Lecture 2: The medical match

Thilo Klein

ZEW Mannheim

Introduction

One of the earliest (and most succesful) use of matching theory for real-life problem is the matching of medical residents to US hospitals.

- Upon completing their degrees medical school students must spend some time at a hospital as residents.

  (An intern is a first-year resident.)

- Today, in the US the match between students and hospitals involve about:
  - 20,000+ candidates
  - 3,800 residency programs.

History

For the first half of the 20th century, the matching was decentralized:

- Candidates had to apply separately for positions.
- Hospitals were deciding themselves who to hire.

Competition between hospitals yield to unravelling: candidates hired several years before graduation.

Problems:

- less incentives to study hard $\rightarrow$ mismatch.
- Students not choosing the specialty that they would eventually prefer.
- Hospitals would forgo better match.

History

For the first half of the 20th century, the matching was decentralized:

- Candidates had to apply separately for positions.
- Hospitals were deciding themselves who to hire.

Competition between hospitals yield to unravelling: candidates hired several years before graduation.

Problems:

- less incentives to study hard $\to$ mismatch.
- Students not choosing the specialty that they would eventually prefer.
- Hospitals would forgo better match.

History

For the first half of the 20th century, the matching was decentralized:

- Candidates had to apply separately for positions.
- Hospitals were deciding themselves who to hire.

Competition between hospitals yield to unravelling: candidates hired several years before graduation.

Problems:

- less incentives to study hard $\rightarrow$ mismatch.
- Students not choosing the specialty that they would eventually prefer.
- Hospitals would forgo better match.

In 1945 the American medical schools agreed not to disclose information about students before a certain date.

But this created bottleneck: less time to match.

In real-life, matching can be a slow process:

- It takes time to reach a candidate (to make her an offer).
- Students wait before accepting an offer (a better offer can arrive tomorrow!)

As a result

- Pessimistic students would accept "bad" offer (too risky to say no).
- Optimistic students would end up with "bad" match (or not match at all).

$\Rightarrow$ mismatch (once again).

In 1945 the American medical schools agreed not to disclose information about students before a certain date.

But this created bottleneck: less time to match.

In real-life, matching can be a slow process:

- It takes time to reach a candidate (to make her an offer).
- Students wait before accepting an offer (a better offer can arrive tomorrow!)

As a result

- Pessimistic students would accept "bad" offer (too risky to say no).
- Optimistic students would end up with "bad" match (or not match at all).

$\Rightarrow$ mismatch (once again).

In 1945 the American medical schools agreed not to disclose information about students before a certain date.

But this created bottleneck: less time to match.

In real-life, matching can be a slow process:

- It takes time to reach a candidate (to make her an offer).
- Students wait before accepting an offer (a better offer can arrive tomorrow!)

As a result

- Pessimistic students would accept "bad" offer (too risky to say no).
- Optimistic students would end up with "bad" match (or not match at all).

$\Rightarrow$ mismatch (once again).

In 1945 the American medical schools agreed not to disclose information about students before a certain date.

But this created bottleneck: less time to match.

In real-life, matching can be a slow process:

- It takes time to reach a candidate (to make her an offer).
- Students wait before accepting an offer (a better offer can arrive tomorrow!)

As a result

- Pessimistic students would accept "bad" offer (too risky to say no).
- Optimistic students would end up with "bad" match (or not match at all).

$\Rightarrow$ mismatch (once again).

Between 1945 and 1950 the delay given to candidates decreased:

- 10 days in 1945
- less than 12 hours in 1950.

But that did not help improving the market.

In 1952, the various American medical associations agreed to switch to a centralized matching mechanism: the National Resident Matching Program (NRMP).

1. Students and hospitals submit (simultaneously) their preferences;
2. A matching is constructed using an algorithm.
3. The matching is announced.

Between 1945 and 1950 the delay given to candidates decreased:

- 10 days in 1945
- less than 12 hours in 1950.

But that did not help improving the market.

In 1952, the various American medical associations agreed to switch to a centralized matching mechanism: the National Resident Matching Program (NRMP).

1. Students and hospitals submit (simultaneously) their preferences;
2. A matching is constructed using an algorithm.
3. The matching is announced.

Between 1945 and 1950 the delay given to candidates decreased:

- 10 days in 1945
- less than 12 hours in 1950.

But that did not help improving the market.

In 1952, the various American medical associations agreed to switch to a centralized matching mechanism: the National Resident Matching Program (NRMP).

1. Students and hospitals submit (simultaneously) their preferences;
2. A matching is constructed using an algorithm.
3. The matching is announced.

Between 1945 and 1950 the delay given to candidates decreased:

- 10 days in 1945
- less than 12 hours in 1950.

But that did not help improving the market.

In 1952, the various American medical associations agreed to switch to a centralized matching mechanism: the National Resident Matching Program (NRMP).

1. Students and hospitals submit (simultaneously) their preferences;
2. A matching is constructed using an algorithm.
3. The matching is announced.

Between 1945 and 1950 the delay given to candidates decreased:

- 10 days in 1945
- less than 12 hours in 1950.

But that did not help improving the market.

In 1952, the various American medical associations agreed to switch to a centralized matching mechanism: the National Resident Matching Program (NRMP).

1. Students and hospitals submit (simultaneously) their preferences;
2. A matching is constructed using an algorithm.
3. The matching is announced.

1984

In a now famous work, Alvin Roth studied the NRMP algorithm.

Roth showed that it is equivalent to the Deferred Acceptance algorithm!

1984

In a now famous work, Alvin Roth studied the NRMP algorithm.

Roth showed that it is equivalent to the Deferred Acceptance algorithm!

The many-to-one matching model

A medical match problem starts with

- A finite set of doctors: $D = \{d_1, d_2, \dots\}$
- A finite set of hospitals: $H = \{h_1, h_2, \dots\}$

In such problems,

- Each doctors wants to be hired by one hospital.
- Each hospital can hire several doctors.

Accordingly, for each hospital $h \in H$ there is a capacity q_h that specifies the maximum number of doctors hospital h can hire.

Preferences

- Doctor's preferences over hospitals are like in the classic one-to-one matching model:

  Each doctor $d \in D$ has a (strict) preference relation P_d of the hospitals and the option of not being hired by any hospital.

- Since hospitals can hire several doctors, each hospital $h \in H$ has a preference relation $P_h^\sharp$ over sets of doctors.

  Example:

  $$\{d_1, d_2\}\, P_h^\sharp\, \{d_3, d_4\}$$

  means that hospital h prefers to hire d_1 and d_2 to hiring d_3 and d_4.

  But we could well have $\{d_5\}P_h^\sharp\{d_1, d_2\}\ldots$

Preferences

- Doctor's preferences over hospitals are like in the classic one-to-one matching model:

  Each doctor $d \in D$ has a (strict) preference relation P_d of the hospitals and the option of not being hired by any hospital.

- Since hospitals can hire several doctors, each hospital $h \in H$ has a preference relation $P_h^\sharp$ over sets of doctors.

  Example:

  $$\{d_1, d_2\} \, P_h^\sharp \, \{d_3, d_4\}$$

  means that hospital h prefers to hire d_1 and d_2 to hiring d_3 and d_4.

  But we could well have $\{d_5\} P_h^\sharp \{d_1, d_2\} \ldots$

Preferences

- Doctor's preferences over hospitals are like in the classic one-to-one matching model:

  Each doctor $d \in D$ has a (strict) preference relation P_d of the hospitals and the option of not being hired by any hospital.

- Since hospitals can hire several doctors, each hospital $h \in H$ has a preference relation $P_h^\sharp$ over sets of doctors.

  Example:

  $$\{d_1, d_2\} \, P_h^\sharp \, \{d_3, d_4\}$$

  means that hospital h prefers to hire d_1 and d_2 to hiring d_3 and d_4.

  But we could well have $\{d_5\} P_h^\sharp \{d_1, d_2\} \ldots$

Preferences

- Doctor's preferences over hospitals are like in the classic one-to-one matching model:

  Each doctor $d \in D$ has a (strict) preference relation P_d of the hospitals and the option of not being hired by any hospital.

- Since hospitals can hire several doctors, each hospital $h \in H$ has a preference relation $P_h^\sharp$ over sets of doctors.

  Example:

  $$\{d_1, d_2\} \, P_h^\sharp \, \{d_3, d_4\}$$

  means that hospital h prefers to hire d_1 and d_2 to hiring d_3 and d_4.

  But we could well have $\{d_5\} P_h^\sharp \{d_1, d_2\} \ldots$

Preferences

- Doctor's preferences over hospitals are like in the classic one-to-one matching model:

  Each doctor $d \in D$ has a (strict) preference relation P_d of the hospitals and the option of not being hired by any hospital.

- Since hospitals can hire several doctors, each hospital $h \in H$ has a preference relation $P_h^\sharp$ over sets of doctors.

  Example:

  $$\{d_1, d_2\} \, P_h^\sharp \, \{d_3, d_4\}$$

  means that hospital h prefers to hire d_1 and d_2 to hiring d_3 and d_4.

  But we could well have $\{d_5\}P_h^\sharp\{d_1, d_2\}\ldots$

Responsive preferences

Working with preferences over sets of doctors can complicate things quite a bit.

The easiest approach consists of assuming that preferences over doctors (i.e., not sets) is enough.

$\Rightarrow$ we assume that hospitals' preferences are responsive.

So, we assume that each hospital $h \in H$ has a preference relation P_h over doctors.

The preference $P_h^\sharp$ will be (partially) deduced from P_h.

Responsive preferences

Working with preferences over sets of doctors can complicate things quite a bit.

The easiest approach consists of assuming that preferences over doctors (i.e., not sets) is enough.

$\Rightarrow$ we assume that hospitals' preferences are responsive.

So, we assume that each hospital $h \in H$ has a preference relation P_h over doctors.

The preference $P_h^\sharp$ will be (partially) deduced from P_h.

Responsive preferences

Working with preferences over sets of doctors can complicate things quite a bit.

The easiest approach consists of assuming that preferences over doctors (i.e., not sets) is enough.

$\Rightarrow$ we assume that hospitals' preferences are responsive.

So, we assume that each hospital $h \in H$ has a preference relation P_h over doctors.

The preference $P_h^\sharp$ will be (partially) deduced from P_h.

Responsive preferences

Working with preferences over sets of doctors can complicate things quite a bit.

The easiest approach consists of assuming that preferences over doctors (i.e., not sets) is enough.

$\Rightarrow$ we assume that hospitals' preferences are responsive.

So, we assume that each hospital $h \in H$ has a preference relation P_h over doctors.

The preference $P_h^\sharp$ will be (partially) deduced from P_h.

Responsive preferences

Working with preferences over sets of doctors can complicate things quite a bit.

The easiest approach consists of assuming that preferences over doctors (i.e., not sets) is enough.

$\Rightarrow$ we assume that hospitals' preferences are responsive.

So, we assume that each hospital $h \in H$ has a preference relation P_h over doctors.

The preference $P_h^\sharp$ will be (partially) deduced from P_h.

$P_h^\sharp$ is built by comparing sets of doctors that differ only by one doctor.

- Suppose that hospital h has already hired Dr. Carol and Dr. Denis and it can hire a third doctor.
- The hospital has the choice between Dr. Alice and Dr. Bob.
- The hospital should compare

$$\{\text{Alice}, \text{Carol}, \text{Denis}\} \qquad \text{and} \qquad \{\text{Bob}, \text{Carol}, \text{Denis}\}$$

The responsive preferences hypothesis implies that it is sufficient to compare Dr. Alice and Dr. Bob:

$$\begin{aligned} & \{\text{Alice}, \text{Carol}, \text{Denis}\} \; P_h^\sharp \; \{\text{Bob}, \text{Carol}, \text{Denis}\} \\ \Leftrightarrow \quad & \text{Alice} \; P_h \; \text{Bob} \end{aligned}$$

$P_h^\sharp$ is built by comparing sets of doctors that differ only by one doctor.

- Suppose that hospital h has already hired Dr. Carol and Dr. Denis and it can hire a third doctor.
- The hospital has the choice between Dr. Alice and Dr. Bob.
- The hospital should compare

$$\{\text{Alice, Carol, Denis}\} \qquad \text{and} \qquad \{\text{Bob, Carol, Denis}\}$$

The responsive preferences hypothesis implies that it is sufficient to compare Dr. Alice and Dr. Bob:

$$\begin{aligned} & \{\text{Alice, Carol, Denis}\}\ P_h^\sharp\ \{\text{Bob, Carol, Denis}\} \\ \Leftrightarrow \quad & \text{Alice}\ P_h\ \text{Bob} \end{aligned}$$

$P_h^\sharp$ is built by comparing sets of doctors that differ only by one doctor.

- Suppose that hospital h has already hired Dr. Carol and Dr. Denis and it can hire a third doctor.
- The hospital has the choice between Dr. Alice and Dr. Bob.
- The hospital should compare

$$\{\text{Alice}, \text{Carol}, \text{Denis}\} \qquad \text{and} \qquad \{\text{Bob}, \text{Carol}, \text{Denis}\}$$

The responsive preferences hypothesis implies that it is sufficient to compare Dr. Alice and Dr. Bob:

$$\begin{aligned} & \{\text{Alice}, \text{Carol}, \text{Denis}\} \; P_h^\sharp \; \{\text{Bob}, \text{Carol}, \text{Denis}\} \\ \Leftrightarrow \quad & \text{Alice} \; P_h \; \text{Bob} \end{aligned}$$

$P_h^\sharp$ is built by comparing sets of doctors that differ only by one doctor.

- Suppose that hospital h has already hired Dr. Carol and Dr. Denis and it can hire a third doctor.
- The hospital has the choice between Dr. Alice and Dr. Bob.
- The hospital should compare

$$\{\text{Alice}, \text{Carol}, \text{Denis}\} \qquad \text{and} \qquad \{\text{Bob}, \text{Carol}, \text{Denis}\}$$

The responsive preferences hypothesis implies that it is sufficient to compare Dr. Alice and Dr. Bob:

$$\begin{aligned} & \{\text{Alice}, \text{Carol}, \text{Denis}\} \; P_h^\sharp \; \{\text{Bob}, \text{Carol}, \text{Denis}\} \\ \Leftrightarrow \quad & \text{Alice} \; P_h \; \text{Bob} \end{aligned}$$

$P_h^\sharp$ is built by comparing sets of doctors that differ only by one doctor.

- Suppose that hospital h has already hired Dr. Carol and Dr. Denis and it can hire a third doctor.
- The hospital has the choice between Dr. Alice and Dr. Bob.
- The hospital should compare

$$\{\text{Alice}, \text{Carol}, \text{Denis}\} \qquad \text{and} \qquad \{\text{Bob}, \text{Carol}, \text{Denis}\}$$

The responsive preferences hypothesis implies that it is sufficient to compare Dr. Alice and Dr. Bob:

$$\begin{aligned} & \{\text{Alice}, \text{Carol}, \text{Denis}\}\ P_h^\sharp\ \{\text{Bob}, \text{Carol}, \text{Denis}\} \\ \Leftrightarrow \quad & \text{Alice}\ P_h\ \text{Bob} \end{aligned}$$

Definition

A preference $P_h^\sharp$ (over sets of doctors) is responsive if for any set S of doctors and two doctors d and d' such that

- $d \notin S$
- $d' \in S$

We have

$$S \; P_h^\sharp \; \underbrace{S \cup \{d\} \backslash \{d'\}}_{\substack{d \text{ added to } S \\ \text{and } d' \text{ withdrawn from } S}} \quad \Leftrightarrow \quad d' \; P_h \; d \; .$$

Responsive preferences: examples

Let $P_h = d_1, d_2, d_3, d_4$.

- Compare $\{d_1, , d_4\}$ and $\{d_1, , d_4\}$.

  The only difference is d_2 and d_3, so

$$\{d_1, d_2, d_4\} \; P_h^\sharp \; \{d_1, d_3, d_4\}$$

- Compare $\{d_1, d_3\}$ and $\{d_2, d_4\}$

$$\left.\begin{array}{l} \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_3\} \\ \{d_2, d_3\} \; P_h^\sharp \; \{d_2, d_4\} \end{array}\right\} \Rightarrow \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_4\}$$

Responsive preferences: examples

Let $P_h = d_1, d_2, d_3, d_4$.

- Compare $\{d_1, d_3, d_4\}$ and $\{d_1, d_2, d_4\}$.

  The only difference is d_2 and d_3, so

$$\{d_1, d_2, d_4\} \; P_h^\sharp \; \{d_1, d_3, d_4\}$$

- Compare $\{d_1, d_3\}$ and $\{d_2, d_4\}$

$$\left.\begin{array}{l} \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_3\} \\ \{d_2, d_3\} \; P_h^\sharp \; \{d_2, d_4\} \end{array}\right\} \Rightarrow \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_4\}$$

Responsive preferences: examples

Let $P_h = d_1, d_2, d_3, d_4$.

- Compare $\{d_1, d_3, d_4\}$ and $\{d_1, d_2, d_4\}$.

  The only difference is d_2 and d_3, so

$$\{d_1, d_2, d_4\}\ P_h^\sharp\ \{d_1, d_3, d_4\}$$

- Compare $\{d_1, d_3\}$ and $\{d_2, d_4\}$

$$\left.\begin{array}{l}\{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\} \\ \{d_2, d_3\}\ P_h^\sharp\ \{d_2, d_4\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_4\}$$

Responsive preferences: examples

Let $P_h = d_1, d_2, d_3, d_4$.

- Compare $\{d_1, d_3, d_4\}$ and $\{d_1, d_2, d_4\}$.

  The only difference is d_2 and d_3, so

$$\{d_1, d_2, d_4\}\ P_h^\sharp\ \{d_1, d_3, d_4\}$$

- Compare $\{d_1, d_3\}$ and $\{d_2, d_4\}$

$$\left.\begin{array}{l}\{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\}\\ \{d_2, d_3\}\ P_h^\sharp\ \{d_2, d_4\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_4\}$$

Responsive preferences: examples

Let $P_h = d_1, d_2, d_3, d_4$.

- Compare $\{d_1, d_3, d_4\}$ and $\{d_1, d_2, d_4\}$.

  The only difference is d_2 and d_3, so

$$\{d_1, d_2, d_4\} \; P_h^\sharp \; \{d_1, d_3, d_4\}$$

- Compare $\{d_1, d_3\}$ and $\{d_2, d_4\}$

$$\left.\begin{array}{l} \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_3\} \\ \{d_2, d_3\} \; P_h^\sharp \; \{d_2, d_4\} \end{array}\right\} \Rightarrow \{d_1, d_3\} \; P_h^\sharp \; \{d_2, d_4\}$$

- Compare $\{d_2\}$ and $\{d_1, d_3\}$.

  $\{d_2\}$ is the same as $\{d_2, \varnothing\}$. So we compare $\varnothing$ and d_3. If d_3 is acceptable we have

$$\left.\begin{array}{l}\{d_2, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\}\\ \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\}$$

- Compare $\{d_1, d_4\}$ and $\{d_2, d_3\}$.
  We cannot deduce which is the preferred set.
  Under responsive preferences both

$$\{d_1, d_4\}\ P_h^\sharp\ \{d_2, d_3\} \qquad \text{and} \qquad \{d_2, d_3\}\ P_h^\sharp\ \{d_1, d_4\}$$

  are possible.

- Compare $\{d_2\}$ and $\{d_1, d_3\}$.

  $\{d_2\}$ is the same as $\{d_2, \varnothing\}$. So we compare $\varnothing$ and d_3. If d_3 is acceptable we have

$$\left.\begin{array}{l}\{d_2, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\} \\ \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\}$$

- Compare $\{d_1, d_4\}$ and $\{d_2, d_3\}$.
  We cannot deduce which is the preferred set.
  Under responsive preferences both

$$\{d_1, d_4\}\ P_h^\sharp\ \{d_2, d_3\} \qquad \text{and} \qquad \{d_2, d_3\}\ P_h^\sharp\ \{d_1, d_4\}$$

  are possible.

- Compare $\{d_2\}$ and $\{d_1, d_3\}$.

  $\{d_2\}$ is the same as $\{d_2, \varnothing\}$. So we compare $\varnothing$ and d_3. If d_3 is acceptable we have

$$\left.\begin{array}{l}\{d_2, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\} \\ \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\}$$

- Compare $\{d_1, d_4\}$ and $\{d_2, d_3\}$.

  We cannot deduce which is the preferred set.
  Under responsive preferences both

$$\{d_1, d_4\}\ P_h^\sharp\ \{d_2, d_3\} \qquad \text{and} \qquad \{d_2, d_3\}\ P_h^\sharp\ \{d_1, d_4\}$$

  are possible.

- Compare $\{d_2\}$ and $\{d_1, d_3\}$.

  $\{d_2\}$ is the same as $\{d_2, \varnothing\}$. So we compare $\varnothing$ and d_3. If d_3 is acceptable we have

$$\left.\begin{array}{l}\{d_2, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\} \\ \{d_1, d_3\}\ P_h^\sharp\ \{d_2, d_3\}\end{array}\right\} \Rightarrow \{d_1, d_3\}\ P_h^\sharp\ \{d_2, \varnothing\}$$

- Compare $\{d_1, d_4\}$ and $\{d_2, d_3\}$.
  We cannot deduce which is the preferred set.
  Under responsive preferences both

$$\{d_1, d_4\}\ P_h^\sharp\ \{d_2, d_3\} \qquad \text{and} \qquad \{d_2, d_3\}\ P_h^\sharp\ \{d_1, d_4\}$$

  are possible.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \to H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \to H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \to H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \rightarrow H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \to H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Matching

A matching is similar to the stability defined for one-to-one matching models, but there are a few changes:

- Hospitals can be matched with more than one doctor.
- Hospitals have a maximum capacity.

Definition

A matching is a function $\mu : H \cup D \to H \cup D$ such that:

- For each doctor $d \in D$, $\mu(d) \in H \cup \{d\}$

  A doctor is matched to one hospital or herself.

- For each hospital $h \in H$,
  - $|\mu(h)| \leq q_h$
  - If $|\mu(h)| \geq 1$ then $\mu(h) \in D$.

  A hospital's match cannot exceed its capacity and a hospital is matched to doctors.

- $\mu(d) = h$ if, and only if $d \in \mu(h)$.

Stability

In a many-to-one matching problem conjunction of three requirements: *individual rationality*, *absence of blocking pairs* and *non-wastefulness*.

Definition

A matching μ is individually rational if

- for each doctor $d \in D$, $\mu(d)\ R_d\ d$;
- for each hospital $h \in H$, there is no doctor $d \in D$ such that $\varnothing\ P_h\ d$

Stability

In a many-to-one matching problem conjunction of three requirements: *individual rationality*, *absence of blocking pairs* and *non-wastefulness*.

Definition

A matching μ is individually rational if

- for each doctor $d \in D$, $\mu(d)\ R_d\ d$;
- for each hospital $h \in H$, there is no doctor $d \in D$ such that $\varnothing\ P_h\ d$

Definition

A pair (d, h) block a matching μ if

- $\mu(d) \neq h$
- $h \; P_d \; \mu(d)$
- $d \; P_h \; d'$ for some doctor $d' \in \mu(h)$.

With responsive preferences this is the same as

$$\mu(h) \cup \{d\} \backslash \{d'\} \; P_h^\sharp \; \mu(h)$$

Definition

A matching μ is non-wasteful if

$$h \; P_d \; \mu(d) \qquad \Rightarrow \qquad |\mu(h)| = q_h$$

If d prefers a hospital to her match then that hospital has filled its capacity.

Definition

A pair (d, h) block a matching μ if

- $\mu(d) \neq h$
- $h \; P_d \; \mu(d)$
- $d \; P_h \; d'$ *for some doctor* $d' \in \mu(h)$.

  With responsive preferences this is the same as

$$\mu(h) \cup \{d\} \backslash \{d'\} \; P_h^\sharp \; \mu(h)$$

Definition

A matching μ is non-wasteful if

$$h \; P_d \; \mu(d) \qquad \Rightarrow \qquad |\mu(h)| = q_h$$

If d prefers a hospital to her match then that hospital has filled its capacity.

Definition

A pair (d, h) block a matching μ if

- $\mu(d) \neq h$
- $h \; P_d \; \mu(d)$
- $d \; P_h \; d'$ *for some doctor* $d' \in \mu(h)$.

  With responsive preferences this is the same as

$$\mu(h) \cup \{d\} \setminus \{d'\} \; P_h^\sharp \; \mu(h)$$

Definition

A matching μ is non-wasteful if

$$h \; P_d \; \mu(d) \qquad \Rightarrow \qquad |\mu(h)| = q_h$$

If d prefers a hospital to her match then that hospital has filled its capacity.

Definition

A pair (d, h) block a matching μ if

- $\mu(d) \neq h$
- $h \; P_d \; \mu(d)$
- $d \; P_h \; d'$ *for some doctor* $d' \in \mu(h)$.

  With responsive preferences this is the same as

$$\mu(h) \cup \{d\} \backslash \{d'\} \; P_h^\sharp \; \mu(h)$$

Definition

A matching μ is non-wasteful if

$$h \; P_d \; \mu(d) \qquad \Rightarrow \qquad |\mu(h)| = q_h$$

If d prefers a hospital to her match then that hospital has filled its capacity.

Definition

A pair (d, h) block a matching μ if

- $\mu(d) \neq h$
- $h \; P_d \; \mu(d)$
- $d \; P_h \; d'$ *for some doctor* $d' \in \mu(h)$.

  With responsive preferences this is the same as

$$\mu(h) \cup \{d\} \backslash \{d'\} \; P_h^\sharp \; \mu(h)$$

Definition

A matching μ is non-wasteful if

$$h \; P_d \; \mu(d) \qquad \Rightarrow \qquad |\mu(h)| = q_h$$

If d prefers a hospital to her match then that hospital has filled its capacity.

Definition

A matching μ is stable if

- it is individually rational;
- there is no pair man-woman that blocks μ;
- it is non-wasteful.

Example

Hospital h_1 has a capacity of 2, $q_{h_1} = 2$ and hospital h_2 has a capacity of 1, $q_{h_2} = 1$.

P_{d_1}	P_{d_2}	P_{d_3}
h_1	h_1	h_1
h_2	h_2	h_2

P_{h_1}	P_{h_2}
d_1	d_1
d_2	d_3
d_3	d_2

- $\mu(d_1) = h_1,\ \mu(d_2) = h_2,\ \mu(d_3) = d_3$ is wasteful.
- $\mu'(d_1) = h_1,\ \mu'(d_2) = h_2,\ \mu'(d_3) = h_1$ is blocked by d_2 and h_1.
- $\mu''(d_1) = h_1,\ \mu''(d_2) = h_1,\ \mu''(d_3) = h_2$ is stable.

Example

Hospital h_1 has a capacity of 2, $q_{h_1} = 2$ and hospital h_2 has a capacity of 1, $q_{h_2} = 1$.

P_{d_1}	P_{d_2}	P_{d_3}
h_1	h_1	h_1
h_2	h_2	h_2

P_{h_1}	P_{h_2}
d_1	d_1
d_2	d_3
d_3	d_2

- $\mu(d_1) = h_1,\ \mu(d_2) = h_2,\ \mu(d_3) = d_3$ is wasteful.
- $\mu'(d_1) = h_1,\ \mu'(d_2) = h_2,\ \mu'(d_3) = h_1$ is blocked by d_2 and h_1.
- $\mu''(d_1) = h_1,\ \mu''(d_2) = h_1,\ \mu''(d_3) = h_2$ is stable.

Example

Hospital h_1 has a capacity of 2, $q_{h_1} = 2$ and hospital h_2 has a capacity of 1, $q_{h_2} = 1$.

P_{d_1}	P_{d_2}	P_{d_3}
h_1	h_1	h_1
h_2	h_2	h_2

P_{h_1}	P_{h_2}
d_1	d_1
d_2	d_3
d_3	d_2

- $\mu(d_1) = h_1,\ \mu(d_2) = h_2,\ \mu(d_3) = d_3$ is wasteful.
- $\mu'(d_1) = h_1,\ \mu'(d_2) = h_2,\ \mu'(d_3) = h_1$ is blocked by d_2 and h_1.
- $\mu''(d_1) = h_1,\ \mu''(d_2) = h_1,\ \mu''(d_3) = h_2$ is stable.

Example

Hospital h_1 has a capacity of 2, $q_{h_1} = 2$ and hospital h_2 has a capacity of 1, $q_{h_2} = 1$.

P_{d_1}	P_{d_2}	P_{d_3}
h_1	h_1	h_1
h_2	h_2	h_2

P_{h_1}	P_{h_2}
d_1	d_1
d_2	d_3
d_3	d_2

- $\mu(d_1) = h_1,\ \mu(d_2) = h_2,\ \mu(d_3) = d_3$ is wasteful.
- $\mu'(d_1) = h_1,\ \mu'(d_2) = h_2,\ \mu'(d_3) = h_1$ is blocked by d_2 and h_1.
- $\mu''(d_1) = h_1,\ \mu''(d_2) = h_1,\ \mu''(d_3) = h_2$ is stable.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):

At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Finding stable matchings

The Deferred Acceptance algorithm can be used to obtain stable matchings.

Like for the one-to-one matching model, there are two versions:

- Doctors propose, hospitals accept and reject proposals.
- Hospitals propose, doctors accept and reject proposals.

The doctor proposing version is similar to the one-to-one model, except that now hospitals can accept many proposals at the same time (up to the capacity):
At any step of the algorithm, each hospital considers:

- The set of doctors it accepted at the previous step (if any)
- The set of doctors who just made an offer (if any)

From this set, the hospital accepts doctors up to its capacity, one at a time starting with the most preferred doctors.

Deferred Acceptance with hospital proposing

In this version of the algorithm hospitals can make several proposals at the same time.

Step 1

Each hospital proposes to its most preferred set of doctors. Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

Step $\mathbf{k}, \mathbf{k} \geq 2$
Each hospital which had one or more rejections at the previous steps proposes to its most preferred set of doctors that satisfies the following conditions:

- The set must contain all doctors the hospital proposed at an earlier step and have not rejected it.
- Any additional doctor in the set must be a doctor to whom the hospital has not proposed yet.

Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

End The algorithm stops when no hospital has an offer that is rejected.

Step $\mathbf{k}, \mathbf{k} \geq 2$
Each hospital which had one or more rejections at the previous steps proposes to its most preferred set of doctors that satisfies the following conditions:

- The set must contain all doctors the hospital proposed at an earlier step and have not rejected it.
- Any additional doctor in the set must be a doctor to whom the hospital has not proposed yet.

Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

End The algorithm stops when no hospital has an offer that is rejected.

Step $\mathbf{k}, \mathbf{k} \geq \mathbf{2}$
Each hospital which had one or more rejections at the previous steps proposes to its most preferred set of doctors that satisfies the following conditions:

- The set must contain all doctors the hospital proposed at an earlier step and have not rejected it.
- Any additional doctor in the set must be a doctor to whom the hospital has not proposed yet.

Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

End The algorithm stops when no hospital has an offer that is rejected.

Step k, $k \geq 2$
Each hospital which had one or more rejections at the previous steps proposes to its most preferred set of doctors that satisfies the following conditions:

- The set must contain all doctors the hospital proposed at an earlier step and have not rejected it.
- Any additional doctor in the set must be a doctor to whom the hospital has not proposed yet.

Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

End The algorithm stops when no hospital has an offer that is rejected.

Step $\mathbf{k}, \mathbf{k} \geq 2$
Each hospital which had one or more rejections at the previous steps proposes to its most preferred set of doctors that satisfies the following conditions:

- The set must contain all doctors the hospital proposed at an earlier step and have not rejected it.
- Any additional doctor in the set must be a doctor to whom the hospital has not proposed yet.

Each doctor rejects all but the most preferred acceptable hospital that proposed to her.

End The algorithm stops when no hospital has an offer that is rejected.

One-to-one v. many-to-one

Many results found for the one-to-one matching model carry over in the many-to-one model:

- Existence of stable matching;
- Doctor proposing DA yields the doctor-optimal matching, the most preferred stable matching for doctors (least preferred for hospitals).

  Hospital proposing DA yields the hospital-optimal matching, the most preferred stable matching for hospitals (least preferred for doctors).
- Doctor proposing DA is strategyproof for doctors.

One-to-one v. many-to-one

Many results found for the one-to-one matching model carry over in the many-to-one model:

- Existence of stable matching;
- Doctor proposing DA yields the doctor-optimal matching, the most preferred stable matching for doctors (least preferred for hospitals).

  Hospital proposing DA yields the hospital-optimal matching, the most preferred stable matching for hospitals (least preferred for doctors).
- Doctor proposing DA is strategyproof for doctors.

One-to-one v. many-to-one

Many results found for the one-to-one matching model carry over in the many-to-one model:

- Existence of stable matching;
- Doctor proposing DA yields the doctor-optimal matching, the most preferred stable matching for doctors (least preferred for hospitals).

  Hospital proposing DA yields the hospital-optimal matching, the most preferred stable matching for hospitals (least preferred for doctors).
- Doctor proposing DA is strategyproof for doctors.

One-to-one v. many-to-one

Many results found for the one-to-one matching model carry over in the many-to-one model:

- Existence of stable matching;
- Doctor proposing DA yields the doctor-optimal matching, the most preferred stable matching for doctors (least preferred for hospitals).

  Hospital proposing DA yields the hospital-optimal matching, the most preferred stable matching for hospitals (least preferred for doctors).
- Doctor proposing DA is strategyproof for doctors.

One-to-one v. many-to-one

Many results found for the one-to-one matching model carry over in the many-to-one model:

- Existence of stable matching;
- Doctor proposing DA yields the doctor-optimal matching, the most preferred stable matching for doctors (least preferred for hospitals).

  Hospital proposing DA yields the hospital-optimal matching, the most preferred stable matching for hospitals (least preferred for doctors).
- Doctor proposing DA is strategyproof for doctors.

However, the hospital proposing DA is not strategyproof for hospitals.

P_{d_1}	P_{d_2}	P_{d_3}	P_{d_4}
h_3	h_2	h_1	h_1
h_1	h_1	h_3	h_2
h_2	h_3	h_2	h_3

P_{h_1}	P_{h_2}	P_{h_3}
d_1	d_1	d_3
d_2	d_2	d_1
d_3	d_3	d_2
d_4	d_4	d_4

DA with hospital proposing yields

$$\mu_H(h_1) = \{d_3, d_4\}, \quad \mu_H(h_2) = d_2 \quad \text{and } \mu_H(h_3) = d_1$$

Consider now a deviation from hospital h_1, submitting $\widehat{P}_{h_1}$.

The deviation is profitable because it yields

$$\widehat{\mu}_H(h_1) = \{d_2, d_4\}, \quad \widehat{\mu}_H(h_2) = d_1 \quad \text{and } \widehat{\mu}_H(h_3) = d_3$$

However, the hospital proposing DA is not strategyproof for hospitals.

P_{d_1}	P_{d_2}	P_{d_3}	P_{d_4}
h_3	h_2	h_1	h_1
h_1	h_1	h_3	h_2
h_2	h_3	h_2	h_3

P_{h_1}	P_{h_2}	P_{h_3}
d_1	d_1	d_3
d_2	d_2	d_1
d_3	d_3	d_2
d_4	d_4	d_4

DA with hospital proposing yields

$$\mu_H(h_1) = \{d_3, d_4\}, \quad \mu_H(h_2) = d_2 \quad \text{and } \mu_H(h_3) = d_1$$

Consider now a deviation from hospital h_1, submitting $\widehat{P}_{h_1}$.

The deviation is profitable because it yields

$$\widehat{\mu}_H(h_1) = \{d_2, d_4\}, \quad \widehat{\mu}_H(h_2) = d_1 \quad \text{and } \widehat{\mu}_H(h_3) = d_3$$

However, the hospital proposing DA is not strategyproof for hospitals.

P_{d_1}	P_{d_2}	P_{d_3}	P_{d_4}
h_3	h_2	h_1	h_1
h_1	h_1	h_3	h_2
h_2	h_3	h_2	h_3

P_{h_1}	P_{h_2}	P_{h_3}
d_1	d_1	d_3
d_2	d_2	d_1
d_3	d_3	d_2
d_4	d_4	d_4

DA with hospital proposing yields

$$\mu_H(h_1) = \{d_3, d_4\}, \quad \mu_H(h_2) = d_2 \quad \text{and } \mu_H(h_3) = d_1$$

Consider now a deviation from hospital h_1, submitting $\widehat{P}_{h_1}$.

The deviation is profitable because it yields

$$\widehat{\mu}_H(h_1) = \{d_2, d_4\}, \quad \widehat{\mu}_H(h_2) = d_1 \quad \text{and } \widehat{\mu}_H(h_3) = d_3$$

However, the hospital proposing DA is not strategyproof for hospitals.

P_{d_1}	P_{d_2}	P_{d_3}	P_{d_4}
h_3	h_2	h_1	h_1
h_1	h_1	h_3	h_2
h_2	h_3	h_2	h_3

$\widehat{P}'_{h_1}$	P_{h_1}	P_{h_2}	P_{h_3}
d_2	d_1	d_1	d_3
d_4	d_2	d_2	d_1
d_3	d_3	d_3	d_2
d_1	d_4	d_4	d_4

DA with hospital proposing yields

$$\mu_H(h_1) = \{d_3, d_4\}, \quad \mu_H(h_2) = d_2 \quad \text{and } \mu_H(h_3) = d_1$$

Consider now a deviation from hospital h_1, submitting $\widehat{P}_{h_1}$.

The deviation is profitable because it yields

$$\widehat{\mu}_H(h_1) = \{d_2, d_4\}, \quad \widehat{\mu}_H(h_2) = d_1 \quad \text{and } \widehat{\mu}_H(h_3) = d_3$$

However, the hospital proposing DA is not strategyproof for hospitals.

P_{d_1}	P_{d_2}	P_{d_3}	P_{d_4}
h_3	h_2	h_1	h_1
h_1	h_1	h_3	h_2
h_2	h_3	h_2	h_3

$\widehat{P}'_{h_1}$	P_{h_1}	P_{h_2}	P_{h_3}
d_2	d_1	d_1	d_3
d_4	d_2	d_2	d_1
d_3	d_3	d_3	d_2
d_1	d_4	d_4	d_4

DA with hospital proposing yields

$$\mu_H(h_1) = \{d_3, d_4\}, \quad \mu_H(h_2) = d_2 \quad \text{and } \mu_H(h_3) = d_1$$

Consider now a deviation from hospital h_1, submitting $\widehat{P}_{h_1}$.

The deviation is profitable because it yields

$$\widehat{\mu}_H(h_1) = \{d_2, d_4\}, \quad \widehat{\mu}_H(h_2) = d_1 \quad \text{and } \widehat{\mu}_H(h_3) = d_3$$

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Why stability matters

The development and success of the NRMP suggests that stable matchings (through a centralized market) is paramount.

In the early 1990's Alvin Roth studied the medical market in the UK:

- Problem similar than in the US: medical graduates have to find a hospital for their residency.
- Unlike the US, the market is split in regional markets.
- Not all markets use the same procedure.

Roth found that markets designed to produce stable matchings performed relatively well, and those that do not were eventually abandoned.

Market	Use stable algorithm?	Still in use? (in 1990)
Edinburg (1969)	Yes	Yes
Cardiff	Yes	Yes
Cambridge	No	Yes
London Hospital	No	Yes
Birmingham	No	No
Edinburgh (1967)	No	No
Newcastle	No	No
Sheffield	No	No

London and Cambridge are exceptions: low markets with a strong social pressure, limiting the incentives to circumvent the matching procedure.

Unraveling in the lab

Analysis of the UK medical markets suggest that stable matching is a key property.

But it could be possible that the evolution of the UK markets is due to other, unobserved factors.

Another question is whether stability is also a factor to control unravelling.

Al Roth and John Kagel conducted a lab experiment to study the transition from a decentralized to a centralized market (that uses a stable matching mechanism).

Unraveling in the lab

Analysis of the UK medical markets suggest that stable matching is a key property.

But it could be possible that the evolution of the UK markets is due to other, unobserved factors.

Another question is whether stability is also a factor to control unravelling.

Al Roth and John Kagel conducted a lab experiment to study the transition from a decentralized to a centralized market (that uses a stable matching mechanism).

Unraveling in the lab

Analysis of the UK medical markets suggest that stable matching is a key property.

But it could be possible that the evolution of the UK markets is due to other, unobserved factors.

Another question is whether stability is also a factor to control unravelling.

Al Roth and John Kagel conducted a lab experiment to study the transition from a decentralized to a centralized market (that uses a stable matching mechanism).

Unraveling in the lab

Analysis of the UK medical markets suggest that stable matching is a key property.

But it could be possible that the evolution of the UK markets is due to other, unobserved factors.

Another question is whether stability is also a factor to control unravelling.

Al Roth and John Kagel conducted a lab experiment to study the transition from a decentralized to a centralized market (that uses a stable matching mechanism).

Unraveling in the lab

Analysis of the UK medical markets suggest that stable matching is a key property.

But it could be possible that the evolution of the UK markets is due to other, unobserved factors.

Another question is whether stability is also a factor to control unravelling.

Al Roth and John Kagel conducted a lab experiment to study the transition from a decentralized to a centralized market (that uses a stable matching mechanism).

The experiment

- Subjects split in two groups: workers and firms.
- Half of the firms & half of the workers identified as high productivity.

  The other workers and firms identified as low productivity.
- subjects would get paid according to their match:
  - *about* \$15 if matched to a high productivity partner.
  - *about* \$5 if matched to a low productivity partner
  - \$0 if not matched.

  "about": small differences introduced so that workers and firms disagree about the ranking of high and low productivity.

The experiment

- Subjects split in two groups: workers and firms.
- Half of the firms & half of the workers identified as high productivity.

  The other workers and firms identified as low productivity.
- subjects would get paid according to their match:
  - *about* \$15 if matched to a high productivity partner.
  - *about* \$5 if matched to a low productivity partner
  - \$0 if not matched.

  "about": small differences introduced so that workers and firms disagree about the ranking of high and low productivity.

The experiment

- Subjects split in two groups: workers and firms.
- Half of the firms & half of the workers identified as high productivity.

  The other workers and firms identified as low productivity.
- subjects would get paid according to their match:
  - *about* \$15 if matched to a high productivity partner.
  - *about* \$5 if matched to a low productivity partner
  - \$0 if not matched.

  "about": small differences introduced so that workers and firms disagree about the ranking of high and low productivity.

The experiment

- Subjects split in two groups: workers and firms.
- Half of the firms & half of the workers identified as high productivity.

  The other workers and firms identified as low productivity.
- subjects would get paid according to their match:
  - *about* \$15 if matched to a high productivity partner.
  - *about* \$5 if matched to a low productivity partner
  - \$0 if not matched.

  "about": small differences introduced so that workers and firms disagree about the ranking of high and low productivity.

Two designs were used:

- **Design 1**: A decentralized market run over 3 periods.
  - At each period firms can make offers to workers.
  - $2 penalty if matched in the first period.
  - $1 penalty if matched in the second period.
- **Design 2**: A centralized market, with 2 variations:
  - One variation used DA.
  - One variation used a non-stable matching algorithm.
- With this experimental design there are two sources of inefficiency:
  - Early match (unravelling).
  - Mismatch: high productivity matched with a low productivity.

  A stable matching would be assortative: high productivity always matched with high productivity.

Two designs were used:

- **Design 1**: A decentralized market run over 3 periods.
  - At each period firms can make offers to workers.
  - \$2 penalty if matched in the first period.
  - \$1 penalty if matched in the second period.
- **Design 2**: A centralized market, with 2 variations:
  - One variation used DA.
  - One variation used a non-stable matching algorithm.
- With this experimental design there are two sources of inefficiency:
  - Early match (unravelling).
  - Mismatch: high productivity matched with a low productivity.

    A stable matching would be assortative: high productivity always matched with high productivity.

Two designs were used:

- **Design 1**: A decentralized market run over 3 periods.
  - At each period firms can make offers to workers.
  - \$2 penalty if matched in the first period.
  - \$1 penalty if matched in the second period.
- **Design 2**: A centralized market, with 2 variations:
  - One variation used DA.
  - One variation used a non-stable matching algorithm.
- With this experimental design there are two sources of inefficiency:
  - Early match (unravelling).
  - Mismatch: high productivity matched with a low productivity.

  A stable matching would be assortative: high productivity always matched with high productivity.

Two designs were used:

- **Design 1**: A decentralized market run over 3 periods.
  - At each period firms can make offers to workers.
  - \$2 penalty if matched in the first period.
  - \$1 penalty if matched in the second period.
- **Design 2**: A centralized market, with 2 variations:
  - One variation used DA.
  - One variation used a non-stable matching algorithm.
- With this experimental design there are two sources of inefficiency:
  - Early match (unravelling).
  - Mismatch: high productivity matched with a low productivity.

    A stable matching would be assortative: high productivity always matched with high productivity.

Protocol

The protocol consists of:

- mimicking the US medical match before the use of a centralized mechanism
- mimicking the transition to a centralized mechanism.

More concretely:

- 10 times Design 1.
- 15 times a combination of Design 1 and Design 2:
  Design 1 for 2 periods, then Design 2 for unmatched subjects.

Protocol

The protocol consists of:

- mimicking the US medical match before the use of a centralized mechanism
- mimicking the transition to a centralized mechanism.

More concretely:

- 10 times Design 1.
- 15 times a combination of Design 1 and Design 2:
  Design 1 for 2 periods, then Design 2 for unmatched subjects.

Protocol

The protocol consists of:

- mimicking the US medical match before the use of a centralized mechanism
- mimicking the transition to a centralized mechanism.

More concretely:

- 10 times Design 1.
- 15 times a combination of Design 1 and Design 2:
  Design 1 for 2 periods, then Design 2 for unmatched subjects.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

$\Rightarrow$ DA makes the market safe for participants: no risk to delay matching decisions.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

$\Rightarrow$ DA makes the market safe for participants: no risk to delay matching decisions.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

$\Rightarrow$ DA makes the market safe for participants: no risk to delay matching decisions.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

⇒ DA makes the market safe for participants: no risk to delay matching decisions.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

$\Rightarrow$ DA makes the market safe for participants: no risk to delay matching decisions.

Results

- In the decentralized design unravelling occurs.
  When repeating the experiment, the rate of unravelling increases.
- In the centralized design with DA unravelling drastically decreases when repeating the experiment.
- In the centralized design with the non-stable algorithm unravelling increases when repeating the experiment.
- Most of the unravelling is made by high productivity subjects: higher cost of not being matched.

$\Rightarrow$ DA makes the market safe for participants: no risk to delay matching decisions.

Rural hospitals

The question of "rural hospitals" quickly arose during the development of the medical match:

candidates tend to prefer hospitals in large urban areas

$\Rightarrow$ hospitals in rural areas have a hard time filling all their openings.

Question: Can we find an algorithm/mechanism that:

- always produce stable matchings, and
- enable rural hospitals to fill all their openings?

Answer: No...

Rural hospitals

The question of "rural hospitals" quickly arose during the development of the medical match:

> candidates tend to prefer hospitals in large urban areas
>
> $\Rightarrow$ hospitals in rural areas have a hard time filling all their openings.

Question: Can we find an algorithm/mechanism that:

- always produce stable matchings, and
- enable rural hospitals to fill all their openings?

Answer: No...

Rural hospitals

The question of "rural hospitals" quickly arose during the development of the medical match:

> candidates tend to prefer hospitals in large urban areas
>
> $\Rightarrow$ hospitals in rural areas have a hard time filling all their openings.

Question: Can we find an algorithm/mechanism that:

- always produce stable matchings, and
- enable rural hospitals to fill all their openings?

Answer: No. . .

Rural hospitals

The question of "rural hospitals" quickly arose during the development of the medical match:

> candidates tend to prefer hospitals in large urban areas
>
> $\Rightarrow$ hospitals in rural areas have a hard time filling all their openings.

Question: Can we find an algorithm/mechanism that:

- always produce stable matchings, and
- enable rural hospitals to fill all their openings?

Answer: No...

Theorem (Rural Hospital Theorem)

For any preferences of doctors and hospitals, if at a stable matching a hospital does not fill all its vacancies then it does not fill all its vacancies at any *stable matching.*

Furthermore, if a hospital does not fill its vacancies at some stable matching it is matched to the same set of doctors at all *stable matchings.*

Proof

We prove the theorem when each hospital has only one vacancy.

Lemma (Decomposition lemma)

Let μ and μ' be two stable matching for the same problem.

- *$A =$ set of doctors who prefer μ' to μ*
- *$B =$ set of hospitals that prefer μ to μ'.*

Then we have:

- *Each doctor in A is matched, under both μ and μ' to a hospital in B (but not the same hospital!).*
- *Each hospital in B is matched, under both μ and μ', to a doctor in A.*

Proof

We prove the theorem when each hospital has only one vacancy.

Lemma (Decomposition lemma)

Let μ and μ' be two stable matching for the same problem.

- $A =$ *set of doctors who prefer* μ' *to* μ
- $B =$ *set of hospitals that prefer* μ *to* μ'.

Then we have:

- *Each doctor in* A *is matched, under both* μ *and* μ' *to a hospital in* B *(but not the same hospital!).*
- *Each hospital in* B *is matched, under both* μ *and* μ', *to a doctor in* A.

Proof

We prove the theorem when each hospital has only one vacancy.

Lemma (Decomposition lemma)

Let μ and μ' be two stable matching for the same problem.

- *A = set of doctors who prefer μ' to μ*
- *B = set of hospitals that prefer μ to μ'.*

Then we have:

- *Each doctor in A is matched, under both μ and μ' to a hospital in B (but not the same hospital!).*
- *Each hospital in B is matched, under both μ and μ', to a doctor in A.*

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

- $\Rightarrow h \; P_d \; d$ (if not then μ' not stable).
- $\Rightarrow$ so $d \in A$ (the set A of the lemma).
- $\Rightarrow h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

Let μ be a stable matching and d a doctor such that $\mu(d) = d$.

Let μ' be another stable matching.

Suppose there exists h such that $\mu'(d) = h$.

$\Rightarrow$ $h \; P_d \; d$ (if not then μ' not stable).

$\Rightarrow$ so $d \in A$ (the set A of the lemma).

$\Rightarrow$ $h \in B$ (if not (h, d) block μ). So $B \neq \emptyset$.

We can then invoke the Decomposition lemma, and deduce that under μ doctor d must be matched to a hospital in B!

So we cannot have $\mu(d) = d$, a contradiction.

The case of couples and the engineering method

The story of the NRMP is not exempt of issues. A major problem started in early 1970's: an increasing number of couples abstained from participating to the NRMP.

An initial fix:

- each couple designs a leading member.
- Once the leading member is matched, the preference list of the partner is edited by removing distant positions.

The problem persisted: couples were not able to submit preferences over pairs of positions.

Mid 1980's fix allowed for such preferences, but the problem persisted.

The case of couples and the engineering method

The story of the NRMP is not exempt of issues. A major problem started in early 1970's: an increasing number of couples abstained from participating to the NRMP.

An initial fix:

- each couple designs a leading member.
- Once the leading member is matched, the preference list of the partner is edited by removing distant positions.

The problem persisted: couples were not able to submit preferences over pairs of positions.

Mid 1980's fix allowed for such preferences, but the problem persisted.

The case of couples and the engineering method

The story of the NRMP is not exempt of issues. A major problem started in early 1970's: an increasing number of couples abstained from participating to the NRMP.

An initial fix:

- each couple designs a leading member.
- Once the leading member is matched, the preference list of the partner is edited by removing distant positions.

The problem persisted: couples were not able to submit preferences over pairs of positions.

Mid 1980's fix allowed for such preferences, but the problem persisted.

The case of couples and the engineering method

The story of the NRMP is not exempt of issues. A major problem started in early 1970's: an increasing number of couples abstained from participating to the NRMP.

An initial fix:

- each couple designs a leading member.
- Once the leading member is matched, the preference list of the partner is edited by removing distant positions.

The problem persisted: couples were not able to submit preferences over pairs of positions.

Mid 1980's fix allowed for such preferences, but the problem persisted.

Failure of the theory

Alice & Albert	Bill	Carol
(h_1, h_2)	h_1	h_2
(h_3, h_3)		
(not hired, h_2)		

1	1	2	cap.
h_1	h_2	h_3	
Bill	Albert	Alice	
Alice	Carol	Albert	

DA with doctors proposing:

- **1st step**: Alice& Bill $\to h_1$, Albert & Carol $\to h_2$.

  Alice Carol rejected.

- **2nd step**: Need to allow Albert to propose (with Alice) to h_3.

  But then h_2 regrets having rejected Carol.

Failure of the theory

Alice & Albert	Bill	Carol
(h_1,h_2)	h_1	h_2
(h_3,h_3)		
$(\text{not hired},h_2)$		

1	1	2	cap.
h_1	h_2	h_3	
Bill	Albert	Alice	
Alice	Carol	Albert	

DA with doctors proposing:

- **1st step**: Alice& Bill $\to h_1$, Albert & Carol $\to h_2$.

  Alice Carol rejected.
- **2nd step**: Need to allow Albert to propose (with Alice) to h_3.

  But then h_2 regrets having rejected Carol.

Failure of the theory

Alice & Albert	Bill	Carol
(h_1,h_2)	h_1	h_2
(h_3,h_3)		
(not hired,h_2)		

1	1	2	cap.
h_1	h_2	h_3	
Bill	Albert	Alice	
Alice	Carol	Albert	

DA with doctors proposing:

- **1st step**: Alice& Bill $\to$ h_1, Albert & Carol $\to$ h_2.

  Alice Carol rejected.

- **2nd step**: Need to allow Albert to propose (with Alice) to h_3.

  But then h_2 regrets having rejected Carol.

Fixing the NRMP

Roth and Peranson proposed the following solution:

- **Switch to the doctor proposing algorithm**:

  Originally NRMP was using the hospital proposing.

  Doctor proposing fairer for candidates (and increase the odds of finding optimal stable matchings).

- **Process some proposals sequentially**:

  In "classic" DA proposals are made simultaneously.

  With sequential offers, it is easier for the algorithm to detect sources of instability and correct them along the way.

Fixing the NRMP

Roth and Peranson proposed the following solution:

- **Switch to the doctor proposing algorithm**:

  Originally NRMP was using the hospital proposing.

  Doctor proposing fairer for candidates (and increase the odds of finding optimal stable matchings).

- **Process some proposals sequentially**:

  In "classic" DA proposals are made simultaneously.

  With sequential offers, it is easier for the algorithm to detect sources of instability and correct them along the way.

Fixing the NRMP

Roth and Peranson proposed the following solution:

- **Switch to the doctor proposing algorithm**:

  Originally NRMP was using the hospital proposing.

  Doctor proposing fairer for candidates (and increase the odds of finding optimal stable matchings).

- **Process some proposals sequentially**:

  In “classic” DA proposals are made simultaneously.

  With sequential offers, it is easier for the algorithm to detect sources of instability and correct them along the way.

NRMP with couples

Step 1:

Run DA with doctors proposing, excluding couples (only use single doctors' preferences).

Step 2: One by one, match couples to pairs of hospitals (in order of their preferences).

Such matches may displace single doctors matched in Step 1.

Step 3: For doctors displaced in Step 2, match them, one by one, to a hospital (in order of their preferences).

NRMP with couples

Step 1:

Run DA with doctors proposing, excluding couples (only use single doctors' preferences).

Step 2: One by one, match couples to pairs of hospitals (in order of their preferences).

Such matches may displace single doctors matched in Step 1.

Step 3: For doctors displaced in Step 2, match them, one by one, to a hospital (in order of their preferences).

NRMP with couples

Step 1:

Run DA with doctors proposing, excluding couples (only use single doctors' preferences).

Step 2: One by one, match couples to pairs of hospitals (in order of their preferences).

Such matches may displace single doctors matched in Step 1.

Step 3: For doctors displaced in Step 2, match them, one by one, to a hospital (in order of their preferences).

NRMP with couples

Step 1:

Run DA with doctors proposing, excluding couples (only use single doctors' preferences).

Step 2: One by one, match couples to pairs of hospitals (in order of their preferences).

Such matches may displace single doctors matched in Step 1.

Step 3: For doctors displaced in Step 2, match them, one by one, to a hospital (in order of their preferences).

- New algorithm first used in 1998.
- Most problems vanished and participation rate went up.
- New approach: theory has little bite. Extensive use of simulations to test various designs.

  $\Rightarrow$ Roth and Peranson worked like engineers: theory provides guidance, but experiments are run to fine tune the details.
- Theory predicts that stable matchings may not exists (when there are couples. But the preferences observed in real-life allow, in general, for the existence of stable matchings.

- New algorithm first used in 1998.
- Most problems vanished and participation rate went up.
- New approach: theory has little bite. Extensive use of simulations to test various designs.

  $\Rightarrow$ Roth and Peranson worked like engineers: theory provides guidance, but experiments are run to fine tune the details.
- Theory predicts that stable matchings may not exists (when there are couples. But the preferences observed in real-life allow, in general, for the existence of stable matchings.

- New algorithm first used in 1998.
- Most problems vanished and participation rate went up.
- New approach: theory has little bite. Extensive use of simulations to test various designs.

  $\Rightarrow$ Roth and Peranson worked like engineers: theory provides guidance, but experiments are run to fine tune the details.
- Theory predicts that stable matchings may not exists (when there are couples. But the preferences observed in real-life allow, in general, for the existence of stable matchings.

- New algorithm first used in 1998.
- Most problems vanished and participation rate went up.
- New approach: theory has little bite. Extensive use of simulations to test various designs.

  $\Rightarrow$ Roth and Peranson worked like engineers: theory provides guidance, but experiments are run to fine tune the details.
- Theory predicts that stable matchings may not exists (when there are couples. But the preferences observed in real-life allow, in general, for the existence of stable matchings.

Take-away

- The medical match is a many-to-one matching model.
- Hospitals can be matched to several doctors at once: they have preferences over sets of doctors.
- Responsive preferences assume that most of the preferences over sets of doctors can be retrieved from preferences over doctors.
- Most of the results of the one-to-one matching model carry over, except strategyproofness for DA with hospitals proposing.

- The US medical matched started as a decentralized market. Competition between hospitals led to unravelling.
- The solution was to adopt a stable matching algorithm in a centralized market.
- Analysis of the UK medical match and experiments showed that stability is a key property for the viability of a matching market: makes the market safe, thereby reducing unraveling.
- Rural hospital theorem: All stable matchings always match the same agents.
- The existence of a stable matching is not guaranteed in the presence of couples.

  When theory "fails" an engineering approach can be fruitful.