University of Glasgow

Summer School on Matching Problems, Markets and Mechanisms

David Manlove

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

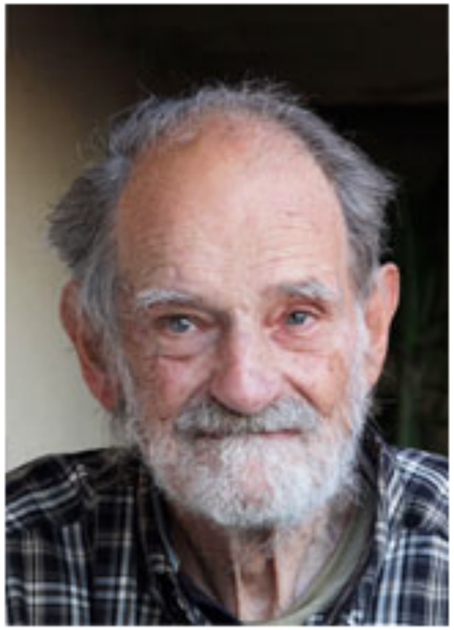


Photo: AP Photo/Reed Saxon

Lloyd S. Shapley

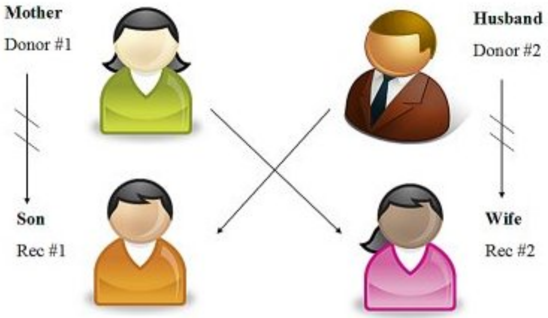
The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012 was awarded jointly to Alvin E. Roth and Lloyd S. Shapley *"for the theory of stable allocations and the practice of market design"*

Outline

1. The Hospitals / Residents problem and its variants

2. The House Allocation problem

3. Kidney exchange



The Hospitals / Residents problem and its variants

with applications to Junior Doctor Allocation

Primer: computational complexity (1)

- 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 $O(n^c)$ for some constant c, 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 has a *polynomial-time algorithm*
- If a decision problem is *NP-complete* it has no polynomial-time algorithm unless **P=NP**

Primer: computational complexity (2)

- 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

Centralised matching schemes

- Intending junior doctors 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)
  - Scotland (Scottish Foundation Allocation Scheme)
    - typically 700-750 applicants and 50 hospitals
- Stability is the key property of a matching
  - [Roth, 1984]

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

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

Hospitals / Residents problem (HR)

- Underlying theoretical model: Hospitals / Residents problem (HR)
- We have 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 instance

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

$M = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_3), (r_6, h_2)\}$ (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-oriented Gale-Shapley algorithm

```
M = ∅;
while (some resident r_i is unmatched and has a non-empty list)
{   r_i applies to the first hospital h_j on his 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 list;
            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 : 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

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

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

Hospitals / Residents problem with Ties

- In practice, residents' preference lists are short
- Hospitals' lists are generally long, so *ties* may be used – *Hospitals / Residents problem with Ties* (*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 [M 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, M et al, 1999], even if (simultaneously):
  - each hospital has capacity **1** (Stable Marriage problem with Ties and Incomplete Lists)
  - 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** [M et al, 2002]
    - *or* each preference list is of length **≤3** [Irving, M, O'Malley, 2009]
- Minimisation problem is NP-hard too, for similar restrictions! [M et al, 2002]

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]

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 [Irving and M, 2009]

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
- Figure below shows median computation times for increasing sizes of 10 HRT instances each with 85% tie density (all preference lists of length 5)

#Residents	#hospitals	Median Matching Size	Median Runtime
450	31	450	11.82 sec
500	35	500	31.20 sec
550	38	550	22.10 sec
600	42	600	44.15 sec
650	45	650	84.41 sec

- Real world SFAS datasets were also solved using the IP model.

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

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 M, 2010; Biró et al, 2011]
- Gives the *Hospitals / Residents problem with Couples* (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 M, 2010; Biró et al, 2011]
- Gives the *Hospitals / Residents problem with Couples* (HRC)
- A stable matching need not exist:

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

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

- Stable matchings can have different sizes

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 M, 2010; Biró et al, 2011]
- Gives the *Hospitals / Residents problem with Couples* (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

- 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**

  [M 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"

    [M 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 : 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)

cycle

Hospital 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**

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)		

Hospital preferences

r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8

Each hospital has capacity 1

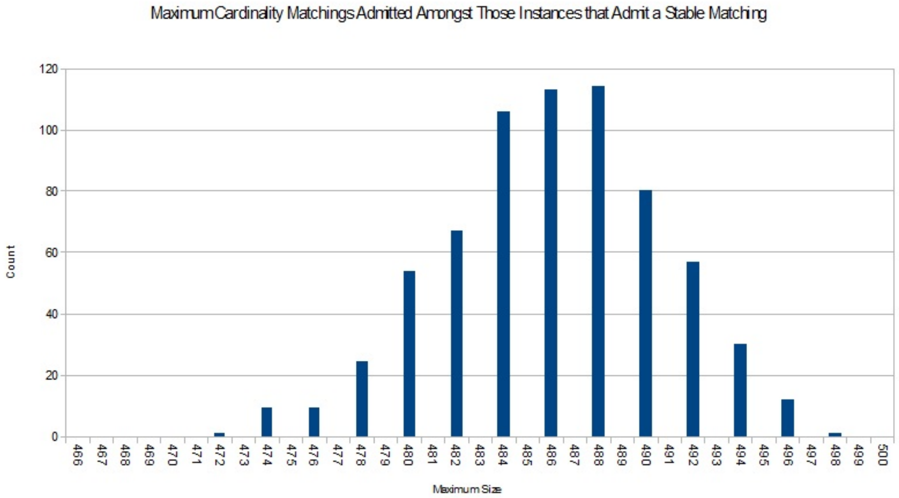
Stable matching: $M = \{(r_1, h_3), (r_2, h_1), (r_3, h_5), (r_4, h_2), (r_5, h_6), (r_7, h_8)\}$

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

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

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

The outcome

- Round 1
  - **710** applicants
  - **52** programmes with a total of **720** posts
  - **17** linked pairs
  - Stable matching found
  - Solution found matched **683** applicants, including all linked pairs

- Round 2
  - **27** applicants
  - **37** posts remaining at **10** programmes
  - No linked pairs
  - Applicants ranked all remaining programmes
  - Stable matching found
  - Solution found matched all remaining applicants

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

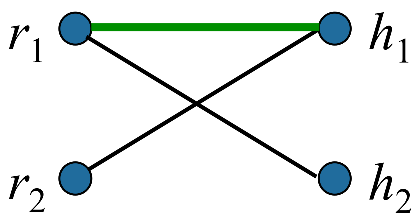
- Example (each hospital has capacity **1**):

r_1: h_1 h_2	h_1: r_1 r_2
r_2: h_1	h_2: r_1

Maximum matchings vs stable matchings

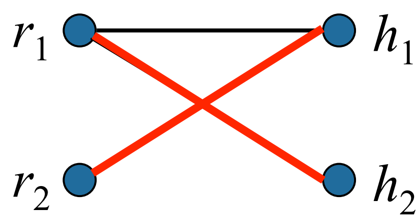
- Maximum matchings can be twice the size of stable matchings
- Example (each hospital has capacity **1**):

r_1 : h_1 h_2 $\quad$ h_1 : r_1 r_2
r_2 : h_1 $\quad$ h_2 : r_1



stable matching

r_1 : h_1 h_2 $\quad$ h_1 : r_1 r_2
r_2 : h_1 $\quad$ h_2 : r_1



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**)
- 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
r_2:	h_2	h_1	h_4
r_3:	h_2	h_4	h_3
r_4:	h_1	h_4	h_2

h_1:	r_4	r_1	r_2	
h_2:	r_3	r_2	r_4	
h_3:	r_1	r_3		
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 **1**

- [Biro, M and Mittal, 2010]

1.1: Classical Hospitals / Residents problem

1.2: Hospitals / Residents problem with Ties

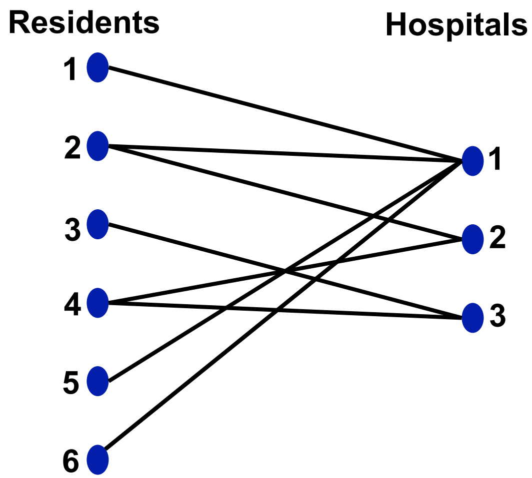
1.3: Hospitals / Residents problem with Couples

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

- 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

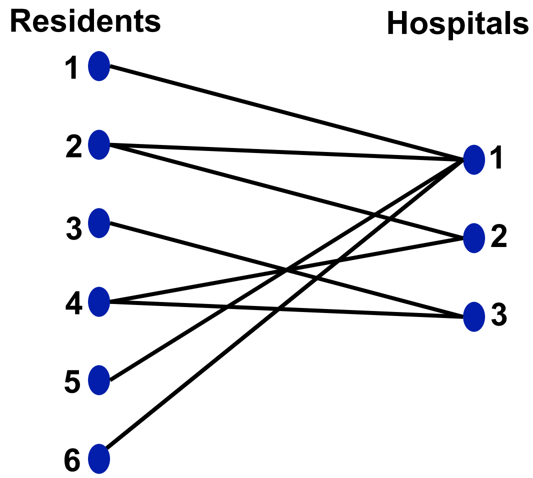
Each hospital has capacity **2**

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

University of Glasgow

Example

- Example:

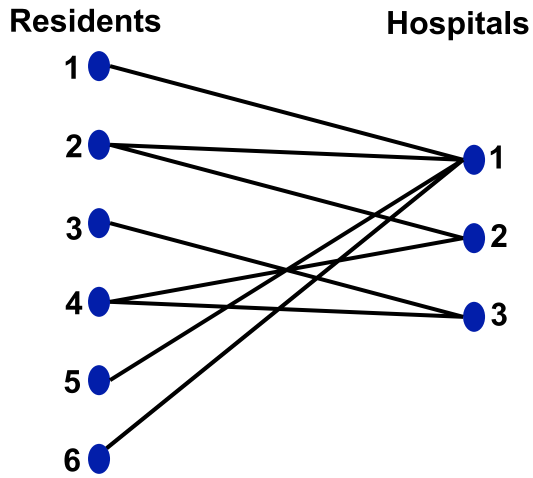
r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

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 *Hospitals / Residents problem under Social Stability* (*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

- Example:

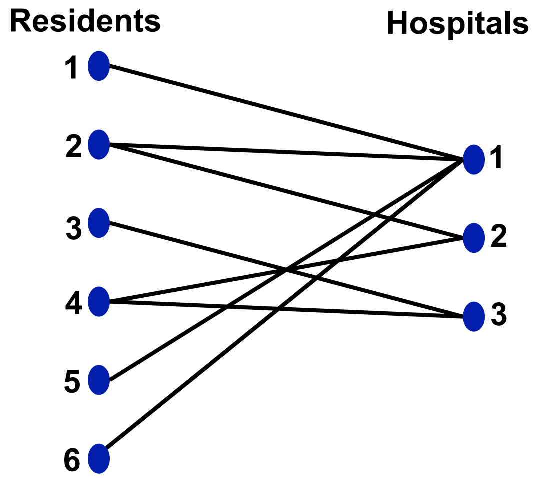
r_1: h_2 h_1
r_2: h_1 h_2
r_3: h_1 h_3
r_4: h_2 h_3
r_5: h_2 h_1
r_6: h_1 h_2

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*

- Socially stable matching of size 6

Socially stable matchings of different sizes

- Example:

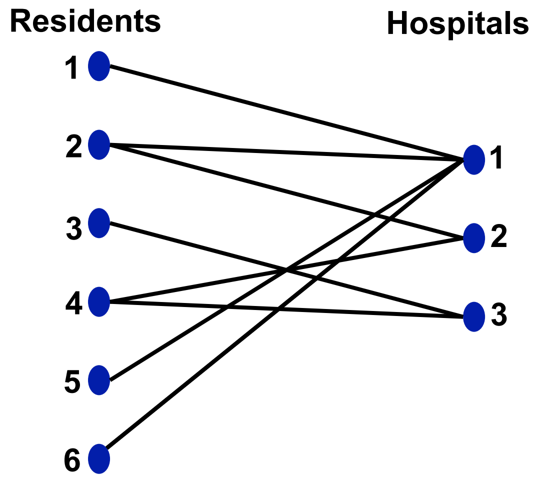
r_1 : h_2 h_1
r_2 : h_1 h_2
r_3 : h_1 h_3
r_4 : h_2 h_3
r_5 : h_2 h_1
r_6 : h_1 h_2

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***

- 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]

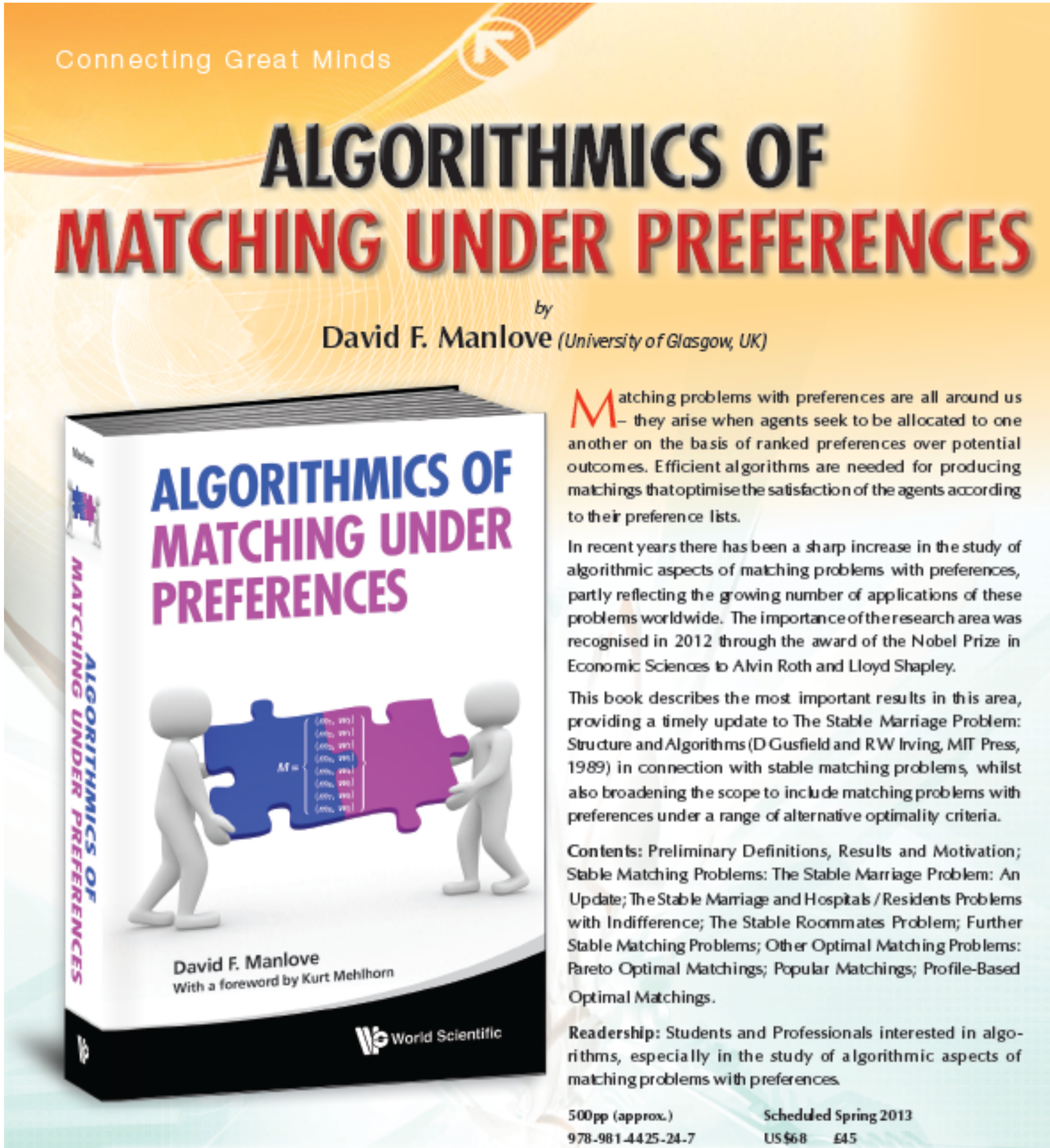
Open problems

- Approximation algorithm for MAX HRT with performance guarantee < **3/2**?
  - consider special cases:
    - ties on one side only
    - master lists

- To cope with the complexity of HRC, try to find a matching that is "as stable as possible"
  - one possibility: find a matching with the minimum number of blocking pairs
    - problem is NP-hard
    - approximability is open

- Acknowledgement: thanks to Iain McBride and Augustine Kwanashie

Further reading



- Chapters 3, 5

References

Abraham, D.J., Blum, A. and Sandholm, T. (2007). Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges, in Proceedings of EC '07: the 8th ACM Conference on Electronic Commerce (ACM), pp. 295–304

Abraham, D.J., Cechlárová, K., Manlove, D.F. and Mehlhorn, K. (2004). Pareto optimality in house allocation problems, in Proceedings of ISAAC '04: the 15th Annual International Symposium on Algorithms and Computation, Lecture Notes in Computer Science, Vol. 3341 (Springer), pp. 3–15

Abraham, D.J., Chen, N., Kumar, V. and Mirrokni, V.S. (2006). Assignment problems in rental markets, in Proceedings of WINE '06: the 2nd International Workshop on Internet and Network Economics, Lecture Notes in Computer Science, Vol. 4286 (Springer), pp. 198–213

Abraham, D.J., Irving, R.W., Kavitha, T. and Mehlhorn, K. (2005). Popular matchings, in Proceedings of SODA '05: the 16th ACM-SIAM Symposium on Discrete Algorithms (ACM-SIAM), pp. 424–432

Ashlagi, I., Fischer, F., Kash, I. and Procaccia, A. D. (2010). Mix and match, in Proceedings of EC '10: the 11th ACM Conference on Electronic Commerce (ACM), pp. 305–314

References

Ashlagi, I. and Roth, A. (2011). Individual rationality and participation in large scale, multi-hospital kidney exchange, in Proceedings of EC’11: the 12th ACM Conference on Electronic Commerce (ACM), pp. 321–322

Ashlagi, I. and Roth, A. (2012). New challenges in multihospital kidney exchange, American Economic Review 102, 3, pp. 354–359

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

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

Biró, P., Manlove, D.F. and Rizzi, R (2009). Maximum weight cycle packing in directed graphs, with application to kidney exchange, Discrete Mathematics, Algorithms and Applications 1, 4, pp. 499–517

References

Caragiannis, I., Filos-Ratsikas, A. and Procaccia, A. (2011). An improved 2-agent kidney exchange mechanism, in Proceedings of WINE ʼ11: the 7th International Workshop on Internet and Network Economics, Lecture Notes in Computer Science Series, vol. 7090 (Springer), pp. 37–48

Chen, Y. and Sönmez, T. (2002). Improving efficiency of on-campus housing: An experimental study, American Economic Review 92, 5, pp. 1669–1686

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

Dubins, L.E. and Freedman, D.A. (1981). Machiavelli and the Gale-Shapley algorithm, American Mathematical Monthly 88, 7, pp. 485–494

Gabow, H.N. and Tarjan, R.E. (1989). Faster scaling algorithms for network problems, SIAM Journal on Computing 18, pp. 1013–1036

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

References

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

Gärdenfors, P (1975). Match making: assignments based on bilateral preferences, Behavioural Science 20, pp. 166–173

Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K. and Mestre, J. (2010). Assigning papers to referees, Algorithmica 58, 1, pp. 119–136

Glorie, K.M., Klundert, J.J. van de and Wagelmans, A. (2013). Iterative branch-and-price for hierarchical multi-criteria kidney exchange. Econometric Institute Research Paper EI 2012-11, Erasmus University Rotterdam

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

Huang, C.-C. (2006). Cheating by men in the Gale-Shapley stable matching algorithm, in Proceedings of ESA'06: the 14th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, Vol. 4168 (Springer), pp. 418–431

References

Huang, C.-C. and Kavitha, T. (2012). Weight-maximal matchings, in Proceedings of MATCH-UP ' 12: the 2nd International Workshop on Matching Under Preferences, pp. 87–98

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. (1985). An efficient algorithm for the "stable roommates" problem, Journal of Algorithms, 6, pp. 577-595

Irving, R.W. (2007). Greedy and generous matchings via a variant of the Bellman-Ford algorithm, Unpublished manuscript

Irving, R.W., Kavitha, T., Mehlhorn, K., Michail, D. and Paluch, K. (2004). Rank-maximal matchings, in Proceedings of SODA ' 04: the 15th ACM-SIAM Symposium on Discrete Algorithms (ACM-SIAM), pp. 68–75

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

Keizer, K.M. , de Klerk, M., Haase-Kromwijk, B.J.J.M., and Weimar, W. (2005). The Dutch algorithm for allocation in living donor kidney exchange. Transplantation Proceedings, 37, pp. 589–591

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

University of Glasgow

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 Computing 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

Manlove, D.F. and O'Malley, G. (2012). Paired and Altruistic Kidney Donation in the UK. In Proceedings of SEA 2012: the 11th International Symposium on Experimental Algorithms, Lecture Notes in Computer Science, Vol. 7276 (Springer), pp. 271-282

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

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

References

Micali, S. and Vazirani, V.V. (1980). An O(√|V |·|E|) algorithm for finding maximum matching in general graphs, in Proceedings of FOCS '80: the 21st Annual IEEE Symposium on Foundations of Computer Science (IEEE Computer Society), pp. 17–27.

Ng, C. and Hirschberg, D.S. (1988). Complexity of the stable marriage and stable roommate problems in three dimensions, Tech. Rep. UCI-ICS 88-28, Department of Information and Computer Science, University of California, Irvine

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

Perach, N., Polak, J. and Rothblum, U.G. (2008). A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion, International Journal of Game Theory 36, 3-4, pp. 519–535

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

Rees, M.A., Kopke, J.E., Pelletier, R.P. et al. (2009). A nonsimultaneous, extended, altruistic-donor chain, New England Journal of Medicine, 360, pp. 1096–1101

References

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

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

Roth, A.E. and Postlewaite, A. (1977). Weak versus strong domination in a market with indivisible goods, Journal of Mathematical Economics 4, pp. 131–137

Roth, A.E., Sönmez, T. and Ünver M.U. (2004). Kidney exchange. Quarterly Journal of Economics, 119, 2, pp. 457–488

References

Roth, A.E., Sönmez, T. and Ünver., M.U. (2005). Pairwise kidney exchange. Journal of Economic Theory, 125, pp. 151–188

Roth, A.E., Sönmez, T. and Ünver., M.U. (2007). Efficient kidney exchange: Coincidence of wants in a market with compatibility-based preferences. American Economic Review, 97, 3, 828–851

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

Toulis, P. and Parkes, D. (2011). A random graph model of kidney exchanges: efficiency, individual rationality and incentives, in Proceedings of the 12th ACM conference on Electronic commerce (ACM), pp. 323–332

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

Yuan, Y. (1996). Residence exchange wanted: a stable residence exchange problem, European Journal of Operational Research 90, pp. 536–546