9. Stable matching

Design and Analysis of Algorithms

Elias Koutsoupias
(borrowed heavily from Kevin Wayne's presentation)

Hilary Term 2022
Stable matching [KT 1.1]

- Stable matching is a simple game-theoretic algorithmic problem
- Multiple applications
- Nobel Prize to Lloyd Shapley and Alvin Roth, 2012
Goal. Given a set of preferences among hospitals and med-school students, design a self-reinforcing admissions process.

Unstable pair. Hospital $h$ and student $s$ form an unstable pair if both:

- $h$ prefers $s$ to one of its admitted students
- $s$ prefers $h$ to assigned hospital.

Stable assignment. Assignment with no unstable pairs.

- Natural and desirable condition.
- Individual self-interest prevents any hospital–student side deal.
Stable matching problem: input

**Input:** A set of $n$ hospitals $H$ and a set of $n$ students $S$.

- Each hospital $h \in H$ ranks students
- Each student $s \in S$ ranks hospitals

More concretely, the input is a set of $2n$ permutations of $(1, \ldots, n)$. 
Example: input

\(n = 3\)

Hospitals = \{A, B, C\}

Students = \{X, Y, Z\}

Preferences of hospitals (left) and students (right)
A matching $M$ is a set of ordered pairs $h - s$ with $h \in H$ and $s \in S$, such that

- Each hospital $h \in H$ appears in at most one pair of $M$
- Each student $s \in S$ appears in at most one pair of $M$

A matching $M$ is perfect if every member of $H$ (and $S$) is matched, i.e., appears in $M$. 

**Definition**

Perfect matching
Example: perfect matching

\( n = 3 \)

Hospitals = \( \{A, B, C\} \)

Students = \( \{X, Y, Z\} \)

Preferences of hospitals (left) and students (right)

A perfect matching: \( \{A - Z, B - Y, C - X\} \).
Unstable pair

Definition

Given a perfect matching \( M \), hospital \( h \) and student \( s \) form an unstable pair if both:

- \( h \) prefers \( s \) to matched student
- \( s \) prefers \( h \) to matched hospital.

An unstable pair \( h–s \) could each improve by joint action.
Example: unstable pair

\[ n = 3 \]

Hospitals = \{ A, B, C \}

Students = \{ X, Y, Z \}

Preferences of hospitals (left) and students (right)

\begin{array}{c|c|c|c}
 & 1 & 2 & 3 \\
\hline
A & X & Y & Z \\
B & Y & X & Z \\
C & X & Y & Z \\
\end{array}

\begin{array}{c|c|c|c}
 & 1 & 2 & 3 \\
\hline
X & B & A & C \\
Y & A & B & C \\
Z & A & B & C \\
\end{array}

A – Y is an unstable pair.
Stable matching problem

Definition (Stable matching)
A **stable matching** is a perfect matching with no unstable pairs.

Stable matching problem.

Definition (Stable matching problem)
Given the preference lists of $n$ hospitals and $n$ students, find a stable matching (if one exists).

A stable matching \{A – X, B – Y, C – Z\}. 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>B</td>
<td>Y</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Y</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
Stable roommate problem

- Do stable matchings always exist?
- 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 showing roommate preferences]

No perfect matching is stable
Stable roommate problem

No perfect roommate matching is stable

\[ A - B, \ C - D \Rightarrow B - C \ \text{is unstable} \]
\[ A - C, \ B - D \Rightarrow A - B \ \text{is unstable} \]
\[ A - D, \ B - C \Rightarrow A - C \ \text{is unstable} \]

Therefore, stable roommate matchings may not exist.
Gale-Shapley deferred acceptance algorithm

A natural algorithm that guarantees to find a stable matching.

\[\text{Gale-Shapley}(\text{lists of preferences})\]

1. \( M \leftarrow \emptyset \)
2. \( \text{while (some hospital } h \text{ is unmatched and hasn’t proposed to every student)} \)
3. \( s \leftarrow \text{first student on } h\text{’s list to whom } h \text{ has not yet proposed} \)
4. \( \text{if (} s \text{ is unmatched)} \)
5. \( \quad \text{then Add } h-s \text{ to matching } M \)
6. \( \quad \text{else if (} s \text{ prefers } h \text{ to current partner } h' \) \)
7. \( \quad \quad \text{then Replace } h'-s \text{ with } h-s \text{ in matching } M \)
8. \( \quad \quad \text{else } s \text{ rejects } h \)
9. \( \text{return stable matching } M. \)
Running time

- Hospitals propose to students in decreasing order of preference
- Once a student is matched, the student never becomes unmatched; only “trades up.”

**Lemma**

*Algorithm terminates after at most $n^2$ iterations of While loop.*

**Proof.**

Each time through the While loop, a hospital proposes to a new student. Thus, there are at most $n^2$ possible proposals. □
**Examples with** $n(n - 1) + 1$ **steps**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>Y</td>
<td>W</td>
<td>Z</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td>W</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td>D</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>B</td>
<td>C</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>X</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Y</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>
Proof of correctness

Lemma

The Gale-Shapley algorithm finds a perfect matching.

Proof.

- Suppose, for sake of contradiction, that some hospital $h$ is unmatched upon termination of the Gale–Shapley algorithm.
- Then some student, say $s$, is unmatched upon termination.
- So $s$ was never proposed to, because once proposed it becomes matched and remains matched thereafter.
- But, $h$ proposes to every student, since $h$ ends up unmatched.
Proof of correctness

Lemma

The matching $M$ returned by the Gale-Shapley algorithm is stable.

Proof.

Consider any pair $h$–$s$ that is not in $M$. We show that it is not unstable.

$h$ never proposed to $s$: Therefore, $h$ prefers its student in $M$ to $s$.

$h$ proposed to $s$: Therefore $s$ rejected $h$ at some point, which means that $s$ ended up with a more preferred hospital.

Theorem (Gale–Shapley 1962)

The Gale–Shapley algorithm guarantees to find a stable matching for any problem instance.
Multiple stable matchings

An instance may have multiple stable matchings. For example:
\{A \rightarrow X, B \rightarrow Y, C \rightarrow Z\} and \{A \rightarrow Y, B \rightarrow X, C \rightarrow Z\}
A student $s$ is a **valid partner** for hospital $h$ if there exists any stable matching in which $h$ and $s$ are matched.

For example

- Both $X$ and $Y$ are valid partners for $A$.
- Both $X$ and $Y$ are valid partners for $B$.
- $Z$ is the only valid partner for $C$.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>Y</td>
<td>X</td>
<td>Z</td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X</strong></td>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td><strong>Y</strong></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
Which stable matching?

**Hospital-optimal assignment:** Each hospital receives best valid partner.

- Is it a perfect matching?
- Is it stable?

**Lemma**

*The Gale-Shapley algorithm returns the hospital-optimal assignment.*

As a corollary, we get that the hospital-optimal assignment is stable.
Which stable matching?

**Student-pessimal assignment:** Each student receives worst valid partner.

**Lemma**

*The Gale-Shapley algorithm returns the student-pessimal assignment.*

As a corollary, we get that the student-pessimal assignment is stable.
Is the Gale-Shapley algorithm truthful? That is, can participants gain by misrepresenting their preferences?

- A hospital cannot get a better solution by lying about their preference
- But a student may gain by lying about their preferences