Incentives in Computer Science

Stable matching
HW:

"since telling the truth is a dominant strategy"

≡ "since the mechanism is truthful"

If a mechanism is [truthful], it is never beneficial to lie, no matter what others do.

Is every truthful mechanism Pareto-optimal?

No, ignore preferences
Today – stable matching

• “The Prize concerns a central economic problem: how to match different agents as well as possible. For example, students have to be matched with schools, and donors of human organs with patients in need of a transplant. How can such matching be accomplished as efficiently as possible? What methods are beneficial to what groups? The prize rewards two scholars who answered these questions on a journey from abstract theory on stable allocations to practical design of market institutions.”
Matching Residents to Hospitals

• Given n hospitals (each with 1 open slot for a resident) and n applicants for a residency, find a "suitable" matching.
  – Each hospital ranks applicants in order of preference from best to worst.
  – Each applicant/resident ranks hospitals in order of preference from best to worst.

```
  hosp  residents
  1:2 A                      1:2 B
     B
```

2-sided matching
Matching Residents to Hospitals

• **Goal.** Given a set of preferences among hospitals and residents looking for a residency, design a good admissions/matching process.

• **Unstable pair:** resident $x$ and hospital $y$ are unstable if:
  - $x$ prefers $y$ to its assigned hospital.
  - $y$ prefers $x$ to one of its admitted students.

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

- **Unstable pair:** resident/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 assignment.** Assignment with no unstable pairs.

**Hospital’s Preference Profile**

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

**Resident’s Preference Profile**

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

$x$ prefers $A$ to $B$, $y$ prefers $Y$ to $Z$, $Z$ prefers $C$ to $X$. 
Stable Matching Problem

- **Perfect matching**: 1-1 matching; everyone matched.
  - Each hospital gets exactly one resident.
  - Each resident is assigned to exactly one hospital.

- **Stability**: no incentive for some pair of participants to undermine assignment by joint action.
  - In matching M, an unmatched pair h-r is unstable if hospital h and applicant r each prefer each other to current matches.
  - Unstable pair h-r could each improve by making a side deal.

- **Stable matching**: perfect matching with no unstable pairs.

- **Stable matching problem**: Given the preference lists of n hospitals and n applicants, find a stable matching if one exists.
Apologies in advance

• Note: I might interchangeably use the terms residents or applicants. In both cases, I mean medical school graduates seeking a residency.

• I may accidentally say “men” for hospitals and “women” for applicants.

• This is because, for many years, when presenting this material, people spoke of “stable marriage” and used men and women as the two sets.

• In that context, you can think of the problem as studying 1950’s dating.
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.

<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>

<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>

8
DA  Deferred Acceptance Algorithm
GS  Gale-Shapley Algorithm [1962]

Initialize all hospitals and residents to be unmatched

while (some hospital unmatched and hasn’t made an offer to every resident)
{
    Choose such a hospital h
    r = 1st applicant on h's list to whom h has not made an offer
    if (r is unmatched)
        tentatively match h and r. (h “proposes” to r.)
    else if (r prefers h to her tentative match h’)
        tentatively match h and r, and set h' to be unmatched
    else
        r rejects h (and h remains unmatched)
}
Initialize all hospitals and residents to be unmatched

while (some hospital unmatched and hasn’t made an offer to every resident) {
    Choose such a hospital $h$
    $r = 1^{st}$ applicant on $h$'s list to whom $h$ has not made an offer
    if (r is unmatched)
        tentatively match $h$ and $r$. (h “proposes” to r.)
    else if (r prefers $h$ to her tentative match $h'$)
        tentatively match $h$ and $r$, and set $h'$ to be unmatched
    else
        $r$ rejects $h$ (and $h$ remains unmatched)
}
Observations:

- Hospitals make offers to residents in order by preference.
- Once a resident is matched, she stays until termination of alg & her successive matches are better & better from her perspective.
- Alg terminates after at most $n^2$ iterations thru while loop.
All hospitals/residents are matched in end (perfect matching)

Pf

Say \( h \) unmatched at end.
Then \( h \) proposed to all residents.
if \( h \) unmatched at end, then
that is unmatched at end.
\[ \rightarrow \leftarrow \]
Thm: The final matching is stable.

Pf: 

by \Rightarrow

Suppose end up with unstable pair \((h, r')\).

Case 1: \(h\) never proposed to \(r'\). \(\Rightarrow\) \(h\) prefers \(r\) to \(r'\) because \(r\) proposes \(h\) in order of preference.

Case 2: \(h\) did propose to \(r'\). \(\Rightarrow\) \(r'\) prefers \(h'\) to \(h\) because \(h'\) got \(h\) for better matches & better hospital.
Summary

• Stable matching problem. Given n hospitals and n residents, and their preferences, find a stable matching if one exists.

• Gale-Shapley (GS) algorithm (also called “Deferred Acceptance” (DA) algorithm). Guaranteed to find a stable matching for any problem instance.

• Algorithm underspecified. Q. If there are multiple stable matchings, which one does GS find?
• Algorithm is under-specified.
• 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?
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. Hospital h is an attainable match of resident r if there exists some stable matching in which they are matched.
Understanding the Solution

• Def. Hospital $h$ is an **attainable match** of resident $r$ if there exists some stable matching in which they are matched.

• **Hospital-optimal assignment.** Each hospital receives **best attainable match**.

• **Claim.** All executions of GS yield hospital-optimal assignment, which is a stable matching!
  – No reason a priori to believe that hospital-optimal assignment is perfect, let alone stable.
  – Simultaneously best for each and every hospital.
Hospital Optimality

• Claim. GS matching is hospital-optimal.

• Pf. (by contradiction)
  In execution of GS, consider first time some hospital $h$ is rejected by its best attainable match $r$.
  
  First rejection by best attainable $h$ rejects $h$ for $h'$.
  
  $h'$ prefers $h$ to $h$.
  
  Claim: $h'$ prefers $r$ to $r'$.
  
  $h'$ has not yet been rejected by best attainable best attainable is $r$ or below $r$ and $r'$ is attainable $\Rightarrow h'$ prefers $r$ to $r'$. 
Stable Matching Summary

• **Stable matching problem.** Given preference profiles of $n$ hospitals and $n$ residents, find a stable matching.

  - no unmatched hospital and resident prefer to be matched to each other

• **Gale-Shapley algorithm.** Finds a stable matching in $O(n^2)$ time.

• **Hospital-optimality.** In version of GS where hospitals make offers, each hospital receives best attainable match.

  - a is an attainable match of $h$ if there exist some stable matching where they are matched

• Q. What about the residents/applicants?
Resident Pessimality

- Resident-pessimal assignment. Each resident receives worst attainable match.

Proof:
- Output of GS & h is not r's worst attainable
- h' is r's worst attainable
- There exists a matching M" in which h' & r" are matched.
- h' & r" are matched by hospital optimally.
- h likes r more than r" and r likes h more than h' because h' is r's worst attainable match.
Honesty

• Are the participants in a stable matching algorithm motivated to report their preferences truthfully?
Honesty for residents in hospital-proposing version

<table>
<thead>
<tr>
<th>Hospitals preferences</th>
<th>1&lt;sup&gt;st&lt;/sup&gt;</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt;</th>
<th>3&lt;sup&gt;rd&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>C</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Y</td>
<td>A</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>Z</td>
<td>C</td>
<td>A</td>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Residents preferences</th>
<th>1&lt;sup&gt;st&lt;/sup&gt;</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt;</th>
<th>3&lt;sup&gt;rd&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td>X</td>
<td>Z</td>
</tr>
</tbody>
</table>

**C will end up with a better match from perspective of true preferences.**

not truthful for non-proposing side if C reports YZX

if C reports YZX
**Thm:** The GS alg is truthful for proposing side.

**Lemma:** Suppose \( \mu \) is hosp-opt stable matching.

Let \( \nu \) be any other matching.

Let \( S \) be hospitals that prefer their match in \( \nu \) to their match in \( \mu \).

\[ f(h, r) \] that are unstable in \( \nu \) s.t. \( h \not \in S \).

**Proof:**

Case 1: \( \mu(S) \neq \nu(S) \)

Claim: \((h, r)\) is unstable for \( \nu \).

- Since \( h \not \in S \), \( h \) doesn't like \( r' \) as much as \( r \).
- \( r \) prefers \( h \) to \( h' \).
- \( h' \) proposed to \( r \) before \( h' \) proposed to \( \mu(h') \).
- \( \& \) was rejected by \( r \).
Case 2: \( M(S) = V(S) = R_o \)

During GS execution, each \( r \in R_o \) received \& rejected a proposal from her match in \( V \).
Let \( r \) be last one in \( R_o \) to receive a proposal during GS (from some hospital, say \( h' \)).

Claim: at that pt, \( r \) was tentatively matched to \( h \) who she rejected for \( h' \).
\( h \) must be outside \( S \).

\((h, r)\) is unstable for \( V \)
\( h \) likes \( r \) at least as much as \( \mu(h) = r' \) likes \( r' \) at least as much as \( V(r) \)
\( r \) likes \( h \) at least as much as \( V(r) \).

because \( V(r) \) proposed to \( r \) before \( h \) did which was before \( h' \) did. \( \square \)