14 February 2025

Stable Matching II

Plan

* Gale-Shapley Algorithm Recap
* Stable Matching Analysis

Problem. Medical Residency Job Market.

* Doctors apply to Residency Programs.

Centralized Matching Market

↳ Doctors / Residencies submit preferences

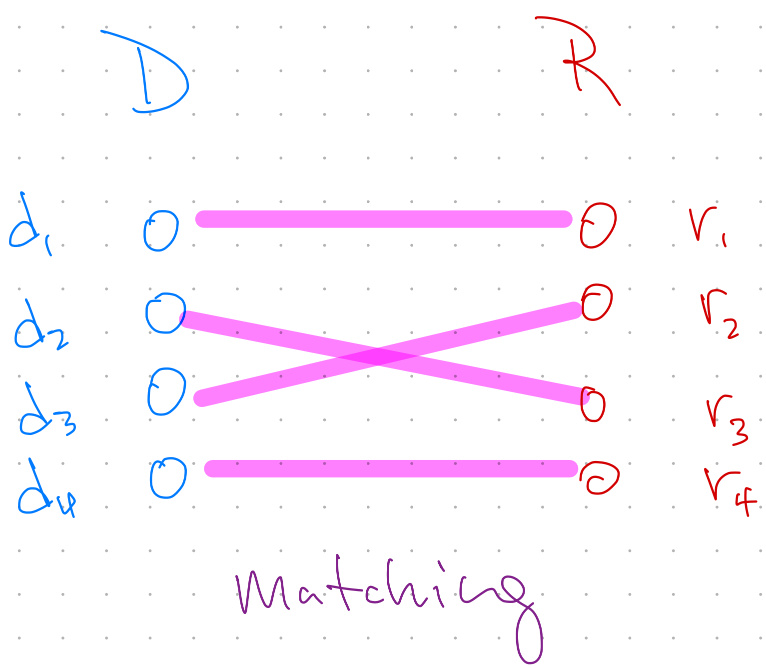
↳ Market outputs Matching

Defn. Given a collection of Doctors D, Residencies R

a matching is a set of pairs M s.t.

↳ Every pair $(d, r) \in M$ consists of exactly one $d \in D$ and one $r \in R$.

↳ Every Doctor / Residency involved in at most one pair in M.

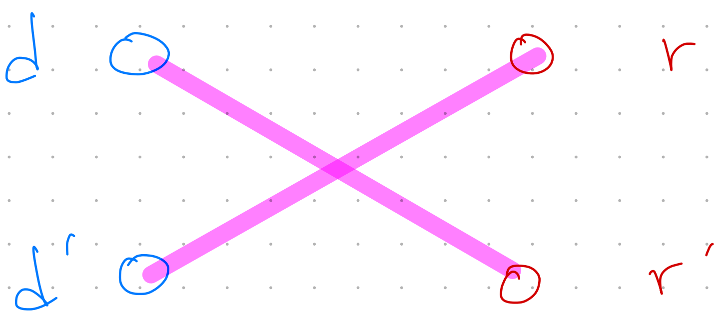


A perfect matching is a matching where every $d \in D$ and $r \in R$ is involved in exactly one pair in M.

* Suppose Market outputs (d, r') and (d', r).

Defn. Instability

d prefers r to matched residency r', AND
r prefers d to matched doctor d'.



market's matching

d and r have incentive to reject matching

Defn. A perfect matching M is stable if it contains no instabilities:

$\forall d \in D \quad \forall r \in R$

* d prefers match in M to r, OR
  ($r' \neq r$)
* r prefers match in M to d
  ($d' \neq d$)

Theorem (Gale-Shapley).

Fix a collection of Doctors D & Residencies R with any set of preferences.

There is an efficient algorithm that returns a stable perfect matching M between D & R.

Idea:

* Each residency maintains list of Doctors according to their prefences.

* Iteratively make offers to next most-preferred doctor

$$r: \underbrace{\cancel{d_5} \succ \cancel{d_3} \succ \cancel{d_7}}_{\text{already made offers}} \succ \underset{\substack{\uparrow \\ \text{next most-preferred}}}{d_1} \succ d_6 \succ d_2 \succ d_4$$

Gale - Shapley (D, R)

Initialize $M \leftarrow \emptyset$.

While $\exists$ unmatched $r \in R$ that hasn't made offers to every $d \in D$

* r makes offer to next most-preferred d

Output M

Gale-Shapley (D, R)

Initialize $M \leftarrow \emptyset$.

While $\exists$ unmatched $r \in R$ that hasn't made offers to every $d \in D$

- $*$ r makes offer to next most-preferred d

  if d is unmatched,

  ↳ $M \leftarrow M \cup \{(d,r)\}$ } doctor accepts offer

  else d is matched to some $r' \in R$.

  if d prefers r to r'

  ↳ $M \leftarrow M \setminus \{(d,r')\} \cup \{(d,r)\}$

Output M

doctor re-matches to preferred r; r' unmatched.

$d_1: r_1 \succ r_2 \succ r_3$ $\quad\quad$ $r_1: d_2 \succ d_1 \succ d_3$

$d_2: r_2 \succ r_1 \succ r_3$ $\quad\quad$ $r_2: d_3 \succ d_2 \succ d_1$

$d_3: r_1 \succ r_3 \succ r_2$ $\quad\quad$ $r_3: d_1 \succ d_2 \succ d_3$

$d_1: r_1 \succ r_2 \succ r_3$ $\quad\quad$ $r_2: d_3 \succ d_2 \succ d_1$

$d_2: r_2 \succ r_1 \succ r_3$ $\quad\quad$ $r_3: d_1 \succ d_2 \succ d_3$

$d_3: r_1 \succ r_3 \succ r_2$ $\quad\quad$ $r_1: d_2 \succ d_1 \succ d_3$

$d_1 : r_1 \succ r_2 \succ r_3$ $\qquad$ $r_1 : d_2 \succ d_1 \succ d_3$

$d_2 : r_2 \succ r_1 \succ r_3$ $\qquad$ $r_2 : d_3 \succ d_2 \succ d_1$

$d_3 : r_1 \succ r_3 \succ r_2$ $\qquad$ $r_3 : d_1 \succ d_2 \succ d_3$

$d_2 : r_2 \succ r_1 \succ r_3$ $\qquad$ $r_1 : d_2 \succ d_1 \succ d_3$

$d_3 : r_1 \succ r_3 \succ r_2$ $\qquad$ $r_2 : d_3 \succ d_2 \succ d_1$

$d_1 : r_1 \succ r_2 \succ r_3$ $\qquad$ $r_3 : d_1 \succ d_2 \succ d_3$

Some Questions

* While loop termination: Does GS always return?

* Matching: is M a perfect matching?

* Stability: is M stable?

Some Questions

* While loop termination: Does GS always return?

Some Questions

* While loop termination: Does GS always return?

unmatched

While $\exists$ $r \in R$ that hasn't made offers to every $d \in D$

Every iteration involves offer from $r \in R$ to NEW $d \in D$.

Some Questions

* While loop termination: Does GS always return?

While $\exists\ r \in R$ that hasn't made offers to every $d \in D$

Every iteration involves offer from $r \in R$ to NEW $d \in D$.

$\downarrow$

Iterations "indexed" by (r, d) pairs

$\downarrow$

$\leq |R| \cdot |D| = n^2$ iterations

Some Questions

* While loop termination: Does GS always return?

* Matching: is M a perfect matching?

* Stability: is M stable?

Key Observation (*)

Once $d \in D$ receives offer, d remains matched.

If d is re-matched from (d, r') to (d, r)

then $r \succ_d r'$.

if d is matched to some $r' \in R$.

if d prefers r to r'

$\hookrightarrow M \leftarrow M \setminus \{(d, r')\} \cup \{(d, r)\}$

re-matched

* Matching: is M a perfect matching?

YES

Proof.

GS Terminated

$\Rightarrow$ Every $r \in R$ is matched, OR

$\exists$ unmatched $r \in R$ made offer to every $d \in D$.

$\Rightarrow$ M is perfect matching

* Matching : is M a perfect matching?

YES

Proof.

GS Terminated

$\Rightarrow$ Every $r \in R$ is matched, OR

$\exists$ unmatched $r \in R$ made offer to every $d \in D$.

By Observation (*)

Every $d \in D$ received offer

$\Rightarrow$ Every $d \in D$ must be matched.

$\Rightarrow$ M is a perfect matching.

∎

Some Questions

* While loop termination: Does GS always return?

* Matching: is M a perfect matching?

* Stability: is M stable?

* Stability: is M stable? YES!

Proof. Consider any $(d, r), (d', r') \in M$.

Need to show NOT an instability.

* Stability: is M stable? YES!

Proof. Consider any $(d, r), (d', r') \in M$.

Need to show NOT an instability.

Suppose d prefers r' to r.

$\implies$ r' never made d an offer

(otherwise d would be matched w/ $r^* \in R$ at least as preferred as r')

By Observation (*)

* Stability: is M stable? YES!

Proof. Consider any $(d, r), (d', r') \in M$.
Need to show NOT an instability.

Suppose d prefers r' to r.

$\Longrightarrow$ r' never made d an offer
(otherwise d would be matched w/ $r^* \in R$ at least as preferred as r')

$\Longrightarrow$ r' prefers d' over d.
(Residencies make offers in order of preference, made offer to d', but not d.)

* Stability: is M stable? YES!

Proof. Consider any $(d, r), (d', r') \in M$.

Need to show NOT an instability.

Suppose d prefers r' to r.

$\Longrightarrow$ r' never made d an offer

(otherwise d would be matched w/ $r^* \in R$ at least as preferred as r')

$\Longrightarrow$ r' prefers d' over d.

(Residencies make offers in order of preference, made offer to d', but not d.)

So $r' \succ_d r \Longrightarrow d' \succ_{r'} d$ No INSTABILITY. ∎

Gale-Shapley returns a stable perfect matching.

But <u>which</u> stable perfect matching does GS output?

Characterization of Gale-Shapley Matching

For every residency $r \in R$

Consider set of feasible doctors $d \in D$

$$F_r = \left\{ d \in D : \exists \underset{\text{matching}}{\overset{\text{stable}}{\text{perfect}}}\ M \text{ s.t } (d, r) \in M \right\}$$

Characterization of Gale-Shapley Matching

For every residency $r \in R$

Consider set of feasible doctors $d \in D$

$$F_r = \left\{ d \in D : \exists \text{ stable perfect matching } M \text{ s.t. } (d, r) \in M \right\}$$

Let $d_r^* =$ most-preferred $d \in F_r$ according to r.

Characterization of Gale-Shapley Matching

For every residency $r \in R$

Consider set of feasible doctors $d \in D$

$$F_r = \{ d \in D : \exists \text{ stable perfect matching } M \text{ s.t } (d, r) \in M \}$$

Let $d_r^* =$ most-preferred $d \in F_r$ according to r.

Theorem. When Residencies make offers, Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

* Non-obvious! (Not even obvious that M^* is a matching!)

↳ Algorithmic analysis reveals insights into nature of stable matchings.

Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

Proof. By contradiction.

* Suppose in some execution, GS returns matching with (d, r) where $d \neq d_r^*$.

Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

Proof. By contradiction.

* Suppose in some execution, GS returns matching with (d, r) where $d \neq d_r^*$.

$\Longrightarrow$ Some residency r has offer rejected by a feasible $d \in F_r$.

(by order of offers from residencies)

Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

<u>Proof.</u> By contradiction.

* Suppose in some execution, GS returns matching with (d, r) where $d \neq d_r^*$.

$\Longrightarrow$ Some residency r has offer rejected by a feasible $d \in F_r$.

* Consider the <u>first</u> time (during GS) that such a d rejects the offer of r.

Gale-Shapley always returns

$$M^* = \{ (d_r^*, r) : r \in R \}$$

Proof. By contradiction.

* Suppose in some execution, GS returns matching with (d, r) where $d \neq d_r^*$.

$\Longrightarrow$ Some residency r has offer rejected by a feasible $d \in F_r$.

* Consider the first time (during GS) that such a d rejects the offer of r.

$$\Downarrow$$

d elects to be matched w/ r' instead of r

What do we know?

* By execution of GS, d prefers r' to r

* By assumption, there is some stable PM M' that includes $(d, r) \in M'$.

What do we know?

* By execution of GS, d prefers r' to r
* By assumption, there is some stable PM M' that includes $(d, r) \in M'$.

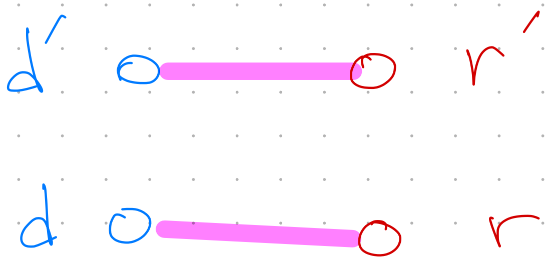
Who is r' matched w/ in M'?

What do we know?

* By execution of GS, d prefers r' to r

* By assumption, there is some stable PM M' that includes $(d, r) \in M'$.

Who is r' matched w/ in M'?



$\Longrightarrow$ $d' \in F_{r'}$ is feasible for r'

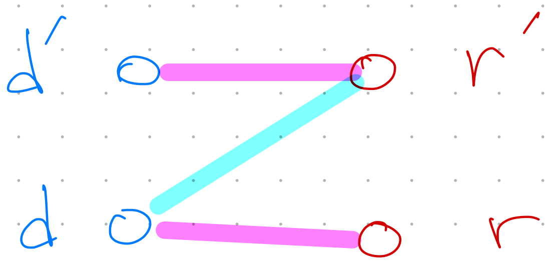
$\Longrightarrow$ d' has not yet rejected r' (by choice of (d, r) as first)

What do we know?

* By execution of GS, d prefers r' to r

* By assumption, there is some stable PM M' that includes $(d, r) \in M'$.

Who is r' matched w/ in M'?



$\Longrightarrow$ $d' \in F_{r'}$ is feasible for r'

$\Longrightarrow$ d' has not yet rejected r' (by choice of (d, r) as first)

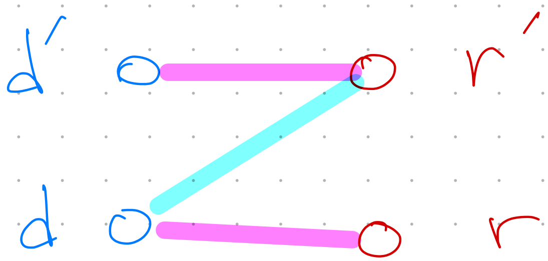
$\Longrightarrow$ r' prefers d to d' (by order of offers)

What do we know?

* By execution of GS, d prefers r' to r

* By assumption, there is some stable PM M' that includes $(d, r) \in M'$.

Who is r' matched w/ in M'?



$\Longrightarrow$ $d' \in F_{r'}$ is feasible for r'

$\Longrightarrow$ d' has not yet rejected r' (by choice of (d, r) as first)

$\Longrightarrow$ r' prefers d to d' (by order of offers)

So (d, r) & (d', r') are an instability!

↳ Contradicts stability of M'. ☒

Conclusion.

* Gale-Shapley returns matching where Residencies receive their best possible feasible doctor.

* On the flip side, doctors receive their least favorable feasible residency.

$\Rightarrow$ GS benefits group that makes offers.

$\Rightarrow$ But order of offers doesn't change solution!