European Agents Systems Summer School

EASSS 2018

Stable Matchings: in Theory and in Practice

Bahar Rastegari

University of BRISTOL

Special thanks to **David Manlove**, from whose excellent slides this talk has benefited from.

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

Nobel prize in Economic Sciences, 2012

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012
Alvin E. Roth, Lloyd S. Shapley

Photo: © Linda A. Cicero/Stanford
Alvin E. Roth

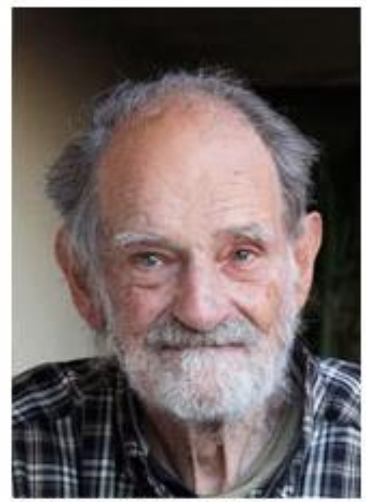


Photo: AP Photo/Reed Saxon
Lloyd S. Shapley

The prize was awarded jointly to Alvin E. Roth and Lloyd S. Shapley
``**for the theory of stable allocations and the practice of market design**''.

Part 1

Please note that the PDF version of these slides that are available through EASSS link does not include the animations. If you would like to have the original PPT version (that includes animations) please send me an email to: baharakr@gmail.com

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

Stable Marriage problem (SM)

- A market with two disjoint sets of agents
  - 2 men: Adam, Bob
  - 2 women: Alice, Bella

- Each agent has a strict preference ordering over the agents on the other side of the market (a.k.a candidates).

Adam:	Alice	Bella	**Alice:**	Adam	Bob
Bob:	Bella	Alice	**Bella:**	Adam	Bob

- Goal: identifying a stable marriage (matching)
- Applications: many, including college admission, hospitals/residents problem

Applications of SM

- Assigning residents to hospitals
- Assigning children to daycare places
- Assigning children to schools (school choice programs)
- Assigning students to colleges/universities (higher education admission)
- Placing military cadets in branches and assigning naval cadets to billets
- Hiring federal judicial law clerks
- Placement of graduating rabbis
- Online dating and online matrimony
- Auction mechanisms for sponsored search
- Kidney exchange

Stable Matchings

Matching:

A pairing of women and men such that each man is paired with at most one woman and vice versa.

Adam:	Alice	Bella	**Alice:**	Adam	Bob
Bob:	Bella	Alice	**Bella:**	Adam	Bob

Stable Matchings

Matching:

A pairing of women and men such that each man is paired with at most one woman and vice versa.

Blocking pair:

A pair who prefer each other to their current partners.

Adam:	Alice	Bella	**Alice:**	Adam	Bob
Bob:	Bella	Alice	**Bella:**	Adam	Bob

Alice and Adam block this matching

Stable Matchings

Matching:

A pairing of women and men such that each man is paired with at most one woman and vice versa.

Blocking pair:

A pair who prefer each other to their current partners.

Stable matching:

A matching with no blocking pair.

Adam:	Alice	Bella	**Alice:**	Adam	Bob
Bob:	Bella	Alice	**Bella:**	Adam	Bob

Not stable

Stable Matchings

Matching:

A pairing of women and men such that each man is paired with at most one woman and vice versa.

Blocking pair:

A pair who prefer each other to their current partners.

Stable matching:

A matching with no blocking pair.

Adam:	Alice	Bella	**Alice:**	Adam	Bob
Bob:	Bella	Alice	**Bella:**	Adam	Bob

Stable

Does every instance of SM problem admit a stable matching?

Theorem (Gale & Shapley, 1962)

A stable matching always exists, and can be found in polynomial time.

```
Deferred-acceptance-man-oriented (men, women, preference orderings)
      Assign all men and women to be free;
      While (some man m is free) {
            w = first woman on m's list to whom m hasn't yet proposed;
            If (w is free)
               assign m and w to be engaged;
            else if (w prefers m to her fiancé m') {
               set m and w to be engaged;
               set m' to be free;
            } else
               w rejects m; //and m remains free
      }
      output the n engaged pairs, who form a stable matching;
```

Example: Man-oriented Gale Shapley (MGS)

```
Assign all men and women to be free;
While (some man m is free) {
      w = first woman on m's list to whom m hasn't yet proposed;
      If (w is free)
         assign m and w to be engaged;
      else if (w prefers m to her fiancé m') {
         set m and w to be engaged;
         set m' to be free;
      } else
         w rejects m; //and m remains free
}
```

m_1: w_2 w_1 w_3

m_2: w_1 w_2 w_3

m_3: $\cancel{w_1}$ $\cancel{w_2}$ w_3

w_1: m_1 m_2 $\cancel{m_3}$

w_2: m_2 m_1 $\cancel{m_3}$

w_3: m_1 m_3 m_2

Extensions of Stable Marriage problem

- Agents may declare some candidates unacceptable
  → **S**table **M**arriage problem with **I**ncomplete lists (**SMI**)

Matching: a set of (man,woman) acceptable pairs
Blocking pair: a pair who prefer each other to their current partners.
(Assume that agents prefer getting matched to an acceptable candidate to remaining unmatched.)

- Agents may be indifferent among several candidates
  → **S**table **M**arriage problem with **T**ies (**SMT**)
- Both incomplete lists and indifferences are allowed
  → **S**table **M**arriage problem with **T**ies and **I**ncomplete lists (**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**)
- Many-to-many stable matching problem
  - **W**orkers/**F**irms problem (**WF**)

Extensions of Stable Marriage problem

- Agents may declare some candidates unacceptable
  → **S**table **M**arriage problem with **I**ncomplete lists (**SMI**)

Matching: a set of (man,woman) acceptable pairs
Blocking pair: a pair who prefer each other to their current partners.
(Assume that agents prefer getting matched to an acceptable candidate to remaining unmatched.)

- Agents may be indifferent among several candidates
  → **S**table **M**arriage problem with **T**ies (**SMT**)
- Both incomplete lists and indifferences are allowed
  → **S**table **M**arriage problem with **T**ies and **I**ncomplete lists (**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**)
- Many-to-many stable matching problem
  - **W**orkers/**F**irms problem (**WF**)

What's next in the tutorial

1.1. Classical Hospitals / Residents problem

1.2. Hospitals / Residents problem with Ties

1.3. Hospitals / Residents problem with Couples

1.4. "Almost stable" matchings

1.5: Social Stability

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]

Hospitals / Residents problem (HR)

- There are n_1 *residents* $r_1, r_2, \ldots, r_{n_1}$ and n_2 *hospitals* $h_1, h_2, \ldots, h_{n_2}$
- Each hospital has a *capacity*
- Residents rank hospitals in order of preference, hospitals do likewise
- r finds h *acceptable* if h is on r's preference list, and unacceptable otherwise (and vice versa)
- A *matching* M is a set of resident-hospital pairs such that:
  1. $(r,h) \in M \Rightarrow r, h$ find each other acceptable
  2. No resident appears in more than one pair
  3. 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

HR: classical results

- A stable matching always exists and can be found in linear time [Gale and Shapley, 1962; Gusfield and Irving, 1989]
- There are *resident-optimal* and *hospital-optimal* stable matchings
- Stable matchings form a distributive lattice [Conway, 1976; Gusfield and Irving, 1989]
- "Rural Hospitals Theorem": for 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.

  [Roth, 1984; Gale and Sotomayor, 1985; Roth, 1986]

Resident-optimal and hospital-optimal stable matchings

- In a *resident-optimal* stable matching
  - each assigned resident is matched with the best hospital s/he can obtain in a stable matching, and
  - each unassigned resident is unassigned in all stable matchings.

- In a *hospital-optimal* stable matching
  - every full hospital $\mathbf{h_j}$ is assigned its $\mathbf{c_j}$ ($\mathbf{c_j}$ being its capacity) best stable partners, and
  - every undersubscribed hospital is assigned the same set of residents in every stable matching.

- The resident-optimal stable matching is *worst-possible* for the hospitals and vice versa.
- The Resident-oriented Gale-Shapley (next slide) produces the resident-optimal stable matching.

Resident-oriented Gale-Shapley algorithm

```
M = ∅;  //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 ∪ {(r_i, h_j)};
        If (h_j is over-subscribed) {
            r_k = worst resident assigned to h_j;
            M = M \ {(r_k, h_j)};
        }
        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 lis;t //r_l is set free
                delete h_j from r_l's list;
            }
        }
```

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

What’s next in the tutorial

1.1. Classical Hospitals / Residents problem

1.2. Hospitals / Residents problem with Ties

1.3. Hospitals / Residents problem with Couples

1.4. “Almost stable” matchings

1.5: Social Stability

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., $\mathbf{h_1}$: $(\mathbf{r_1}\ \mathbf{r_3})\ \mathbf{r_2}\ (\mathbf{r_5}\ \mathbf{r_6}\ \mathbf{r_8})$
- Matching M is *stable* if there is no pair (r,h) such that:
  1. r, h find each other acceptable
  2. *either* r is unmatched in M
     or r prefers h to his/her assigned hospital in M
  3. *either* h is undersubscribed in M
     or h prefers r to its worst resident assigned in M

- A matching M is stable in an HRT instance I if and only if M is stable in some instance I' of HR obtained from I by breaking the ties [Manlove et al, 1999]

HRT: stable matching (1)

r_1: h_1 h_2
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_2 r_3 r_5 r_6
h_2: r_2 r_1 r_6 (r_4 r_5)
h_3: r_4 r_3

Hospital preferences

HRT: stable matching (1)

r_1: h_1 h_2
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_2 r_3 r_5 r_6
h_2: r_2 r_1 r_6 (r_4 r_5)
h_3: r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_1), (r_3, h_3), (r_4, h_2), (r_6, h_2)\}$ (size 5)

HRT: stable matching (2)

r_1: h_1 h_2
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_2 r_3 r_5 r_6
h_2: r_2 r_1 r_6 $(r_4$ $r_5)$
h_3: r_4 r_3

Hospital preferences

$M = \{(r_1, h_1), (r_2, h_1), (r_3, h_3), (r_4, h_3), (r_5, h_2), (r_6, h_2)\}$ (size **6**)

Maximum stable matchings

- Stable matchings can have different sizes
- A maximum stable matching can be (at most) twice the size of a minimum stable matching
- Problem of finding a maximum stable matching (MAX HRT) is NP-hard [Iwama, Manlove et al, 1999], even if (simultaneously):
  - each hospital has capacity **1** (SMTI)
  - the ties occur on one side only
  - each preference list is either strictly ordered or is a single tie
  - *and*
    - *either* each tie is of length **2** [Manlove et al, 2002]
    - *or* each preference list is of length **≤3** [Irving, Manlove, O'Malley, 2009]
- Minimisation problem is NP-hard too, for similar restrictions! [Manlove et al, 2002]

Reminder: computational complexity

- Given two functions f and g, we say $f(n)=O(g(n))$ if there are positive constants c and N such that $f(n) \leq c.g(n)$ for all $n \geq N$
- An algorithm for a problem has *time complexity* $O(g(n))$ if its running time f satisfies $f(n)=O(g(n))$ where n is the input size
- An algorithm runs in *polynomial time* if its time complexity is $O(n^k)$ for some constant k, where n is the input size
- A *decision problem* is a problem whose solution is yes or no for any input
- A decision problem belongs to the class **P** if it can be solved by a *polynomial-time algorithm*
- A decision problem belongs to the class **NP** if it can be verified in *polynomial time*
- A decision problem *A* is **NP**-hard if every other problem in **NP** reduces to *A*.
- A decision problem *A* is **NP**-complete if it **NP**-hard and it belongs to **NP**.
- If a decision problem is **NP**-complete it has no polynomial-time algorithm unless **P=NP**

Reminder: approximation algorithms

- An *optimisation problem* is a problem that involves maximising or minimising (subject to a suitable measure) over a set of feasible solutions for a given instance
  - e.g., colour a graph using as few colours as possible
- If an optimisation problem is **NP**-hard it has no polynomial-time algorithm unless **P=NP**
- An *approximation algorithm* A for an optimisation problem is a polynomial-time algorithm that produces a feasible solution $A(I)$ for any instance I.
- A has *performance guarantee* c, for some $c>1$ if
  - $|A(I)| \leq c.opt(I)$ for any instance I (in the case of a minimisation problem)
  - $|A(I)| \geq (1/c).opt(I)$ for any instance I (in the case of a maximisation problem)

  where $opt(I)$ is the measure of an optimal solution and $|A(I)|$ the size of the solution produced by A.

  ➢ We say that A is a ***c-approximation*** algorithm for this problem.

Master lists

- In practice there may be a common ranking of residents according to some objective criteria (e.g., academic ability) – a *master list*

- Each hospital's preference list is then derived from this master list

- Depending on how fine-grained the scoring system is, ties may arise as a result of residents having equal scores

- MAX HRT is NP-hard even if (simultaneously):
  - each hospital's preference list is derived from a master list of residents
  - each resident's preference list is derived from a master list of hospitals
  - each hospital has capacity **1**
  - *and*
    - *either* there is only a single tie that occurs in one of the master lists
    - *or* the ties occur in one master list only and are of length **2**

  [Irving, M and Scott, 2008]

Example: master list

$\mathbf{r}_1$: $\mathbf{h}_1$ $\mathbf{h}_2$
$\mathbf{r}_2$: $\mathbf{h}_1$ $\mathbf{h}_2$
$\mathbf{r}_3$: $\mathbf{h}_1$ $\mathbf{h}_2$
$\mathbf{r}_4$: $\mathbf{h}_2$ $\mathbf{h}_3$
$\mathbf{r}_5$: $\mathbf{h}_1$ $\mathbf{h}_3$
$\mathbf{r}_6$: $\mathbf{h}_2$

Resident preferences

Each hospital has capacity **2**

$\mathbf{h}_1$: $\mathbf{r}_1$ $\mathbf{r}_2$ $\mathbf{r}_3$ $\mathbf{r}_5$
$\mathbf{h}_2$: $\mathbf{r}_1$ $\mathbf{r}_2$ $(\mathbf{r}_3$ $\mathbf{r}_4)$ $\mathbf{r}_6$
$\mathbf{h}_3$: $\mathbf{r}_4$ $\mathbf{r}_5$

Hospital preferences

Hospitals' preferences derived from the following master list:
$\mathbf{r}_1$ $\mathbf{r}_2$ $(\mathbf{r}_3$ $\mathbf{r}_4)$ $\mathbf{r}_5$ $\mathbf{r}_6$

Residents' preferences derived from the following master list:
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_3$

MAX HRT: approximability

- MAX HRT is not approximable within **33/29** unless P=NP, even if each hospital has capacity **1** [Yanagisawa, 2007]
- MAX HRT is not approximable within **4/3-ε** assuming the *Unique Games Conjecture* (UGC) [Yanagisawa, 2007]
- Trivial **2**-approximation algorithm for MAX HRT
- Succession of papers gave improvements, culminating in:
  - MAX HRT is approximable within **3/2** [McDermid, 2009; Király, 2012; Paluch 2012]
- Experimental comparison of approximation algorithms and heuristics for MAX HRT and MAX SMTI [Irving and M, 2009; Podhradský 2010]

Kiraly's $\frac{3}{2}$-approximation for MAX SMTI (man-oriented version)

- When a man is rejected by all women in his list, he is given a *second chance*

- For a man **m**, and for two women w_i and w_j, we say that **m** prefers w_i to w_j if
  1. either he prefers w_i in the usual sense
  2. or he is indifferent between the two, w_j is engaged and w_i is free.

- For a woman **w**, and for two men m_i and m_j, we say that **w** prefers m_i to m_j if
  1. either she prefers m_i in the usual sense
  2. or she is indifferent between the two, m_i has a second chance (he is proposing to the women in his list for the 2nd time) and m_j does not (he is proposing to the women in his list for the 1st time).

Kiraly's $\frac{3}{2}$-approximation for MAX SMTI (man-oriented version) contd.

- An unassigned man proposes to his most-preferred woman on his list, according to his new definition of *prefers*
- An unassigned woman always accepts a proposal (as was the case in GS)
- An assigned woman **w** accepts a new proposal from a man **m**, and rejects her current partner $\mathbf{m_k}$, if
  1. either she prefers **m** to her current partner, according to her new definition of *prefers*
  2. or her current partner prefers some woman to **w**, again according to his new definition of *prefers*. (In this case we call **w** *precarious*.)
- When a woman **w** rejects a man **m**, and she is not precarious, **m** and **w** are deleted from each others' lists

SMTI: stable matching (1)

m_1 : (w_1 w_2)

m_2 : w_1

m_3 : w_3 w_4

m_4 : w_3

w_1 : (m_1 m_2)

w_2 : m_1

w_3 : (m_3 m_4)

w_4 : m_3

$M = \{(m_1, w_1), (m_3, w_3)\}$ (size 2)

SMTI: stable matching (2)

m_1 : (w_1 w_2)
m_2 : w_1
m_3 : w_3 w_4
m_4 : w_3

w_1 : (m_1 m_2)
w_2 : m_1
w_3 : (m_3 m_4)
w_4 : m_3

$M = \{(m_1, w_2), (m_2, w_1), (m_3, w_4), (m_4, w_3)\}$ (size **4**)

Example: Kiraly's algorithm

$m_1 : (\cancel{w_1}\ w_2)$

$m_2 : w_1$

$m_3 : \cancel{w_3}\ w_4$

$m_4 : \cancel{w_3}$

$w_1 : (\cancel{m_1}\ m_2)$

$w_2 : m_1$

$w_3 : (\cancel{m_3}\ \cancel{m_4})$

$w_4 : m_3$

- w_1 is precarious: her current partner m_1 prefers another woman, w_2, according to his new definition of prefers.
- w_3 is not precarious and is indifferent between m_3 and m_4, even according to her new definition of prefers.
- m_4 is given a second chance.
- w_3 prefers m_4 to m_3, according to her new definition of prefers.

Example: Kiraly's algorithm

m_1 : (w_1 w_2)

m_2 : w_1

m_3 : w_3 w_4

m_4 : w_3

w_1 : (m_1 m_2)

w_2 : m_1

w_3 : (m_3 m_4)

w_4 : m_3

$M = \{(m_1, w_2), (m_2, w_1), (m_3, w_4), (m_4, w_3)\}$ (size **4**)

Tutorial Outline

1.1: Classical Hospitals / Residents problem

1.2: Hospitals / Residents problem with Ties

1.3: Hospitals / Residents problem with Couples

1.4: “Almost stable” matchings

1.5: Social Stability

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

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:

$$\begin{array}{rl} (r_1,r_2): & (h_1,h_2) \\ r_3: & h_1 \; h_2 \end{array} \qquad\qquad \begin{array}{rl} h_1:1: & r_1 \; r_3 \; r_2 \\ h_2:1: & r_1 \; r_3 \; r_2 \end{array}$$

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

- Stable matchings can have different sizes

Couples in HR

- 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

- 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
  - it cannot terminate if there is no stable matching
  - it need not terminate even if there is a stable matching

Algorithm C: example

Resident preferences

r_3 : $\cancel{h_1}$ h_5

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

(r_1, r_5) : (h_1, h_2) (h_3, h_6)

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

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

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

Resident preferences

r_3	:	h_1	h_5	
r_7	:	h_6	h_8	
(r_1, r_5)	:	(h_1, h_2)	(h_3, h_6)	
(r_2, r_4)	:	(h_4, h_5)	(h_1, h_2)	(h_3, h_7)
(r_6, r_8)	:	(h_6, h_8)		

Hospitals' preferences

r_1 r_2 r_3 r_4 r_5 r_6 r_7 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

- 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

Tutorial Outline

1.1: Classical Hospitals / Residents problem

1.2: Hospitals / Residents problem with Ties

1.3: Hospitals / Residents problem with Couples

1.4: "Almost stable" matchings

1.5: Social Stability

Maximum matchings vs stable matchings

- Maximum matchings can be twice the size of stable matchings

- Example (each hospital has capacity **1**):

$\mathbf{r_1}$:	$\mathbf{h_1}$	$\mathbf{h_2}$		$\mathbf{h_1}$:	$\mathbf{r_1}$	$\mathbf{r_2}$
$\mathbf{r_2}$:	$\mathbf{h_1}$			$\mathbf{h_2}$:	$\mathbf{r_1}$	

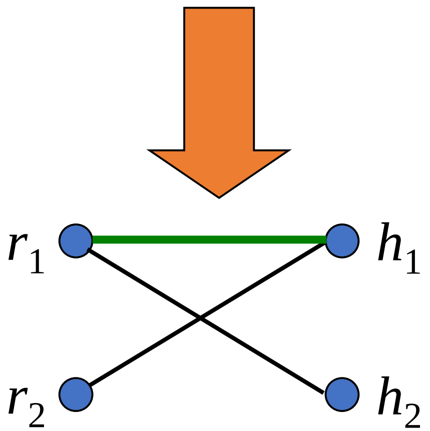
Maximum matchings vs stable matchings

- Maximum matchings can be twice the size of stable matchings

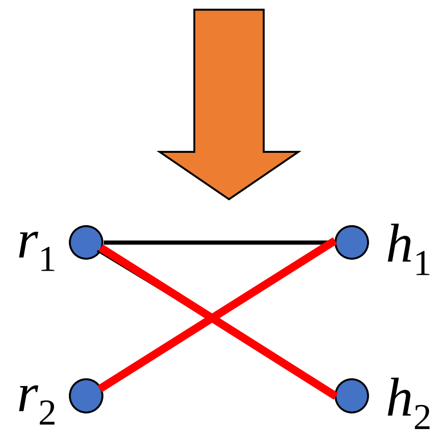
- Example (each hospital has capacity **1**):

$\mathbf{r}_1$: $\mathbf{h}_1$ $\mathbf{h}_2$	$\mathbf{h}_1$: $\mathbf{r}_1$ $\mathbf{r}_2$
$\mathbf{r}_2$: $\mathbf{h}_1$	$\mathbf{h}_2$: $\mathbf{r}_1$

$\mathbf{r}_1$: $\mathbf{h}_1$ $\mathbf{h}_2$	$\mathbf{h}_1$: $\mathbf{r}_1$ $\mathbf{r}_2$
$\mathbf{r}_2$: $\mathbf{h}_1$	$\mathbf{h}_2$: $\mathbf{r}_1$



stable matching



maximum matching

Maximum matchings vs stable matchings

- A small number of blocking pairs could be tolerated if it is possible to find a larger matching

- But, different maximum matchings can have different numbers of blocking pairs

- Example:
  (each hospital has capacity **1**)

r_1:	h_4	h_1	h_3	h_1:	r_4	r_1	r_2	
r_2:	h_2	h_1	h_4	h_2:	r_3	r_2	r_4	
r_3:	h_2	h_4	h_3	h_3:	r_1	r_3		
r_4:	h_1	h_4	h_2	h_4:	r_4	r_1	r_3	r_2

- Every stable matching has size **3**

Maximum matchings vs stable matchings

- A small number of blocking pairs could be tolerated if it is possible to find a larger matching
- But, different maximum matchings can have different numbers of blocking pairs
- Example:
(each hospital has capacity **1**)

r_1:	h_4	h_1	h_3		h_1:	r_4	r_1	r_2	
r_2:	h_2	h_1	h_4		h_2:	r_3	r_2	r_4	
r_3:	h_2	h_4	h_3		h_3:	r_1	r_3		
r_4:	h_1	h_4	h_2		h_4:	r_4	r_1	r_3	r_2

- Maximum matching $M_1=\{(r_1,h_1), (r_2,h_2), (r_3,h_3), (r_4,h_4)\}$
- Blocking pairs of M_1: (r_3,h_2), (r_4,h_1) **(2)**

Maximum matchings vs stable matchings

- A small number of blocking pairs could be tolerated if it is possible to find a larger matching

- But, different maximum matchings can have different numbers of blocking pairs

- Example:
(each hospital has capacity **1**)

r_1:	h_4	h_1	h_3		h_1:	r_4	r_1	r_2	
r_2:	h_2	h_1	h_4		h_2:	r_3	r_2	r_4	
r_3:	h_2	h_4	h_3		h_3:	r_1	r_3		
r_4:	h_1	h_4	h_2		h_4:	r_4	r_1	r_3	r_2

- Maximum matching $M_2=\{(r_1,h_1), (r_2,h_4), (r_3,h_3), (r_4,h_2)\}$

- Blocking pairs of M_2: (r_1,h_4), (r_2,h_2), (r_3,h_2), (r_3,h_4), (r_4,h_1), (r_4,h_4) **(6)**

Maximum matchings vs stable matchings

- A small number of blocking pairs could be tolerated if it is possible to find a larger matching

- But, different maximum matchings can have different numbers of blocking pairs

- Example:
  (each hospital has capacity **1**)

r_1:	h_4	h_1	h_3	h_1:	r_4	r_1	r_2	
r_2:	h_2	h_1	h_4	h_2:	r_3	r_2	r_4	
r_3:	h_2	h_4	h_3	h_3:	r_1	r_3		
r_4:	h_1	h_4	h_2	h_4:	r_4	r_1	r_3	r_2

- Maximum matching $M_3=\{(r_1,h_4), (r_2,h_2), (r_3,h_3), (r_4,h_1)\}$

- Blocking pairs of M_3: (r_3,h_2) **(1)**

“Almost stable” matchings

- Given an instance of HR, the problem is to find a maximum matching that is “almost stable”, i.e., admits the minimum number of blocking pairs

- The problem is:
  - NP-hard
    - even if every preference list is of length ≤ 3
  - not approximable within $n^{1-\varepsilon}$, for any $\varepsilon > 0$, unless P=NP, where n is the number of residents
  - solvable in polynomial time if each resident’s list is of length ≤ 2

- In all cases the result is true if each hospital has capacity $\mathbf{1}$

[Biro, Manlove and Mittal, 2010]

Tutorial Outline

1.1: Classical Hospitals / Residents problem

1.2: Hospitals / Residents problem with Ties

1.3: Hospitals / Residents problem with Couples

1.4: “Almost stable” matchings

1.5: Social Stability

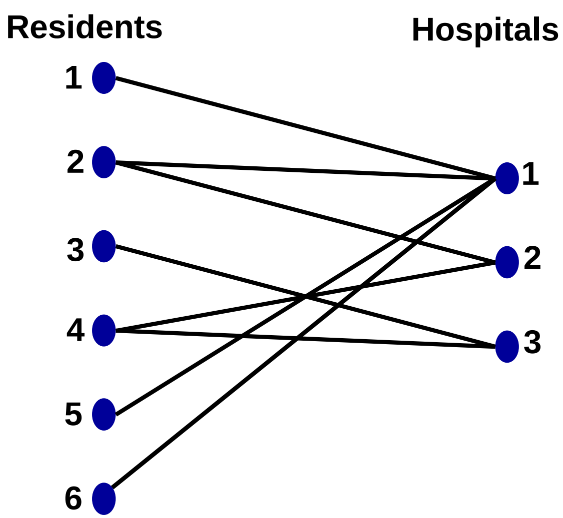
The Social Network Graph

- A blocking pair (r,h) of a matching M may not necessarily lead to M being undermined in practice
  - Especially if r and h are unaware of each other's preference list

- Consider an HR instance I augmented by a *social network graph*
  - A bipartite graph comprising a subset of the acceptable resident-hospital pairs that have some social ties

- A resident-hospital pair is *acquainted* if they form an edge in the social network graph, and *unacquainted* otherwise

- Unacquainted pairs cannot block a matching



Social network graph G

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

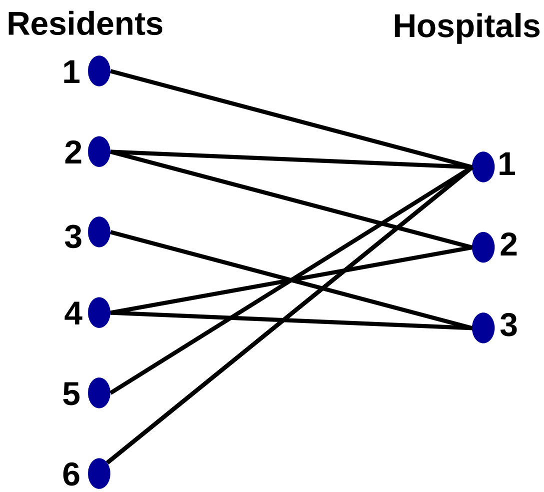
Each hospital has capacity **2**

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



Social network graph G

Unacquainted pairs: $\{(r_1,h_2), (r_3,h_1), (r_5,h_2)\}$

Example

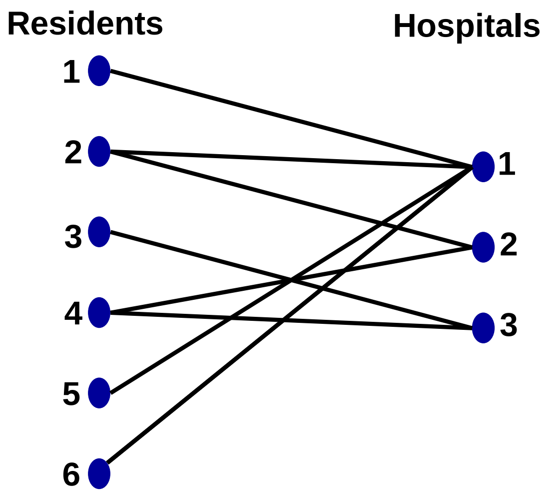
r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

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



Social network graph G

Unacquainted pairs: $\{(r_1,h_2), (r_3,h_1), (r_5,h_2)\}$

(r_3,h_1) is no longer allowed to block the matching

Social stability

- A pair (r,h) *socially blocks* a matching M if:
  - (r,h) blocks M in the classical sense
  - (r,h) is an acquainted pair

- M is *socially stable* if it has no social blocking pair

- An instance of the ***H**ospitals / **R**esidents problem under **S**ocial **S**tability* (*HRSS*) comprises an HR instance I and a social network graph G

- Given an HRSS instance (I,G), any stable matching in I is socially stable in (I,G)

Socially stable matchings of different sizes

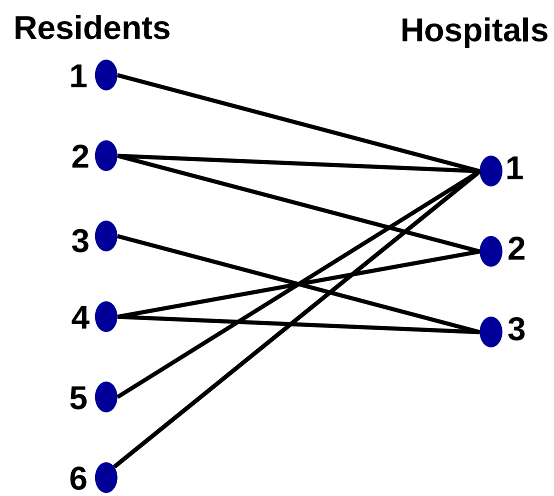
r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

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



Social network graph G

Socially stable matching of size **6**

Socially stable matchings of different sizes

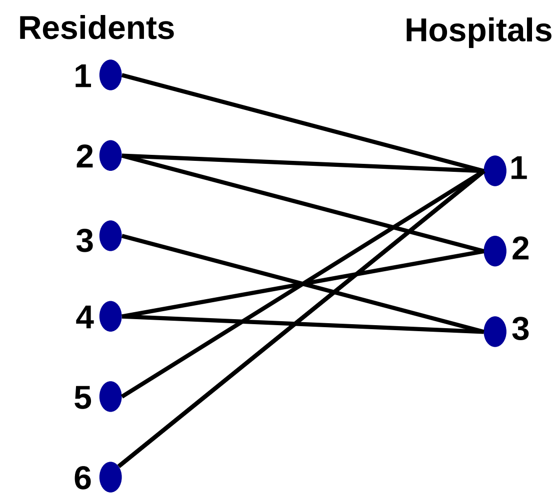
r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

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



Social network graph G

Socially stable matching of size 5

Algorithmic results

- The problem of finding a maximum socially stable matching, given an instance of HRSS, is:
  - NP-hard, even if all preference lists are of length **≤3** and each hospital has capacity **1**
  - solvable in polynomial-time if:
    - each resident's list is of length **≤2**, *or*
    - the number of acquainted pairs is constant, *or*
    - the number of unacquainted pairs is constant
  - approximable within **3/2**
  - not approximable better than **3/2** assuming the *Unique Games Conjecture*
  - [Askalidis, Immorlica, Kwanashie, M and Pountourakis, 2013]

What's next in the tutorial

2.1. Strategic agents

2.2. Integer Programming

2.3. Parameterised complexity

2.4. Preference elicitation

2.6: School Choice

Dominant-strategy truthfulness

- Agents (e.g. hospitals and residents) declare their preference lists to the centralised system (e.g. NRMP)
- Agent are strategic: they misreport if it is in their benefit
  - i.e. if providing a different ranking over candidates results in the system matching them with a better one
- A matching mechanism is *dominant-strategy truthful* (*DS truthful*), if every agent finds it in his/her best interest to declare his/her true preference list, no matter what other agents choose to do.

- Is Gale-Shapley DS truthful?

Is Gale-Shapley DS truthful?

Will all agents reveal their preferences truthfully?

- When the man-oriented version of Gale-Shapley algorithm is executed, all men find it in their best interest to be truthful.
- Some women, however, may benefit from misreporting their preferences.

In truth w_1 prefers m_2 to m_3

$m_1 : \cancel{w_2}\ w_1\ w_3$

$m_2 : \cancel{w_1}\ w_2\ w_3$

$m_3 : \cancel{w_1}\ \cancel{w_2}\ w_3$

$w_1 : m_1\ \cancel{m_3}\ \cancel{m_2}$

$w_2 : m_2\ \cancel{m_1}\ \cancel{m_3}$

$w_3 : m_1\ m_3\ m_2$

Is Gale-Shapley DS truthful?

Will all agents reveal their preferences truthfully?

- When the man-oriented version of Gale-Shapley algorithm is executed, all men find it in their best interest to be truthful.
- Some women, however, may benefit from misreporting their preferences.

In truth w_1 prefers m_2 to m_3

m_1 : w_2 w_1 w_3

m_2 : w_1 w_2 w_3

m_3 : w_1 w_2 w_3

w_1 : m_1 m_3 m_2

w_2 : m_2 m_1 m_3

w_3 : m_1 m_3 m_2

Is Gale-Shapley DS truthful?

- When the man-oriented version of Gale-Shapley algorithm is executed, all men find it in their best interest to be truthful.
- Some women, however, may benefit from misreporting their preferences.

$m_1 : w_2 \quad w_1 \quad w_3$

$m_2 : w_1 \quad w_2 \quad w_3$

$m_3 : w_1 \quad w_2 \quad w_3$

$w_1 : m_1 \quad m_3 \quad m_2$

$w_2 : m_2 \quad m_1 \quad m_3$

$w_3 : m_1 \quad m_3 \quad m_2$

- In fact, there is no mechanism that can always induce all agents to be truthful.

Theorem (Roth, 1982)

No stable matching mechanism exists for which truth-telling is a dominant strategy for every agent.

What’s next in the tutorial

2.1. Strategic agents

2.2. Integer Programming

2.3. Parameterised complexity

2.4. Preference elicitation

2.6: School Choice

Recall: Integer (Linear) Programming

- Integer programming: Objective function
  - min $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ subject to $A\mathbf{x}\leq\mathbf{b}$ Constraints

    Variables
  - where $\mathbf{c}=(c_1, c_2,\ldots, c_n)^{\mathrm{T}}$, $\mathbf{x}=(x_1, x_2,\ldots, x_n)^{\mathrm{T}}$, $\mathbf{b}=(b_1, b_2,\ldots, b_m)^{\mathrm{T}}$ $A=(a_{ij})$ $(1\leq i\leq m, 1\leq j\leq n)$, the c_i, a_{ij} and b_j are real-valued known coefficients and the x_i are integer-valued variables
- Linear programming: relaxation in which x_i are real-valued
  - solvable in polynomial time
- General integer programming problem is NP-hard
  - but there are some powerful solvers

Integer Programming for MAX HRT

- Model developed by Augustine Kwanashie (2012)
- Solved using CPLEX IP solver
- IP models of HRT instances with tie density of about 85% are the most likely to be computationally hard
- Real world SFAS (Scottish Foundation Allocation Scheme) datasets were also solved using the IP model
  - Ties only exist in hospitals' lists

Year	#Residents	#hospitals	Tie density	Matching Size	Runtime
2005/2006	759	53	92%	758	92.96 sec
2006/2007	781	53	76%	746	21.78 sec
2007/2008	748	52	81%	709	75.50 sec

Scottish Foundation Allocation Scheme

- Set of applicants and programmes (residents and hospitals)
- Up to 2012: each applicant
  - ranks **10** programmes in strict order of preference
  - has a *score* in the range **40..100**
- Two applicants can *link* their applications
  - preferences are interleaved in a precise way to form their *joint preference list*
  - only *compatible programmes* appear on joint preference list
- Each programme
  - has a *capacity* indicating the number of *posts* it has
  - has a preference list derived from the above scoring function
  - so *ties* are possible

Integer Programming for HRC

- Model developed by Iain McBride (2013)
- Solved using CPLEX IP solver
- Random instances, scalability (preference lists of length between 5 and 10):
  - 5000 residents, 500 hospitals, 500 couples, 5000 posts (x25)
    - solved in 99.6 seconds on average
  - 10000 residents, 1000 hospitals, 1000 couples, 10000 posts (x1)
    - solved in 10 minutes
- Random instances, solvability / sizes of largest stable matchings found:
  - 500 residents, 50 hospitals, 250 couples, 500 posts (x1000)
    - around 70% of instances were solvable
    - Average time taken 75s per instance
- SFAS instances:
  - 2012: 710 residents, stable matching of size 681 found in 16s
  - 2011: 736 residents, stable matching of size 688 found in 17s
  - 2010: 734 residents, stable matching of size 681 found in 65s

IP for “Almost stable” matchings in HRC

- Model developed by Iain McBride and James Trimble (2016)
- Solved using CPLEX IP solver
- A Constraint Programming (CP) model also developed
- Both of these models evaluated on 28,000 randomly generated instances of HRC
  - taking into account the fact that, in reality, some hospitals and residents are more popular than others
- Major findings
  - The CP model is about 1.15 times faster than the IP model
  - the number of blocking pairs admitted by a solution is very small, i.e., usually at most 1, and never more than 2

What’s next in the tutorial

2.1. Strategic agents

2.2. Integer Programming

2.3. Parameterised complexity

2.4. Preference elicitation

2.6: School Choice

Recall: parameterised complexity

- A parameterised problem with total input size **n** and parameter **k** is considered to be tractable if it can be solved by an algorithm whose running time is bounded by $\mathbf{f}(\boldsymbol{k}).\boldsymbol{n}^{\boldsymbol{O}(\mathbf{1})}$
  - **f** can be any computable function

- The problem is then said to be *fixed-parameter tractable* and belong to the class **FPT**

- And the algorithm is called an *fpt-algorithm*

Parameterised complexity of MAX SMTI

- Studied by Marx and Schlotter [2010]
- Three parameters were considered:
  - the number of ties
  - the maximum length of a tie
  - the total length of the ties
- MAX SMTI is
  - in **FPT** when parameterised by the total length of the ties
  - W[1]-hard when parameterised by the number of ties, even if all the men have strictly ordered preference lists.
- If W[1]$\neq$ FPT, there is no FPT local search algorithm for MAX SMTI with parameterisation l, the size of the neighbourhood to be searched, even if the maximum length of a tie is 2 and ties occur in the women's lists only.

Parameterised complexity of MAX HRC

- MAX HRC is the problem of finding a maximum cardinality stable matching, or reporting that none exists, in an instance of HRC.

- Studied by Marx and Schlotter [2011]

- If W[1]$\neq$ FPT, there is no FPT local search algorithm for MAX SMTI with parameterisation **l**, the size of the neighbourhood to be searched, even if each hospital has capacity 1.

- If the problem is parameterised by both **l** and **$|R_c|$**, the number of couples, then there is an FPT local search algorithm (with no assumption on the hospital capacities).

What's next in the tutorial

2.1. Strategic agents

2.2. Integer Linear Programming

2.3. Parameterised complexity

2.4. Preference elicitation

2.6: School Choice

Preference elicitation (via interviews)

- Agents don’t always know their preferences
  - Specially in large markets such as hospitals/residents problem, or academic hiring

- They usually start by having some rough ideas about their preferences
  - e.g. residents mentally rank hospitals into top tier, second tier, and so on.

- And then use interviews to help them refine their preferences

Interview minimisation

- Interviews are costly so it’s generally a good idea to minimise their numbers
- R. et al. [2013] investigated the design of a centralised system that schedules/recommends interviews (an interview scheduling policy)
  - with the goal of minimising the total number of interviews (an optimal policy)
  - while ensuring a matching stable w.r.t. the true underlying, albeit unknown, preferences of the agents can be found
- They considered three different optimality criteria
- Their results prove or suggest that
  - finding an optimal policy is NP-hard
  - but if agents’ preferences are correlated, then we can execute an optimal policy in polynomial time

Pairwise comparison queries

- Drummond and Boutilier [2013] studied a setting where agents use pairwise comparison queries to refine their preferences
- They proposed a method for finding approximately stable matchings, using minimax regret as a measure, while keeping the number of required comparisons relatively low.
- In Drummond and Boutilier [2014] the authors introduced a unified model where both comparison queries and interviews (together) can be used to refine preferences.
- They provided a polynomial-time policy for generating queries and interviews, and examined the effectiveness of their policy via empirical evaluation including comparison against the polynomial-time algorithm of R. et al. [13] for the restricted setting in which participants on one side of the market have the same partially ordered preferences.

What's next in the tutorial

2.1. Strategic agents

2.2. Integer Linear Programming

2.3. Parameterised complexity

2.4. Preference elicitation

2.5: School Choice

School choice

Background

- Historically, children have gone to neighborhood schools
- More recently: several countries have adopted school choice programs, allowing parents additional flexibility and creating competition between schools.

Goal: designing mechanism that

- produces a Pareto efficient assignment of students to schools,
- has a fair procedure and outcomes, and
- is easy to understand and use (and can be implemented in polynomial time).

School choice Model

- Similar to HR
  - Schools replace hospitals
  - Students replace residents

- Schools’ preferences are referred to as priorities
  - Sibling in the school
  - Distance
  - entrance exam results

- Priorities are usually course (ties exist)

Pareto efficiency

- An assignment of students to schools is *Pareto efficient* if it is impossible to reassign the students so as to
  - make no students worse off and
  - make at least one student better off.
- In this example assume that each school/college has capacity **1**
- Red matching is not Pareto efficient: $\{(s_1, c_1), (s_2, c_3), (s_3, c_2)\}$
- $\mathbf{s_1}$ and $\mathbf{s_3}$ prefer the green matching: $\{(s_1, c_2), (s_2, c_3), (s_3, c_1)\}$

s_1: c_2 c_1	c_1: s_1 s_2 s_3
s_2: c_1 c_2 c_3	c_2: s_3 s_1 s_2
s_3: c_1 c_2	c_3: s_2 s_4
s_4: c_3	

Fairness and stability

- We already know that it is important to require stability, so as to avoid (student,school) pairs undermining the prescribed matching

- Requiring stability can be viewed as a way of enforcing fairness
    - No student can be forced to attend a school they don't want to attend, and no school can be forced to take a student they view as unqualified
    - There is no justified envy. That is, there is no student **s** who is assigned to a school they prefer less than **c**, only to see a student with lower priority end up at **c**

Boston Mechanism

- Let **k** denote the length of the longest preference list among students.

Boston mechanism

Assign all students and school to be free;

For (i = 1 to k) {

For each school $c \in C$ that is undersubscribed, assign seats in c to the not-yet-assigned students that have ranked it i 'th, according to c's priority ordering;

}

s_1: c_2 c_1

s_2: c_1 c_2 c_3

s_3: $\cancel{c_1}$ $\cancel{c_2}$

s_4: c_3

c_1: s_1 s_2 $\cancel{s_3}$

c_2: $\cancel{s_3}$ s_1 s_2

c_3: s_2 s_4

Each school has capacity **1**

Boston mechanism is not DS truthful

- If you don't put your priority school high on your rank list, you may lose it!
- Example: you want school c_1 most and c_2 second. You have high priority at c_2 but not c_1. Both are in high demand so to get in to either, you need to rank it first and have high priority. It will be best to rank c_2 first.
  - s_3 is unassigned when reporting truth, but is assigned c_2 when ranking it first

s_1: $\cancel{c_2}$ $\cancel{c_1}$	c_1: $\cancel{s_1}$ s_2 $\cancel{s_3}$
s_2: c_1 c_2 c_3	c_2: $\cancel{s_3}$ $\cancel{s_1}$ s_2
s_3: $\cancel{c_2}$ $\cancel{c_1}$	c_3: s_2 s_4
s_4: c_3	

Each school has capacity **1**

Boston mechanism is not always fair

- The resulting assignment of Boston mechanism may be unstable.
- Boston mechanism generates the red matching: $\{(s_1, c_2), (s_2, c_1), (s_4, c_3)\}$
- But (s_3, c_2) block this matching
  - s_3 prefers c_2 to be unassigned
  - c_2 prefers s_3 to s_1

s_1: c_2 c_1

s_2: c_1 c_2 c_3

s_3: c_1 c_2

s_4: c_3

c_1: s_1 s_2 s_3

c_2: s_3 s_1 s_2

c_3: s_2 s_4

Each school has capacity **1**

Student-oriented Gale-Shapley

- It is dominant-strategy truthful for students
  - usually safe to assume that schools priorities are clearly stated and known
- It produces a stable matching and hence is fair (according to our definition of fairness)
- But it can be inefficient!
- For this example, SGS produces the red matching
- But s_1 and s_3 prefer the green matching

s_1 : c_2 c_1

s_2 : c_1 c_2 c_3

s_3 : c_1 c_2

s_4 : c_3

c_1 : s_1 s_2 s_3

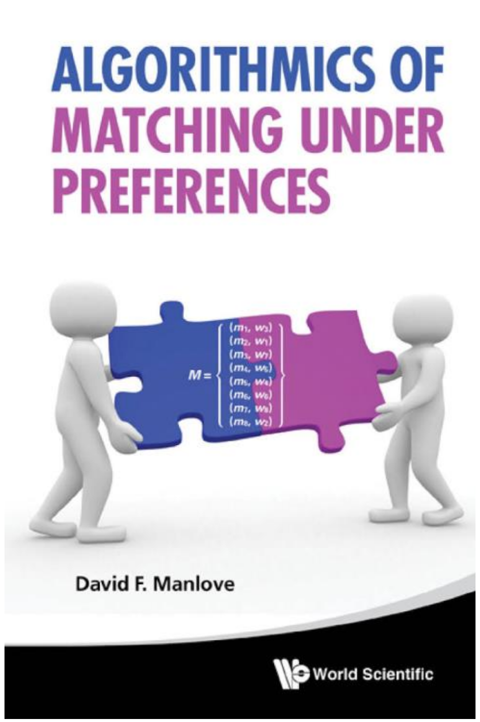
c_2 : s_3 s_1 s_2

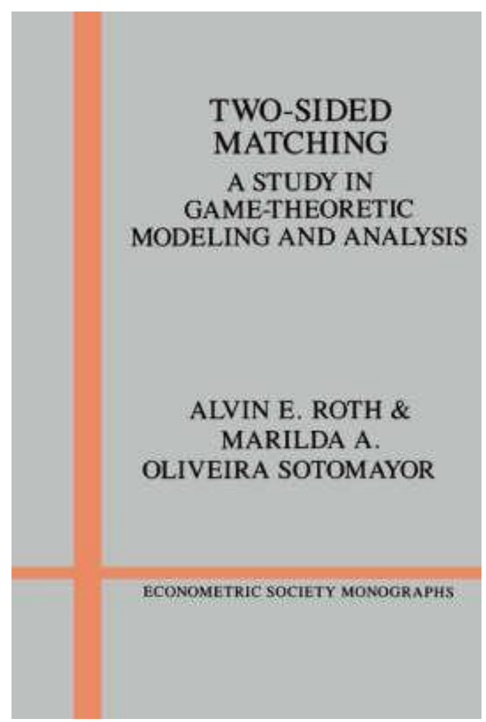
c_3 : s_2 s_4

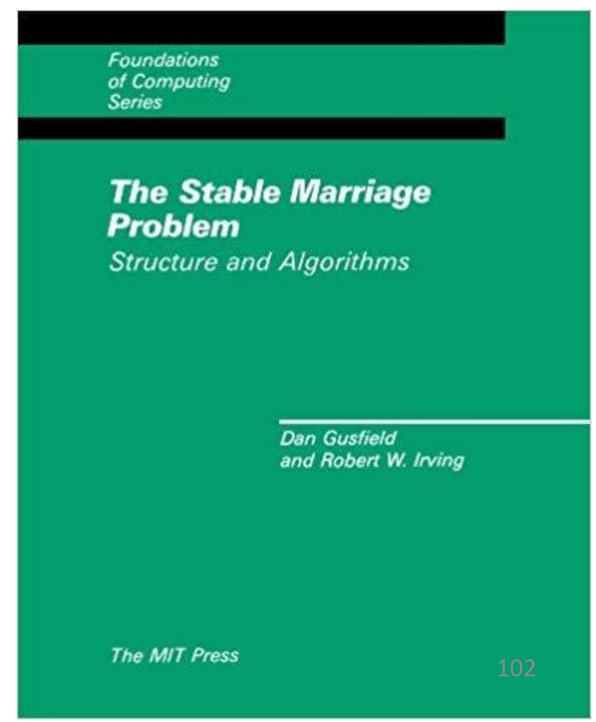
Each school has capacity 1

Books

- **The Stable Marriage Problem: Structure and Algorithms**, by D. Gusfield & R.W. Irving, MIT Press, Boston, Ma., 1989.
- **Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis**, by A. E. Roth, and M. Sotomayor. Cambridge University Press, 1990.
- **Algorithmics of Matching under Preferences** , by David Manlove, World Scientific, 2013.







Thank You!

References

Abdulkadiroğlu, A., Pathak, P.A. and Roth, A.E. (2005). The Boston public school match, American Economic Review 95, 2, pp. 368–371

Abdulkadiroğlu, A., Pathak, P.A. and Roth, A.E. (2005). The New York City high school match, American Economic Review 95, 2, pp. 364–367

Abdulkadiroğlu, A., Pathak, P.A. and Roth, A.E. (2009). Strategyproofness versus efficiency in matching with indifferences: Redesigning the NYC high school match, American Economic Review 99, 5, pp. 1954–1978

Abdulkadiroğlu, A., Pathak, P.A., Roth, A.E. and Sönmez, T. (2006). Changing the Boston school-choice mechanism, NBER working paper 11965

Askalidis, G., Immorlica, I., Kwanashie, A., Manlove, D.F., Pountourakis, E. (2013). Socially Stable matchings in the Hospitals / Residents problem. In Proceedings of WADS 2013: the 13th Algorithms and Data Structures Symposium, Lecture Notes in Computer Science, Springer, 2013

References

Biró, P., Irving, R.W. and Schlotter, I. (2011). Stable matching with couples: an empirical study, ACM Journal of Experimental Algorithmics 16, section 1, article 2, 27 pages

Biró, P., Manlove, D.F. and Mittal, S. (2010). Size versus stability in the Marriage problem. Theoretical Computer Science 411, pp. 1828-1841

Conway, J.H. (1976). Personal communication, reported in Knuth, D.E. (1976). Mariages Stables (Les Presses de L'Université de Montréal). English translation in Stable Marriage and its Relation to Other Combinatorial Problems, volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997

Drummond, J., and Boutilier, C. (2014). Preference Elicitation and Interview Minimization in Stable Matchings, Proc. of the Twenty-eighth Conference on Artificial Intelligence (AAAI-14)

Drummond, J., and Boutilier, C. (2013). Elicitation and Approximately Stable Matching with Partial Preferences, Proc. of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI-13)

References

Gale, D. and Shapley, L.S. (1962). College admissions and the stability of marriage, American Mathematical Monthly 69, pp. 9–15

Gale, D. and Sotomayor, M. (1985). Ms. Machiavelli and the stable matching problem, American Mathematical Monthly 92, 4, pp. 261–268

Gale, D. and Sotomayor, M. (1985). Some remarks on the stable matching problem, Discrete Applied Mathematics 11, pp. 223–232

Gusfield, D. and Irving, R.W. (1989). The Stable Marriage Problem: Structure and Algorithms (MIT Press)

Immorlica, N. and Mahdian, M. (2005). Marriage, honesty and stability, in Proceedings of SODA ’05: the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (ACM-SIAM), pp. 53–62

Irving, R.W. and Manlove, D.F. (2009). Finding large stable matchings, ACM Journal of Experimental Algorithmics 14, section 1, article 2, 30 pages

References

Irving, R.W., Manlove, D.F. and O'Malley, G. (2009). Stable marriage with ties and bounded length preference lists, Journal of Discrete Algorithms 7, 2, pp. 213–219

Irving, R.W., Manlove, D.F. and Scott, S. (2008). The stable marriage problem with master preference lists, Discrete Applied Mathematics 156, 15, pp. 2959–2977

Iwama, K., Manlove, D., Miyazaki, S. and Morita, Y. (1999). Stable marriage with incomplete lists and ties, in Proceedings of ICALP '99: the 26th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science, Vol. 1644 (Springer), pp. 443–452

Király, Z. (2012). Linear time local approximation algorithm for maximum stable marriage, in Proceedings of MATCH-UP '12: the 2nd International Workshop on Matching Under Preferences, pp. 99–110

Kobayashi, H. and Matsui, T. (2010). Cheating strategies for the Gale-Shapley algorithm with complete preference lists, Algorithmica 58, 1, pp. 151–169

References

Manlove, D.F., Irving, R.W., Iwama, K., Miyazaki, S. and Morita, Y. (1999). Hard variants of stable marriage, Tech. Rep. TR-1999-43, University of Glasgow, School of Comp. Science

Manlove, D.F., Irving, R.W., Iwama, K., Miyazaki, S. and Morita, Y. (2002). Hard variants of stable marriage, Theoretical Computer Science 276, 1-2, pp. 261–279

Manlove, D.F. and McBride, I. (2013). The Hospitals / Residents problem with Couples, Unpublished manuscript

Marx, D. and Schlotter, I. (2010). Parameterized complexity and local search approaches for the stable marriage problem with ties, Algorithmica 58, 1, pp. 170–187.

Marx, D. and Schlotter, I. (2011). Stable assignment with couples: parameterized complexity and local search, Discrete Optimization 8, pp. 25–40.

McDermid, E. (2009). A 3/2 approximation algorithm for general stable marriage, in Proceedings of ICALP ’09: the 36th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, Vol. 5555 (Springer), pp. 689–700

References

McDermid, E.J. and Manlove, D.F. (2010). Keeping partners together: Algorithmic results for the hospitals / residents problem with couples, Journal of Combinatorial Optimization 19, 3, pp. 279–303

Paluch, K. (2012). Faster and simpler approximation of stable matchings, in Proceedings of WAOA ’11: 9th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science, Vol. 7164 (Springer), pp. 176–187

Pini, M.S., Rossi, F., Venable, K.B. and Walsh, T. (2011). Manipulation complexity and gender neutrality in stable marriage procedures, Autonomous Agents and Multi-Agent Systems 22, 1, pp. 183–199

Rastegari, B., Condon, A., Immorlica, N., Irving, R.W., Leyton-Brown, K. (2014). Reasoning about Optimal Stable Matching under Partial Information, In the Proceedings of the Fifteenth ACM Conference on Economics and Computation (EC’14)

References

Rastegari, B., Condon, A., Immorlica, N., Leyton-Brown, K.(2013). Two-sided Matching with Partial Information. In the Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC’13)

Rastegari, B., Goldberg, P., Manlove, D. (2016). Preference Elicitation in Matching Markets via Interviews: A Study of Offline Benchmarks. In the Proceedings of the Fifteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS’16)

Ronn, E. (1990). NP-complete stable matching problems, Journal of Algorithms 11, pp. 285–304

Roth, A.E. (1982). The economics of matching: Stability and incentives, Mathematics of Operations Research 7, 4, pp. 617–628

Roth, A.E. (1982a). Incentive compatibility in a market with indivisible goods, Economics Letters 9, pp. 127–132

References

Roth, A.E. (1984). The evolution of the labor market for medical interns and residents: a case study in game theory, Journal of Political Economy 92, 6, pp. 991–1016

Roth, A.E. (1986). On the allocation of residents to rural hospitals: a general property of two-sided matching markets, Econometrica 54, pp. 425–427

Teo, C.-P., Sethuraman, J. and Tan, W.-P. (1999). Gale-Shapley stable marriage problem revisited: strategic issues and applications, Management Science 47, 9, pp. 1252–1267

Yanagisawa, H. (2007). Approximation Algorithms for Stable Marriage Problems, Ph.D. thesis, Kyoto University, School of Informatics

Teo, C.-P., Sethuraman, J. and Tan, W.-P. (1999). Gale-Shapley stable marriage problem revisited: strategic issues and applications, Management Science 47, 9, pp. 1252–1267

Yanagisawa, H. (2007). Approximation Algorithms for Stable Marriage Problems, Ph.D. thesis, Kyoto University, School of Informatics