Incentives in Computer Science

Stable matching

- HW:

  "since telling the truth is a dominant strategy"

  $\equiv$ "since the mechanism is truthful"

- If a mechanism is $\boxed{\text{truthful}}$, it is never beneficial to lie, no matter what others do.

- Is every truthful mechanism Pareto-optimal?

No, ignore preferences

Today – stable matching

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

Matching Residents to Hospitals

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

2-sided matching

hosp — residents

1, 2 A — 1 B, A

1, 2 B — 2 B, A

Matching Residents to Hospitals

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

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

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

Stable Matching Problem

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

- Stable assignment. Assignment with no unstable pairs.

favorite ↓ … least favorite ↓

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

Hospital's Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Resident's Preference Profile

X-A , Y-B , Z-C

Stable Matching Problem

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

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

- Stable matching: perfect matching with no unstable pairs.

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

Apologies in advance

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

- I may accidentally say "men" for hospitals and "women" for applicants.

- This is because, for many years, when presenting this material, people spoke of "stable marriage" and used men and women as the two sets.
- In that context, you can think of the problem as studying 1950's dating.

Stable Roommate Problem

- Q. Do stable matchings always exist?
- A. Not obvious a priori.

"nonbipartite graph" version of stable matching

- Stable roommate problem.
  - 2n people; each person ranks others from 1 to 2n-1.
  - Assign roommate pairs so that no unstable pairs.

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Doofus	A	B	C

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Doofus	A	B	C

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Doofus	A	B	C

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

```
Initialize all hospitals and residents to be unmatched

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

```
Initialize all hospitals and residents to be unmatched

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

Hosps

	1st	2nd	3rd	4th
V	[illegible]	A	D	C
W	D	B	A	C
X	B	A	C	D
Y	~~A~~	~~D~~	C	B

Res

	1st	2nd	3rd	4th
A	X	V	W	[illegible]
B	X	W	Y	[illegible]
C	W	X	Y	V
D	V	W	[illegible]	X

Observations:

- hospitals make offers to residents in ↓ order by preference
- once a resident is matched, she stays until termination of alg

$\Rightarrow$ & her successive matches are better & better from her perspective.

- Alg terminates after at most n^2 iterations thru while loop

All hospitals / residents are matched in end (perfect matching)

Pf Say h unmatched at end.

Then h proposed to all residents.

if h unmatched at end, then

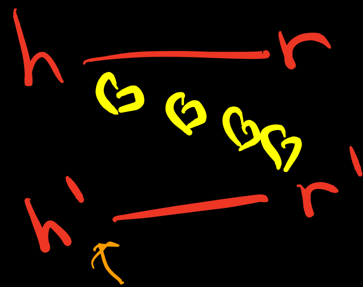
$\exists$ r that is unmatched at end.

$\rightarrow\leftarrow$

Thm The final matching is stable.

Pf by $\rightarrow\leftarrow$

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



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

Case 2: h did propose to r'. $\Rightarrow$ r' prefers h' to h because she rejected h for a better hospital & her matches got better & better

Stability: if no (h, r) unstable pair then ∃ no subset (H, R) that would prefer not to participate

Summary

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

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

- Algorithm underspecified. Q. If there are multiple stable matchings, which one does GS find?

Understanding the Solution

- Algorithm is under-specified.
- Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one

X

A

B

C

	1st	2nd	3rd
X	A	B	C
Y	B	A	C
Z	A	B	C

	1st	2nd	3rd
A	Y	X	Z
B	X	Y	Z
C	X	Y	Z

Understanding the Solution

- Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

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

Understanding the Solution

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

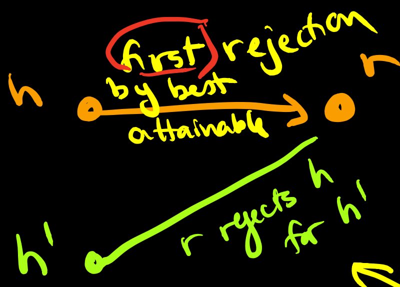
- Hospital-optimal assignment. Each hospital receives **best** attainable match.

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

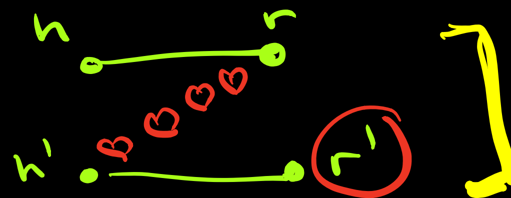
Hospital Optimality

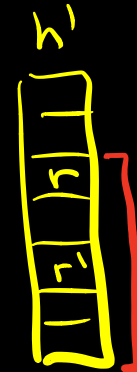
- Claim. GS matching is hospital-optimal.
- Pf. (by contradiction)

In execution of GS, consider first time some hospital h is rejected by its best attainable match r.



∃ stable matching M' where h & r are matched.





r prefers h' to h

Claim: h' prefers r to r'

h' has not yet been rejected by best attainable
best attainable is r or below r, and r' is attainable ⇒ h' prefers r to r'

Stable Matching Summary

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

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

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

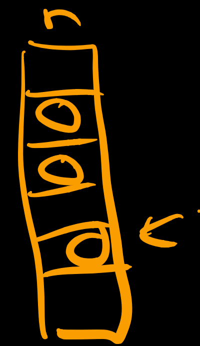
- Hospital-optimality. In version of GS where hospitals make offers, each hospital receives best attainable match.
-

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

- Q. What about the residents/applicants?

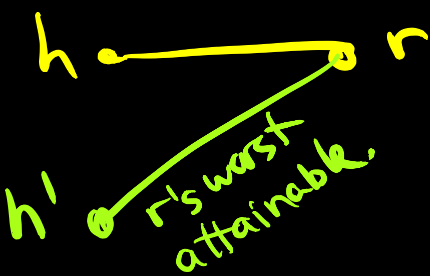
Resident Pessimality

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

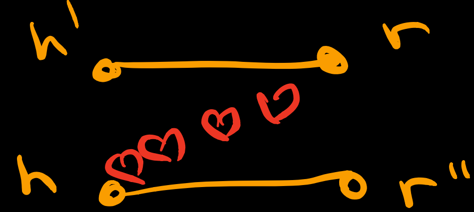


Proof:

output of GS & h is not r's worst attainable



∃ matching M'' in which h' & r are matched.



by hospital optimality h likes r more than r''

r likes h more than h' because h' is r's worst attain

Honesty

- Are the participants in a stable matching algorithm motivated to report their preferences truthfully?

Honesty for residents in hospital-proposing version

	1st	2nd	3rd
X	C	A	B
Y	A	C	B
Z	C	A	B

Hospitals preferences

	1st	2nd	3rd
A	X	Y	Z
B	X	Y	Z
C	Y	X	Z

Residents preferences

not truthful for non-proposing side

if C reports Y Z X

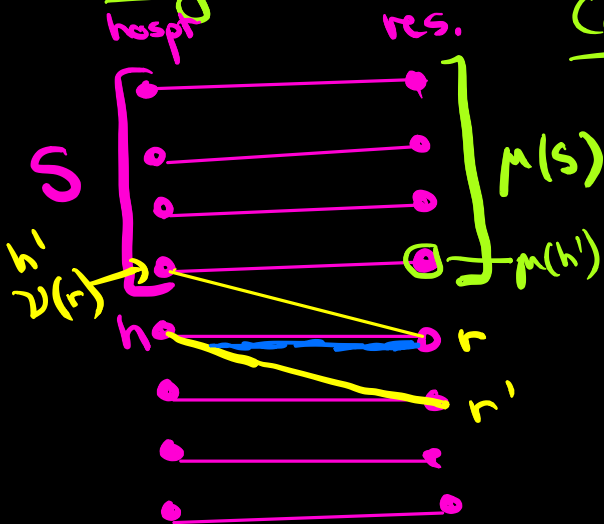
C will end up with a better match from perspective of true preferences.

Thm: The GS alg is truthful for proposing side.

Lemma: Suppose μ is hosp-opt stable matching. Let ν be any other matching. Let S be hospitals that prefer their match in ν to their match in μ (GS). $\exists\,(h,r)$ that are unstable in ν s.t. $h \notin S$.

pink is outcome of GS (hosp opt.)

Proof!



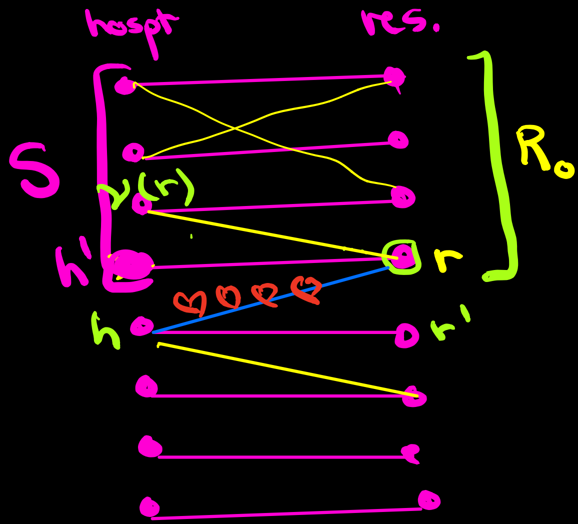
Case 1: $\mu(S) \neq \nu(S)$

Claim: (h,r) is unstable for ν yellow

- Since $h \notin S$, h doesn't like r' as much as r.
- r prefers h to h'

h' proposed to r before h' proposed to $\mu(h')$ & was rejected by r.

Case 2: $\mu(S) = \nu(S) = R_0$



ν yellow.

During GS execution, each $r \in R_0$ received & rejected a proposal from her match in ν.

Let r be last one in R_0 to receive a proposal during GS (from some hospital, say h')

Claim: at that pt, r was tentatively matched to h who she rejected for h'.

h must be outside S

(h, r) is unstable for ν

h likes r at least as much as $\mu(h) = r'$

likes r' at least as much as $\nu(h)$

r likes h at least as much as $\nu(r)$.

because $\nu(r)$ proposed to r before h did which was before h' did $\blacksquare$

[redacted]