Algorithmic Game Theory
COMP6207

Lecture 13: **Hospitals/Residents Problem**

Baharak Rastegari
b.rastegari@soton.ac.uk
Electronics and Computer Science
University of Southampton

Learning Outcomes

- By the end of this session, you should be able to
  - ***Describe*** the hospital/residents (HR) problems and its variants
  - ***Describe*** the relationship between hospital-optimal matchings and resident-optimal matchings
  - ***Identify*** blocking pairs in instances of HR
  - ***Compute*** the matching produced by Resident-oriented GS (RGS)
  - ***Compute*** the matching produced by Hospital-oriented GS (HGS)
  - ***Describe*** the implications of couples participating in HR

Extensions of Stable Matching problem

- Agents may declare some candidates unacceptable
  → **S**table **M**atching problem with **I**ncomplete lists (**SMI**)
- Agents may be indifferent among several candidates
  → **S**table **M**atching problem with **T**ies (**SMT**)
- Both incomplete lists and indifferences are allowed
  → **S**table **M**atching problem with **T**ies and **I**ncomplete lists (**SMTI**)

Today

- Agents on one side can get matched to several candidates
  - Many-one stable matching problem
  - **H**ospitals/**R**esidents problem (**HR**) and HR with **T**ies (**HRT**)

Note: In HR incomplete lists are allowed.

From Stable Marriage to the **Hospitals/Residents problem**

***Match Day 2017**. Credit: Charles E. Schmidt College of Medicine, FAU.*
For more photos of this important day of medical students' life click here.

and its variants

How it works in practice, usually

- Junior doctors (or residents) must undergo training in hospitals
- Applicants rank hospitals in order of preference
- Hospitals do likewise with their applicants
- Centralised matching schemes (clearinghouses) produce a matching in several countries
  - US (National Resident Matching Program)
  - Canada (Canadian Resident Matching Service)
  - Japan (Japan Residency Matching Program)
  - UK (UK Foundation Programme Office)

- Stability is the key property of a matching
  - [Roth, 1984]

Many-One variants of SMI and SMTI

- Agents on one side can get matched to several candidates
  - Many-one stable matching problem
  - **H**ospitals/**R**esidents problem (**HR**) and HR with **T**ies (**HRT**)

Note: In HR incomplete lists are allowed.

Hospitals / Residents problem (HR)

- **Participants**
  - A set of n_1 residents $\{r_1, r_2, \ldots, r_{n_1}\}$
  - A set of n_2 hospitals $\{h_1, h_2, \ldots, h_{n_2}\}$
- Each hospital has a **capacity**
- **Preferences**
  - Residents rank acceptable hospitals in strict order of preference, hospitals do likewise
  - We assume that unacceptability is mutual: if **h** finds **r** unacceptable then **r** finds **h** unacceptable and vice versa

Matching in HR

A *matching* M is a set of resident-hospital pairs such that:

- $(r,h) \in M \Rightarrow r, h$ find each other acceptable
- No resident appears in more than one pair
- No hospital appears in more pairs than its capacity

HR: example matching

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

HR: example matching

r_1: h_2 h_1

r_2: h_1 h_2

r_3: h_1 h_3

r_4: h_2 h_3

r_5: h_2 h_1

r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6

h_2: r_2 r_6 r_1 r_4 r_5

h_3: r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_5, h_2), (r_6, h_1)\}$ (size **5**)

HR: stability

- Matching M is *stable* if M admits no *blocking pair*
  - (r, h) is a blocking pair of matching M if:

1. r, h find each other acceptable

   and

2. *either* r is unmatched in M
   or r prefers h to his/her assigned hospital in M

   and

3. *either* h is undersubscribed in M
   or h prefers r to its worst resident assigned in M

HR: blocking pair (1)

r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1 : r_1 r_3 r_2 r_5 r_6
h_2 : r_2 r_6 r_1 r_4 r_5
h_3 : r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_5, h_2), (r_6, h_1)\}$ (size **5**)

(r_2, h_1) is a blocking pair of M

HR: blocking pair (2)

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_5, h_2), (r_6, h_1)\}$ (size **5**)

(r_4, h_2) is a blocking pair of M

HR: blocking pair (3)

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_5, h_2), (r_6, h_1)\}$ (size **5**)

(r_4, h_3) is a blocking pair of M

HR: stable matching

r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1 : r_1 r_3 r_2 r_5 r_6
h_2 : r_2 r_6 r_1 r_4 r_5
h_3 : r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_5, h_2), (r_6, h_1)\}$ (size **5**)

r_5 is unmatched
h_3 is undersubscribed

Stable Matching in HR

A *matching* M is a set of resident-hospital pairs such that:

- $(r,h) \in M \Rightarrow r, h$ find each other acceptable
- No resident appears in more than one pair
- No hospital appears in more pairs than its capacity

Matching M is *stable* if M admits no *blocking pair*

– (r, h) is a blocking pair of matching M if:

1. r, h find each other acceptable

   and

2. *either* r is unmatched in M
   or r prefers h to his/her assigned hospital in M

   and

3. *either* h is undersubscribed in M
   or h prefers r to its worst resident assigned in M

Resident-oriented GS **(RGS)**

An extension of **Gale-Shapley** to **HR**

Resident-oriented GS (residents, hospitals, capacities, preferences)

$M = \emptyset$; //assign all residents and hospitals to be free

While (some resident r_i is unmatched and has a non-empty list)

  r_i applies to the first hospital h_j on her list;

  $M = M \cup \{(r_i, h_j)\}$;

  If (h_j is over-subscribed)

    r_k = worst resident assigned to h_j;

    $M = M \setminus \{(r_k, h_j)\}$; //$r_k$ is set free

  If (h_j is full)

    r_k = worst resident assigned to h_j;

    For (each successor r_l of r_k on h_j's list)

      delete r_l from h_j's list;

      delete h_j from r_l's list;

output the engaged pairs, who form a stable matching;

RGS algorithm: example

r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : $\cancel{h_2}$ h_3
r_5 : $\cancel{h_2}$ $\cancel{h_1}$
r_6 : $\cancel{h_1}$ h_2

Resident preferences

Each hospital has capacity **2**

h_1 : r_1 r_3 r_2 $\cancel{r_5}$ $\cancel{r_6}$
h_2 : r_2 r_6 r_1 $\cancel{r_4}$ $\cancel{r_5}$
h_3 : r_4 r_3

Hospital preferences

RGS algorithm: example

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

Stable matching: $M = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_3), (r_6, h_2)\}$

Optimal stable matchings

Straightforward extension of results for **SMI:**

Claim. When preferences are **strict**, all executions of resident-oriented GS (RGS) return the resident-optimal (stable) matching.

Claim. When preferences are **strict**, all executions of hospital-oriented GS (HGS) return the hospital-optimal (stable) matching.

Claim. When preferences are strict, the hospital-optimal matching is the resident-pessimal matching, and vice versa.

Hospital-oriented GS **(HGS)**

An extension of **Gale-Shapley** to **HR**

```
Hospital-oriented GS (residents, hospitals, capacities, preferences)
    M = ∅;  //assign all residents and hospitals to be free
    While (some hospital h_i is undersubscribed and has non-empty list)
          s_i = q_i - |M(h_i)|; //the number of available seats in h_i
          h_i applies to the first s_i residents on its list;
          foreach (r_j that h_i applies to)
               If (r_j is free)   M = M ∪{(r_j, h_i)};
               else if (r_j prefers h_i to her current hospital h')
                    M= M ∪{(r_j, h_i)} \ {(r_j, h')}; //h' is set free
               If (r_j is not free)
                    For (each successor h_l of M(r_j) on r_j's list)
                         delete h_l from r_j's list;
                         delete r_j from h_l's list;
      output the engaged pairs, who form a stable matching;
```

This If statement is not necessary.
We can skip to For loop right away.

HGS algorithm: example

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

HGS algorithm: example

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 $\cancel{h_3}$
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 $\cancel{r_3}$

Hospital preferences

HGS algorithm: example

r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

Resident preferences

Each hospital has capacity 2

h_1: r_1 r_3 r_2 r_5 r_6
h_2: r_2 r_6 r_1 r_4 r_5
h_3: r_4 r_3

Hospital preferences

Stable matching: $M = \{(r_1, h_1), (r_2, h_2), (r_3, h_1), (r_4, h_3), (r_6, h_2)\}$

Hospital-optimal matching

RGS algorithm: example

r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1 : r_1 r_3 r_2 r_5 r_6
h_2 : r_2 r_6 r_1 r_4 r_5
h_3 : r_4 r_3

Hospital preferences

Stable matching: $M = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_3), (r_6, h_2)\}$

Resident-optimal matching

Hospital-optimal vs Resident-optimal

r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

Resident preferences

Each hospital has capacity **2**

h_1 : r_1 r_3 r_2 r_5 r_6
h_2 : r_2 r_6 r_1 r_4 r_5
h_3 : r_4 r_3

Hospital preferences

Resident-optimal matching

Hospital-optimal matching

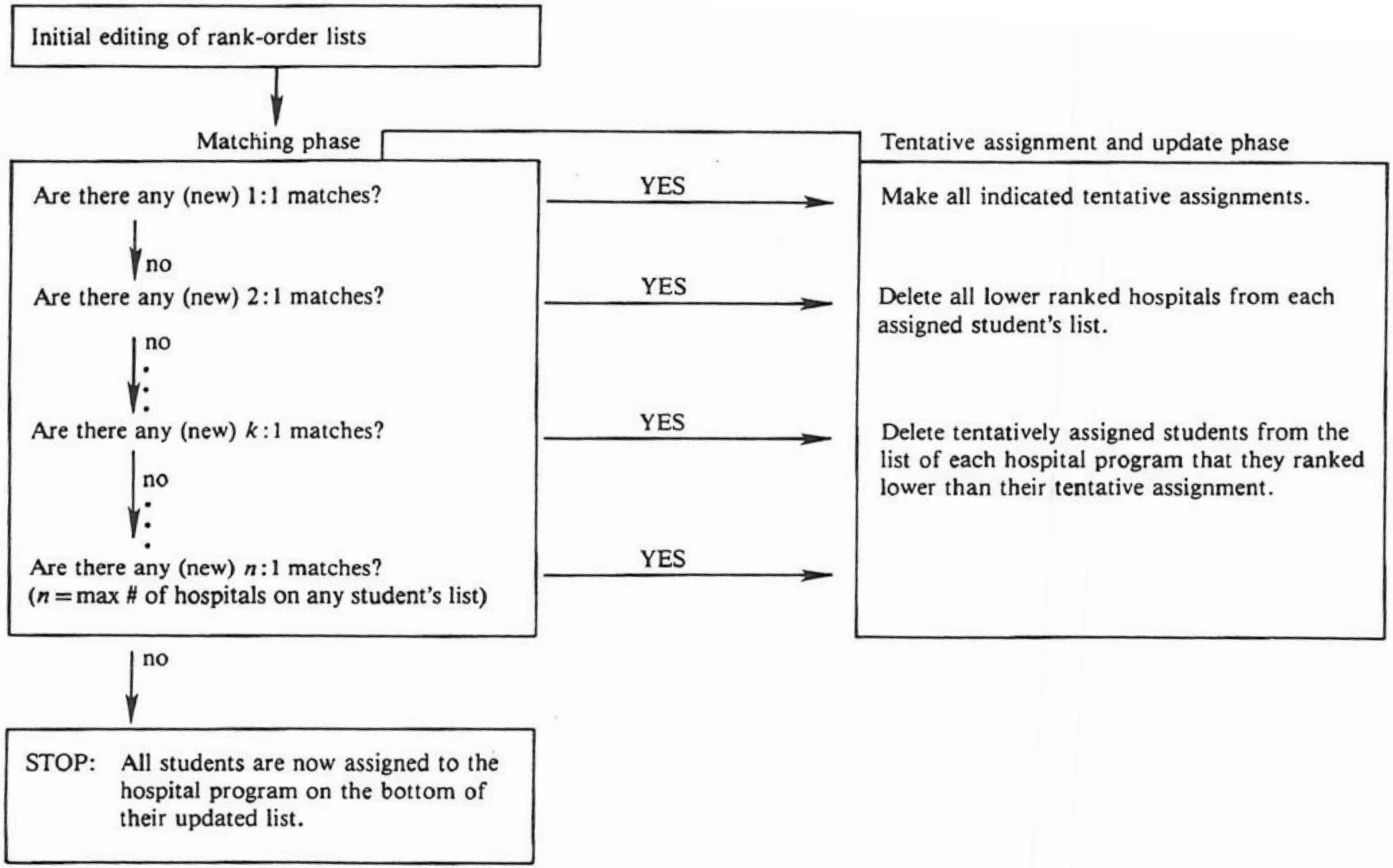

Figure 5.1. The NIMP algorithm.

NIMP (later NRMP) algorithm, adopted in 1951.
Figure from TSM, reproduced from Roth 1984

Rural Hospitals Theorem

Theorem (Roth, 1984; Gale and Sotomayor, 1985; Roth, 1986)

In a given instance of **HR**:

1. the same residents are assigned in all stable matchings;
2. each hospital is assigned the same number of residents in all stable matchings;
3. any hospital that is undersubscribed in one stable matching is assigned exactly the same set of residents in all stable matchings.

DS truthfulness

Theorem (Roth; 5.16 in TSM)

A stable matching mechanism that yields the resident-optimal matching makes it a dominant strategy for all residents to state their true preferences.

- Under strict preferences, RGS is dominant-strategy truthful for residents.

Theorem (Roth; 5.14 in TSM)

No stable matching mechanism exists that makes it a dominant strategy for all hospitals to state their true preferences.

- Even when hospital-oriented GS (HGS) is executed and preferences are strict, some hospitals may benefit from misreporting their preferences.

Hospitals / Residents problem with Ties (HRT)

- In practice, residents' preference lists are short
- Hospitals' lists are generally long, so *ties* may be used – ***H****ospitals /* ***R****esidents problem with* ***T****ies* (***HRT***)
- A hospital may be *indifferent* among several residents
  - E.g., $\mathbf{h_1}$: $(\mathbf{r_1}\ \mathbf{r_3})\ \mathbf{r_2}\ (\mathbf{r_5}\ \mathbf{r_6}\ \mathbf{r_8})$

- An instance of HRT may not admit a hospital-optimal and/or a resident-optimal matching.
- A matching $\boldsymbol{M}$ is stable in an HRT instance $\boldsymbol{I}$ if and only if $\boldsymbol{M}$ is stable in some instance $\boldsymbol{I'}$ of HR obtained from $\boldsymbol{I}$ by breaking the ties [Manlove et al, 1999].

Couples in HR

- Pairs of residents who wish to be matched to geographically close hospitals form *couples*
- Each couple (r_i, r_j) ranks in order of preference a set of pairs of hospitals (h_p, h_q) representing the assignment of r_i to h_p and r_j to h_q
- Stability definition may be extended to this case [Roth, 1984; McDermid and Manlove, 2010; Biró et al, 2011]
- Gives the ***H**ospitals / **R**esidents problem with **C**ouples* (**HRC**)
- A stable matching need not exist:

(r_1, r_2) : (h_1, h_2)

r_3 : h_1 h_2

h_1 : 1 : r_1 r_3 r_2

h_2 : 1 : r_1 r_3 r_2

Couples in HR

- Pairs of residents who wish to be matched to geographically close hospitals form *couples*
- Each couple (r_i, r_j) ranks in order of preference a set of pairs of hospitals (h_p, h_q) representing the assignment of r_i to h_p and r_j to h_q
- Stability definition may be extended to this case [Roth, 1984; McDermid and Manlove, 2010; Biró et al, 2011]
- Gives the ***H**ospitals / **R**esidents problem with **C**ouples* (**HRC**)
- A stable matching need not exist:

(r_1,r_2) : (h_1,h_2)

r_3 : h_1 h_2

h_1:1: r_1 r_3 r_2

h_2:1: r_1 r_3 r_2

Couples in HR

- Pairs of residents who wish to be matched to geographically close hospitals form *couples*
- Each couple (r_i, r_j) ranks in order of preference a set of pairs of hospitals (h_p, h_q) representing the assignment of r_i to h_p and r_j to h_q
- Stability definition may be extended to this case [Roth, 1984; McDermid and Manlove, 2010; Biró et al, 2011]
- Gives the ***H**ospitals / **R**esidents problem with **C**ouples* (**HRC**)
- A stable matching need not exist:

$(r_1, r_2) : (h_1, h_2)$ $h_1 : 1 : r_1\ r_3\ r_2$

$r_3 : h_1\ h_2$ $h_2 : 1 : r_1\ r_3\ r_2$

- Stable matchings can have different sizes

HRC is hard

- The problem of determining whether a stable matching exists in a given HRC instance is NP-complete, even if each hospital has capacity **1** and:
  - there are no single residents
    [Ng and Hirschberg, 1988; Ronn, 1990]
  - there are no single residents, *and*
  - each couple has a preference list of length **≤2**, *and*
  - each hospital has a preference list of length **≤3**
    [Manlove and McBride, 2013]
  - the preference list of each single resident, couple and hospital is derived from a strictly ordered master list of hospitals, pairs of hospitals and residents respectively [Biró et al, 2011], *and*
  - each preference list is of length **≤3**, *and*
  - the instance forms a "dual market"
    [Manlove and McBride, 2013]

Algorithm for HRC

Not in the Exam

- Algorithm C described in [Biró et al, 2011]:
- A Gale-Shapley like heuristic
- An *agent* is a single resident or a couple
- Agents apply to entries on their preference lists
- When a member of an assigned couple is rejected their partner must withdraw from their assigned hospital
- This creates a vacancy – so any resident previously rejected by the hospital in question may have to be reconsidered

- The algorithm need not terminate
  - if it terminates, the matching found is guaranteed to be stable (according to the definition of stability by Biró et al, 2011)
  - it cannot terminate if there is no stable matching
  - it need not terminate even if there is a stable matching

Algorithm C: example

Not in the Exam

Resident preferences

$r_3 \quad : \quad \cancel{h_1} \ h_5$

$\boxed{r_7} \quad : \quad \cancel{h_6} \ \cancel{h_8}$

$(r_1, r_5) \quad : \quad (h_1, h_2) \quad (h_3, h_6)$

$(r_2, r_4) \quad : \quad (\cancel{h_4}, \cancel{h_5}) \quad (\cancel{h_1}, \cancel{h_2}) \quad (h_3, h_7)$

$\boxed{(r_6, r_8)} \quad : \quad (\cancel{h_6}, \cancel{h_8})$

cycle

Hospitals' preferences derived from the following master list:

$r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7 \ r_8$

Each hospital has capacity **1**

Stable matching

Not in the Exam

Resident preferences

$\mathbf{r}_3$: $\mathbf{h}_1$ $\mathbf{h}_5$

$\mathbf{r}_7$: $\mathbf{h}_6$ $\mathbf{h}_8$

$(\mathbf{r}_1,\mathbf{r}_5)$: $(\mathbf{h}_1,\mathbf{h}_2)$ $(\mathbf{h}_3,\mathbf{h}_6)$

$(\mathbf{r}_2,\mathbf{r}_4)$: $(\mathbf{h}_4,\mathbf{h}_5)$ $(\mathbf{h}_1,\mathbf{h}_2)$ $(\mathbf{h}_3,\mathbf{h}_7)$

$(\mathbf{r}_6,\mathbf{r}_8)$: $(\mathbf{h}_6,\mathbf{h}_8)$

Hospitals' preferences

$\mathbf{r}_1$ $\mathbf{r}_2$ $\mathbf{r}_3$ $\mathbf{r}_4$ $\mathbf{r}_5$ $\mathbf{r}_6$ $\mathbf{r}_7$ $\mathbf{r}_8$

Each hospital has capacity **1**

Stable matching: $M = \{(r_1, h_3), (r_2, h_1), (r_3, h_5), (r_4, h_2), (r_5, h_6), (r_7, h_8)\}$

Empirical evaluation

Not in the Exam

- Extensive empirical evaluation due to [Biró et al, 2011]:
- Compared 5 variants of Algorithm C against 10 other algorithms
- Instances generated with varying:
  - sizes
  - numbers of couples
  - densities of the "compatibility matrix"
  - lengths of time given to each instance
- Measured proportion of instances found to admit a stable matching
- Clear conclusion:
  - high likelihood of finding a stable matching (with Algorithm C) if the number / proportion of couples is low

What was in this lecture

- Hospitals/Residents problem (HR) and HR with ties (HRT)
- Resident-oriented and Hospital-oriented GS for HR
- Rural hospitals theorem
- DS truthfulness in HR
- Couples in HR

Acknowledgement

Some of the slides in this lecture were based on the slides by **David Manlove**.

Books

- Algorithmics of Matching under Preferences by David F. Manlove.

- Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis (**TSM**) by Alvin E. Roth, Marilda A. Oliviera Sotomayor.

- Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations (**MAS**) by Yoav Shoham and Kevin Leyton-Brown

- Algorithmic Game Theory (**AGT**), edited by Noam Nisan, Tim Roughgarden, Eva Tardos,Vijay V. Vazirani

Optional additional reading

- On the ``history of labor market for medical interns''
  - **TSM** section 1.1, or alternatively
  - *The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory* by Alvin E. Roth, Journal of Political Economy, Vol 92, 1984

- The NIMP algorithm
  - **TSM** section 5.4
  - *The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory* by Alvin E. Roth, Journal of Political Economy, Vol 92, 1984