ops} \\
2 & \text{ Run GS (} n^2 \text{ offers} \times 1 \text{ operation} \text{)} \quad n^2 \text{ ops} \\
& \approx n^2 \text{ operations}
\end{align*}
# Real World Impact

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>STABLE AND UNSTABLE (CENTRALIZED) MECHANISMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Market</td>
</tr>
<tr>
<td></td>
<td>American medical markets</td>
</tr>
<tr>
<td></td>
<td>NRMP</td>
</tr>
<tr>
<td></td>
<td>Medical Specialties</td>
</tr>
<tr>
<td></td>
<td>British Regional Medical Markets</td>
</tr>
<tr>
<td></td>
<td>Edinburgh ('69)</td>
</tr>
<tr>
<td></td>
<td>Cardiff</td>
</tr>
<tr>
<td></td>
<td>Birmingham</td>
</tr>
<tr>
<td></td>
<td>Edinburgh ('67)</td>
</tr>
<tr>
<td></td>
<td>Newcastle</td>
</tr>
<tr>
<td></td>
<td>Sheffield</td>
</tr>
<tr>
<td></td>
<td>Cambridge</td>
</tr>
<tr>
<td></td>
<td>London Hospital</td>
</tr>
<tr>
<td></td>
<td>Other healthcare markets</td>
</tr>
<tr>
<td></td>
<td>Dental Residencies</td>
</tr>
<tr>
<td></td>
<td>Osteopaths (&lt;‘94)</td>
</tr>
<tr>
<td></td>
<td>Osteopaths (≥‘94)</td>
</tr>
<tr>
<td></td>
<td>Pharmacists</td>
</tr>
<tr>
<td></td>
<td>Other markets and matching processes</td>
</tr>
<tr>
<td></td>
<td>Canadian Lawyers</td>
</tr>
<tr>
<td></td>
<td>Sororities</td>
</tr>
</tbody>
</table>

Table 1. Reproduced from Roth (2002, Table 1).
Real World Impact

- **Doctors ↔ Hospitals**
  - Have to deal with two-body problems
  - Have to make sure doctors do not game the system

- **Kidneys ↔ Patients**
  - Not all matches are feasible (blood types)
  - Certain pairs must be matched

- **Students ↔ Public Schools**
  - Siblings, walking zones, diversity

- **Reform Rabbis ↔ Synagogues**
  - No idea, just a fun example

---

2012 Nobel Prize to Lloyd Shapley and Al Roth!