Computational aspects of matching problems under preferences (1st talk)

Péter Biró
Institute of Economics
Hungarian Academy of Sciences
peter.biro@krtk.mta.hu

Summer school on matchings
Moscow
5-8 October 2015

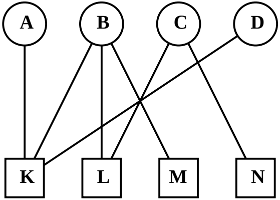
Matching without preferences...

Outline of the first part:

- introduction to matching theory
- basics of computational complexity
- chess pairings (FIDE rules)
- kidney exchange programs (UK experience)
- matching with couples

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...

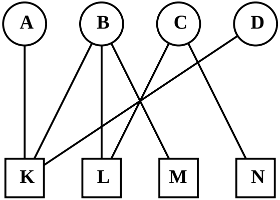


	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin:

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...

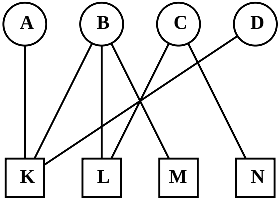


	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...

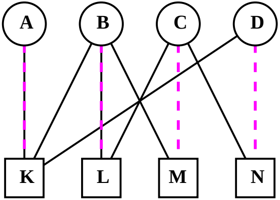


	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin:

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...



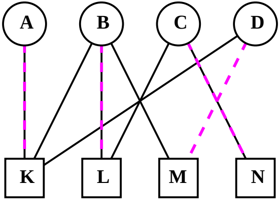
	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin:

Cannot he just try every possible combination?

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...

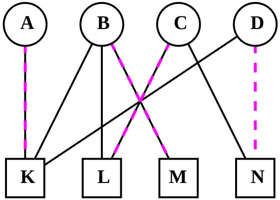


	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin:

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...



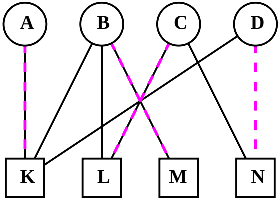
	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin:

This would be $4 * 3 * 2 * 1 = 4! = 24$ possibilities.

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...

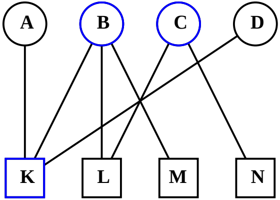


	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin:
But what if next time Arthur invites 100 men and 100 women?
(n! is more than the number of atoms in the universe for $n \geq 61$)

A tale on matchings...

Once upon a time, King Arthur wanted to organise a party. He invited four men and four women. He knew which of his invitees had known each other. He wanted to prepare a dance schedule where no man and woman are matched to each other if they have never met before. He asked Merlin the wizard to help...



	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

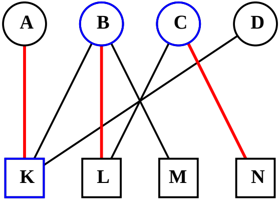
Arthur: Could you find me such a pairing?
Merlin: No, unfortunately not.
Arthur: Why? (tell me a good reason or you will be executed...)
Merlin: Since without B, C and K we have no more possible pair, so we cannot create more than three pairs.

The Kőnig theorem (1931)

Def: For a graph $G(N, E)$, a set of nodes $X \subset N$ is a vertex-cover if every edge in E is incident to some node in X.

For every bipartite graph,

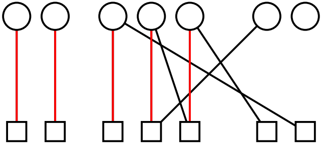
minimum size of a vertex-cover $=$ maximum size of a matching



	A	B	C	D
K	1	1	0	1
L	0	1	1	0
M	0	1	0	0
N	0	0	1	0

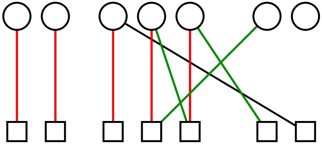
Proof of Kőnig's theorem

We keep looking for alternating paths from unmatched women to unmatched men...



Proof of Kőnig's theorem

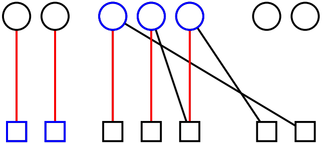
We keep looking for alternating paths from unmatched women to unmatched men...



- if we find one then we can enlarge the matching

Proof of Kőnig's theorem

We keep looking for alternating paths from unmatched women to unmatched men...

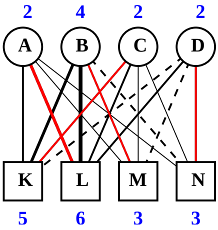


- if we find one then we can enlarge the matching
- if there is no augmenting path then we can find a vertex-cover of minimum size

Weighted and nonbipartite graphs: still tractable

Egerváry (1931): For every **weighted** bipartite graph,
minimum value of a cover = maximum weight of a matching

	A	B	C	D	
K	6	8	7	0	5
L	8	10	7	7	6
M	4	7	3	0	3
N	4	0	4	5	3
	2	4	2	2	

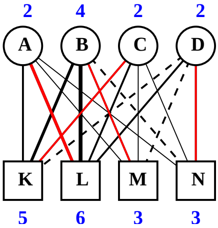


Kuhn (1955): A maximum weight matching can be found efficiently (in strongly polynomial time) by the Hungarian method.

Weighted and nonbipartite graphs: still tractable

Egerváry (1931): For every **weighted** bipartite graph,
minimum value of a cover = maximum weight of a matching

	A	B	C	D	
K	6	8	7	0	5
L	8	10	7	7	6
M	4	7	3	0	3
N	4	0	4	5	3
	2	4	2	2	

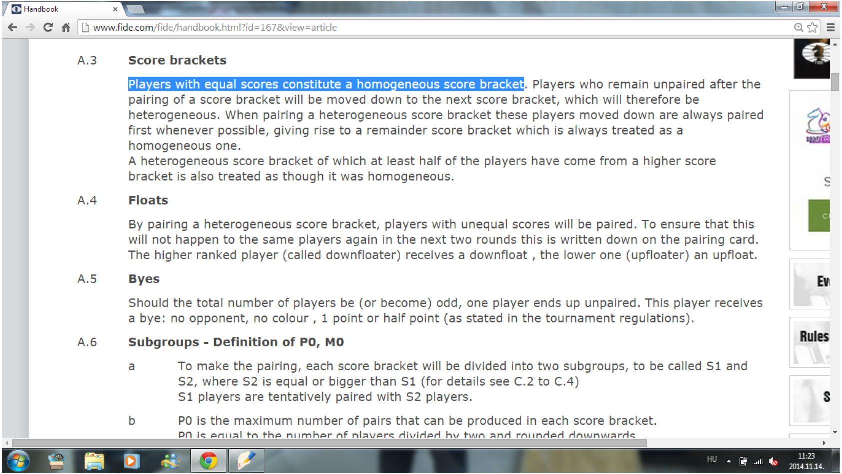


Kuhn (1955): A maximum weight matching can be found efficiently (in strongly polynomial time) by the Hungarian method.

Edmonds (1967): For nonbipartite graphs, finding a maximum size or maximum weight matching is solvable efficiently.

Example for brute force matching: chess pairing

Example for brute force matching: chess pairing



Handbook

www.fide.com/fide/handbook.html?id=167&view=article

A.3 **Score brackets**

Players with equal scores constitute a homogeneous score bracket. Players who remain unpaired after the pairing of a score bracket will be moved down to the next score bracket, which will therefore be heterogeneous. When pairing a heterogeneous score bracket these players moved down are always paired first whenever possible, giving rise to a remainder score bracket which is always treated as a homogeneous one.
A heterogeneous score bracket of which at least half of the players have come from a higher score bracket is also treated as though it was homogeneous.

A.4 **Floats**

By pairing a heterogeneous score bracket, players with unequal scores will be paired. To ensure that this will not happen to the same players again in the next two rounds this is written down on the pairing card. The higher ranked player (called downfloater) receives a downfloat , the lower one (upfloater) an upfloat.

A.5 **Byes**

Should the total number of players be (or become) odd, one player ends up unpaired. This player receives a bye: no opponent, no colour , 1 point or half point (as stated in the tournament regulations).

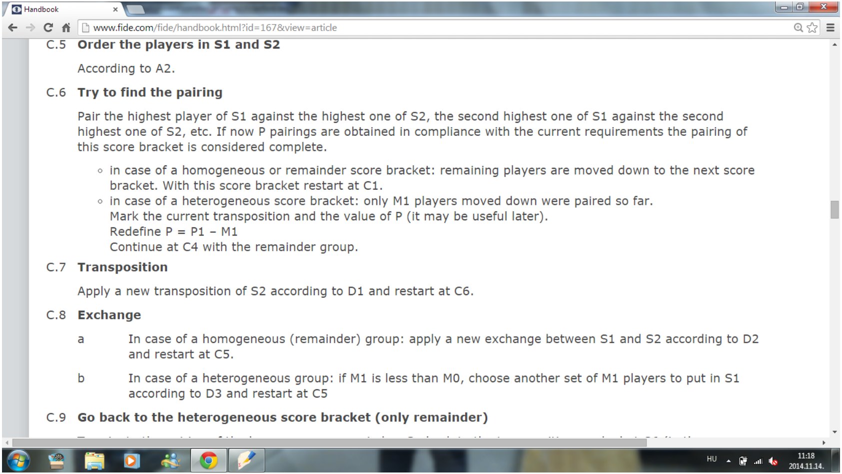
A.6 **Subgroups - Definition of P0, M0**

a To make the pairing, each score bracket will be divided into two subgroups, to be called S1 and S2, where S2 is equal or bigger than S1 (for details see C.2 to C.4)
S1 players are tentatively paired with S2 players.

b P0 is the maximum number of pairs that can be produced in each score bracket.
P0 is equal to the number of players divided by two and rounded downwards.

Dutch system

Example for brute force matching: chess pairing


Handbook

www.fide.com/fide/handbook.html?id=167&view=article

C.5 **Order the players in S1 and S2**

According to A2.

C.6 **Try to find the pairing**

Pair the highest player of S1 against the highest one of S2, the second highest one of S1 against the second highest one of S2, etc. If now P pairings are obtained in compliance with the current requirements the pairing of this score bracket is considered complete.

- in case of a homogeneous or remainder score bracket: remaining players are moved down to the next score bracket. With this score bracket restart at C1.
- in case of a heterogeneous score bracket: only M1 players moved down were paired so far.
  Mark the current transposition and the value of P (it may be useful later).
  Redefine P = P1 – M1
  Continue at C4 with the remainder group.

C.7 **Transposition**

Apply a new transposition of S2 according to D1 and restart at C6.

C.8 **Exchange**

a In case of a homogeneous (remainder) group: apply a new exchange between S1 and S2 according to D2 and restart at C5.

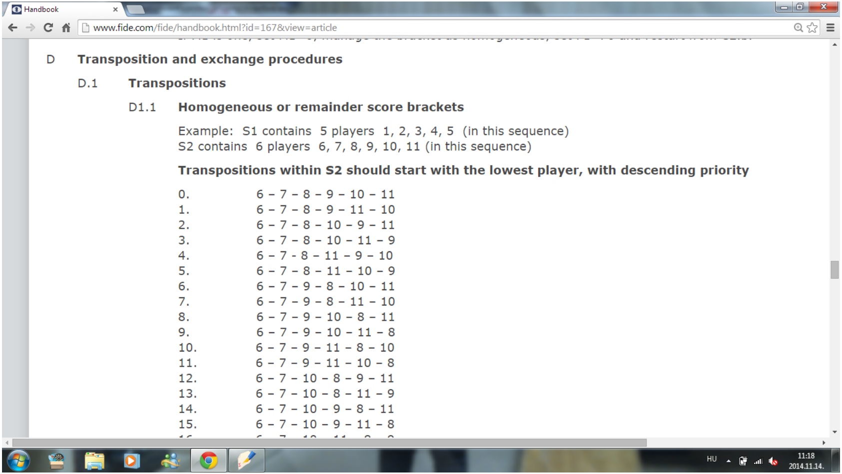
b In case of a heterogeneous group: if M1 is less than M0, choose another set of M1 players to put in S1 according to D3 and restart at C5

C.9 **Go back to the heterogeneous score bracket (only remainder)**

HU 11:18 2014.11.14.

Dutch system

Example for brute force matching: chess pairing



D Transposition and exchange procedures

D.1 Transpositions

D1.1 Homogeneous or remainder score brackets

Example: S1 contains 5 players 1, 2, 3, 4, 5 (in this sequence)
S2 contains 6 players 6, 7, 8, 9, 10, 11 (in this sequence)

Transpositions within S2 should start with the lowest player, with descending priority

0.	6 – 7 – 8 – 9 – 10 – 11
1.	6 – 7 – 8 – 9 – 11 – 10
2.	6 – 7 – 8 – 10 – 9 – 11
3.	6 – 7 – 8 – 10 – 11 – 9
4.	6 – 7 - 8 – 11 – 9 – 10
5.	6 – 7 – 8 – 11 – 10 – 9
6.	6 – 7 – 9 – 8 – 10 – 11
7.	6 – 7 – 9 – 8 – 11 – 10
8.	6 – 7 – 9 – 10 – 8 – 11
9.	6 – 7 – 9 – 10 – 11 – 8
10.	6 – 7 – 9 – 11 – 8 – 10
11.	6 – 7 – 9 – 11 – 10 – 8
12.	6 – 7 – 10 – 8 – 9 – 11
13.	6 – 7 – 10 – 8 – 11 – 9
14.	6 – 7 – 10 – 9 – 8 – 11
15.	6 – 7 – 10 – 9 – 11 – 8

Dutch system

Example for brute force matching: chess pairing

www.fide.com/fide/handbook.html?id=168&view=article

11.2 In the following example of a score-group with six players, and pairing downward, the attempt is first made to find a compatible opponent for Player #1, the highest numbered player in the score-group.
Six players in a score-group with proposed pairings as follows:
1 v 4
2 v 5
3 v 6
If the pairing 1 v 4 is not compatible, for example, because the players had met in an earlier round, the positions of Player #4 and Player #5 are exchanged so that we have:
1 v 5
2 v 4
3 v 6
If the pairing 1 v 5 is also not compatible, a further exchange is made. The original proposed pairing and possible exchanges made to find a compatible opponent for Player #1 are as follows:
Proposed Pairing (col. 1) and Possible exchanges to find compatible opponent for #1

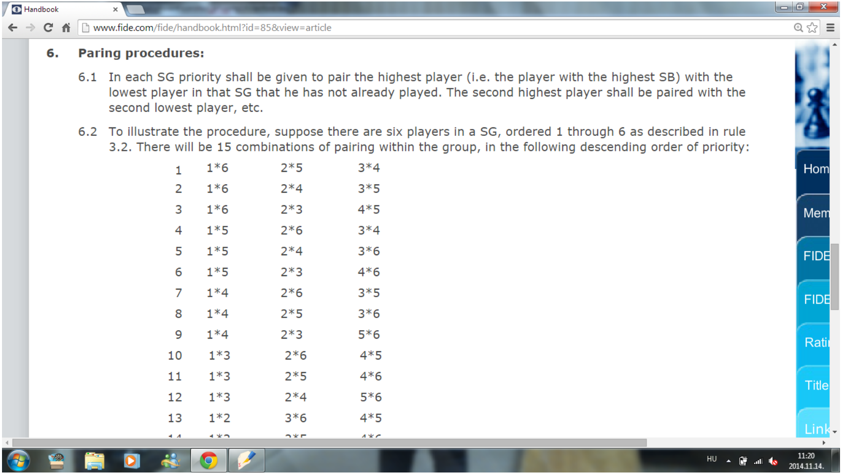
1 v 4	1 v 5	1 v 6	1 v 3	1 v 2
2 v 5	2 v 4	2 v 4	2 v 5	3 v 5
3 v 6	3 v 6	3 v 5	4 v 6	4 v 6

11.3 After a compatible opponent, for example, #6, has been found for Player #1, the proposed pairing for Player #2 is scrutinised. Exchanges to find a compatible opponent for Player #2 are as follows:
Proposed Pairing (col. 1) and Possible exchanges to find compatible opponent for #2

1 v 6	1 v 6	1 v 6	1 v 3	1 v 2
2 v 4	2 v 5	2 v 3	2 v 6	3 v 5
3 v 5	3 v 4	4 v 5	4 v 5	4 v 6

Lim system

Example for brute force matching: chess pairing



6. Paring procedures:

6.1 In each SG priority shall be given to pair the highest player (i.e. the player with the highest SB) with the lowest player in that SG that he has not already played. The second highest player shall be paired with the second lowest player, etc.

6.2 To illustrate the procedure, suppose there are six players in a SG, ordered 1 through 6 as described in rule 3.2. There will be 15 combinations of pairing within the group, in the following descending order of priority:

1	1*6	2*5	3*4
2	1*6	2*4	3*5
3	1*6	2*3	4*5
4	1*5	2*6	3*4
5	1*5	2*4	3*6
6	1*5	2*3	4*6
7	1*4	2*6	3*5
8	1*4	2*5	3*6
9	1*4	2*3	5*6
10	1*3	2*6	4*5
11	1*3	2*5	4*6
12	1*3	2*4	5*6
13	1*2	3*6	4*5

Burstein system

the tale continues...

King Arthur decided to make the dance party more colorful, so he asked Merlin to pick a different color for each dancing couple such that the color is matching with the flags of the corresponding noble families. Suppose that we have as many available colors as dancing couples. Can Merlin find a suitable solution, or a good excuse for not being able to find a suitable solution?

the tale continues...

King Arthur decided to make the dance party more colorful, so he asked Merlin to pick a different color for each dancing couple such that the color is matching with the flags of the corresponding noble families. Suppose that we have as many available colors as dancing couples. Can Merlin find a suitable solution, or a good excuse for not being able to find a suitable solution?

Now Merlin faces the **3D-matching** problem:
Given three sets of items, $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$, $C = \{c_1, \ldots, c_n\}$ and a set of possible triples: $\mathcal{F} = \{\ldots, (a_i, b_j, c_k), \ldots\}$. The question is whether there exists a set of disjoint triples, $F \subset \mathcal{F}$, s.t. all items are covered.

the tale continues...

King Arthur decided to make the dance party more colorful, so he asked Merlin to pick a different color for each dancing couple such that the color is matching with the flags of the corresponding noble families. Suppose that we have as many available colors as dancing couples. Can Merlin find a suitable solution, or a good excuse for not being able to find a suitable solution?

Now Merlin faces the **3D-matching** problem:
Given three sets of items, $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_n\}$, $C = \{c_1, \dots, c_n\}$ and a set of possible triples: $\mathcal{F} = \{\dots, (a_i, b_j, c_k), \dots\}$. The question is whether there exists a set of disjoint triples, $F \subset \mathcal{F}$, s.t. all items are covered.

Unfortunately this problem was shown to be NP-hard by Karp (1972), so it is highly unlikely that Merlin would be able to find a suitable solution, even if there exists one quickly, or give a good excuse for not finding a suitable solution...

NP-hard problems, complexity theory

For a decision problem Q, we say that Q∈P if there exists an algorithm, implementable with a **deterministic** Turing machine, which can decide in polynomial time in the input size for any instance $I \in Q$ whether I is a YES-instance.

Q∈NP if there exists an algorithm, implementable with a **non-deterministic** Turing machine, which can decide in polynomial time in the input size for any instance $I \in Q$ whether I is a YES-instance.

Alternative def: Q∈NP if for any instance I∈Q there is a proof T, polynomial size in I, that shows that I is a YES-instance and this be verified in polynomial time.

Q∈Co-NP: if there exists an algorithm, implementable with a **non-deterministic** Turing machine, which can decide in polynomial time in the input size for any instance $I \in Q$ whether I is a NO-instance.

NP-hard problems, complexity theory

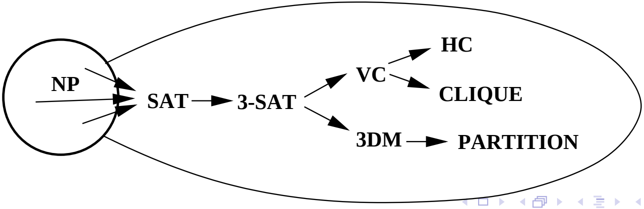
Polynomial-time reduction: problem A can be reduced to problem B if for any instance I of A we can create another instance I' of B, where

- the size of I' is polynomial in the size of I
- I is a YES-instance $\iff$ I' is a YES-instance.

A problem is NP-hard, if ANY problem in NP can be reduced to it.

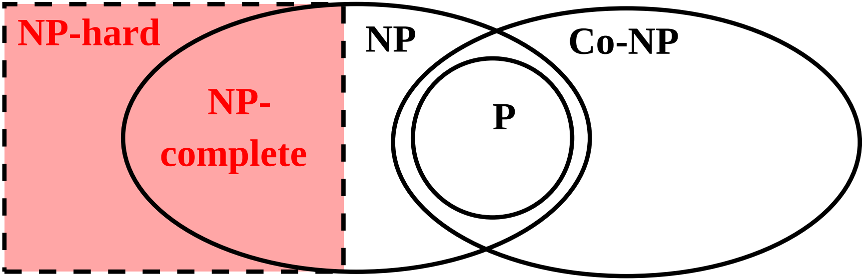
NP-complete$=$ NP $\cap$ NP-hard

Cook (1971): SAT is the first problem proved to be NP-complete. Since then there are thousands of relevant problems showed to be NP-complete.



NP-hard problems, complexity theory

Most likely picture:



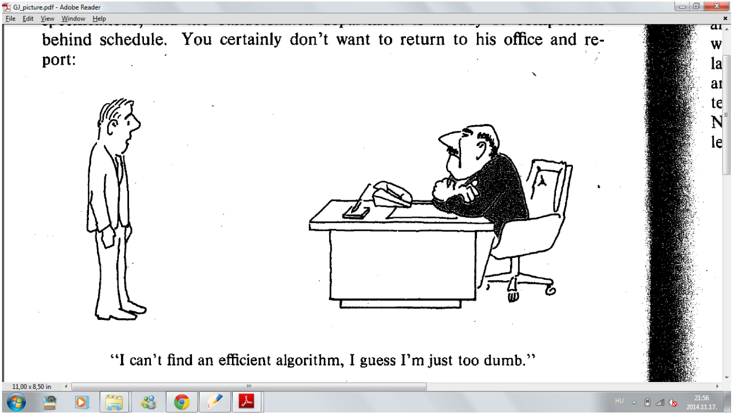
Although we still do not know whether P=NP?
or whether P=NP∩Co-NP?

NP-hard problems, complexity theory

So, if a problem is NP-hard then there exist no polynomial time algorithm to solve it, unless P=NP. (If we could solve an NP-hard problem in polynomial time then we could solve every problem in NP in polynomial time. This is very unlikely...)

NP-hard problems, complexity theory

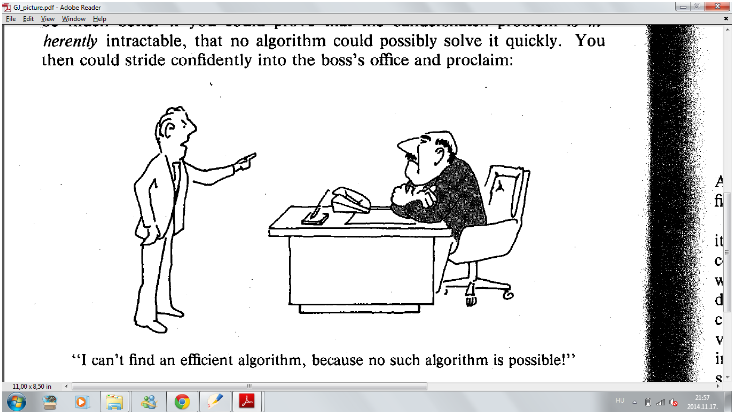
So, if a problem is NP-hard then there exist no polynomial time algorithm to solve it, unless P=NP. (If we could solve an NP-hard problem in polynomial time then we could solve every problem in NP in polynomial time. This is very unlikely...)



- M.R. Garey and D.S. Johnson. Computers and intractability. A guide to the theory of NP-completeness. Macmillan Higher Education, 1979.

NP-hard problems, complexity theory

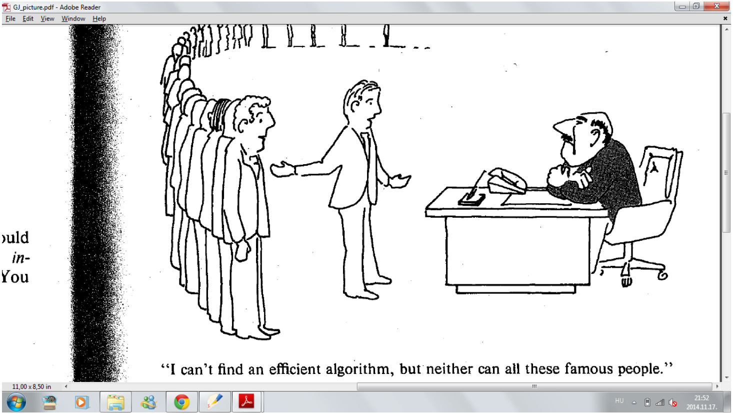
So, if a problem is NP-hard then there exist no polynomial time algorithm to solve it, unless P=NP. (If we could solve an NP-hard problem in polynomial time then we could solve every problem in NP in polynomial time. This is very unlikely...)



- M.R. Garey and D.S. Johnson. Computers and intractability. A guide to the theory of NP-completeness. Macmillan Higher Education, 1979.

NP-hard problems, complexity theory

So, if a problem is NP-hard then there exist no polynomial time algorithm to solve it, unless P=NP. (If we could solve an NP-hard problem in polynomial time then we could solve every problem in NP in polynomial time. This is very unlikely...)



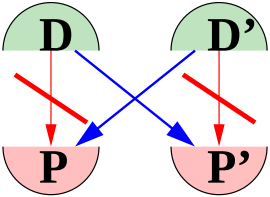
- M.R. Garey and D.S. Johnson. Computers and intractability. A guide to the theory of NP-completeness. Macmillan Higher Education, 1979.

NP-hard problems, complexity theory

If a problem turns out to be NP-hard, then we can still

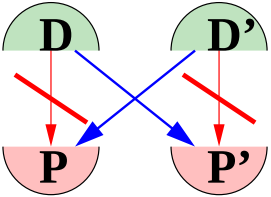
- specify the settings when the problem is still tractable (bipartite graphs, bounded length lists, etc.)
- give exact algorithm (exponential time, but terminating for small/sparse instances)
- give polynomial time algorithms with good approximation guarantees
- engineering (experimental) approach: construct heuristics with good performance on realistic instances
- use integer programming or other robust optimisation techniques

Kidney exchange problem



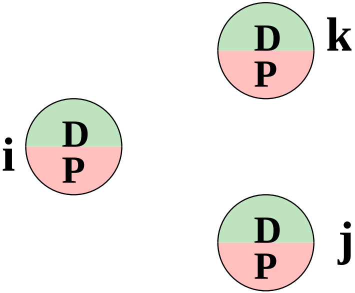
Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a pairwise exchange is possible between them.

Kidney exchange problem

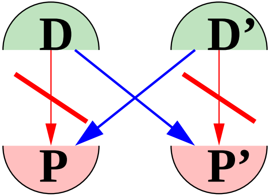


Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a pairwise exchange is possible between them.

We consider these pairs as single vertices of a directed graph, $D(V, A)$.



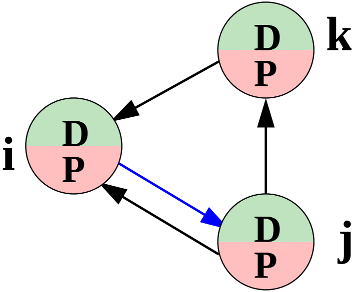
Kidney exchange problem



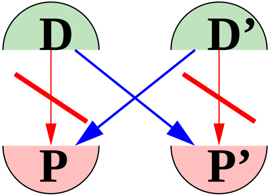
Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a pairwise exchange is possible between them.

We consider these pairs as single vertices of a directed graph, $D(V, A)$.

$(i, j) \in A$ iff the donor i is compatible with the patient j.



Kidney exchange problem

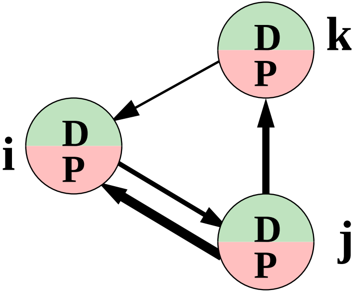


Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a pairwise exchange is possible between them.

We consider these pairs as single vertices of a directed graph, $D(V, A)$.

$(i, j) \in A$ iff the donor i is compatible with the patient j.

The **weight** *of an arc is the* **score** *of the corresponding donation (PRA, HLA-mismatch, age).*



The basic optimisation problems:

A **set of exchanges** is a permutation of V, s.t. $i \neq \pi(i)$ implies $(i, \pi(i)) \in A(D)$.

The basic optimisation problems:

A **set of exchanges** is a permutation of V, s.t. $i \neq \pi(i)$ implies $(i, \pi(i)) \in A(D)$.

We say that a set of exchanges is **optimal**, if the sum of the weights is maximal. (i.e., when the total score is maximal.)

The basic optimisation problems:

A **set of exchanges** is a permutation of V, s.t. $i \neq \pi(i)$ implies $(i, \pi(i)) \in A(D)$.

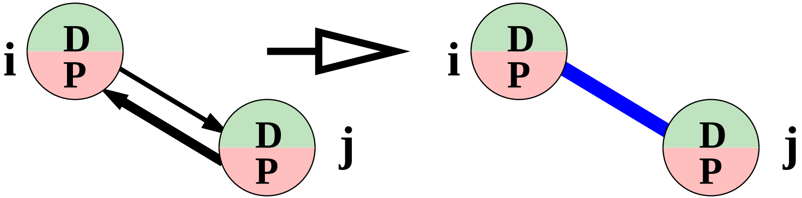
We say that a set of exchanges is **optimal**, if the sum of the weights is maximal. (i.e., when the total score is maximal.)

We study 3 cases:

- Only 2-cycles are possible.
- Unrestricted length cycles.
- 2- and 3-cycles are allowed.

2-way exchanges $\Longrightarrow$ matching problem

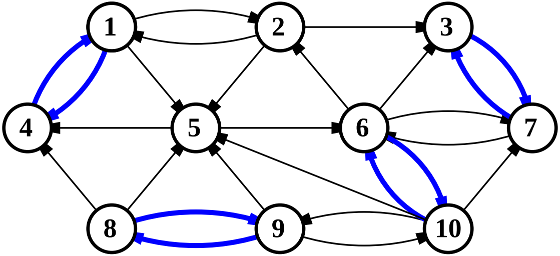
We transform the **directed graph** D to an undirected graph G.



A **set of 2-way exchanges in** D corresponds to a matching in G with the same weight, since $w(\{i,j\}) = w(i,j) + w(j,i)$ for every edge $\{i,j\}$ of G.

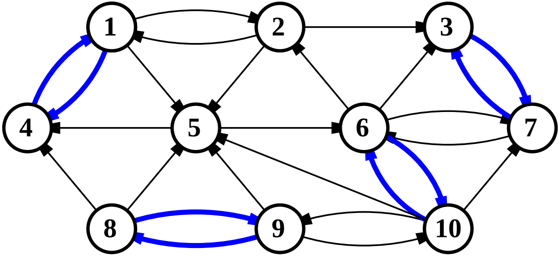
The problem of finding a maximum weight matching in G can be solved by Edmonds' algorithm in polynomial time.

Optimal pairwise exchanges in two examples



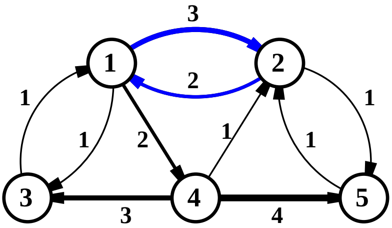
Maximum cardinality pairwise exchange

Optimal pairwise exchanges in two examples



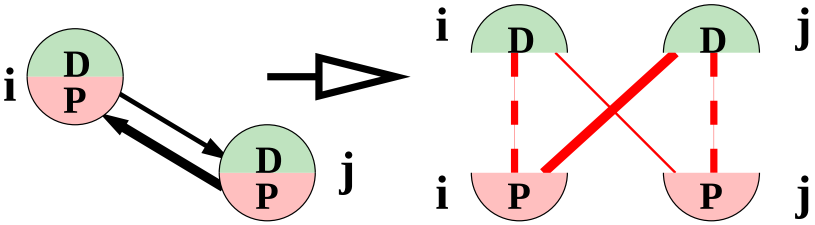
Maximum cardinality pairwise exchange

Maximum weight pairwise exchange



Unrestricted exchanges $\Longrightarrow$ matching problem

We transform the **directed graph** D to an bipartite graph G.

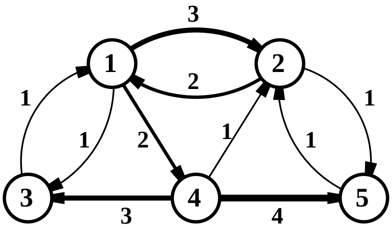


With an edge of weight 0, between each patient and his/her donor.

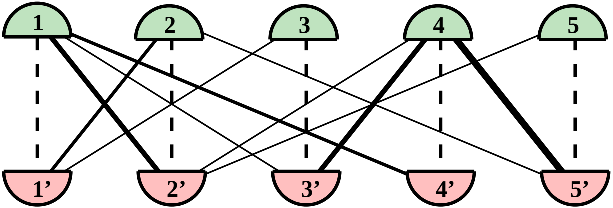
A **set of exchanges in** D corresponds to a complete matching in G with the same weight.

The problem of finding a maximum weight complete matching in G can be solved in polynomial time by the Hungarian method.

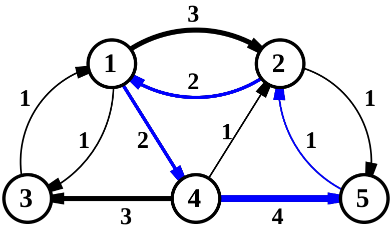
The transformation in an example



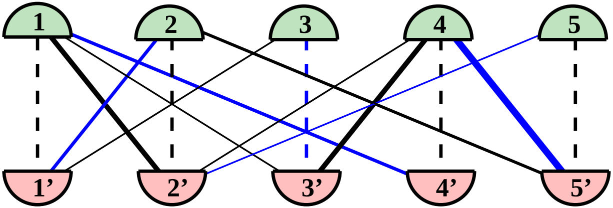
From a directed graph D,
we create a bipartite graph G,



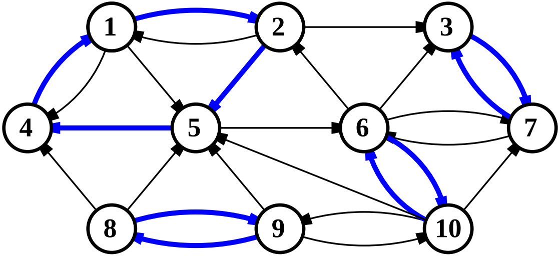
The transformation in an example



From a directed graph D, maximum weight unrestricted exchanges
we create a bipartite graph G, maximum weight complete matching

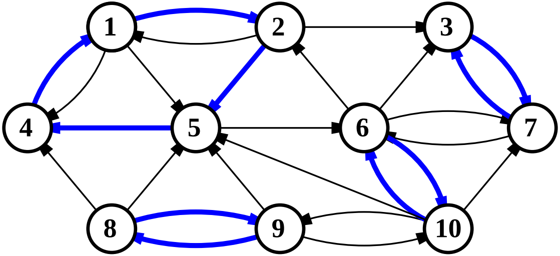


Optimal unrestricted exchanges in two examples



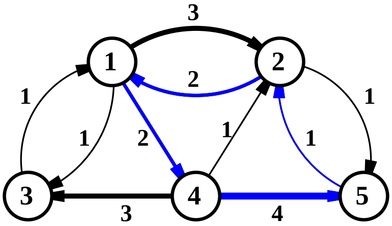
Maximum cardinality unrestricted exchanges

Optimal unrestricted exchanges in two examples



Maximum cardinality unrestricted exchanges

Maximum weight unrestricted exchanges



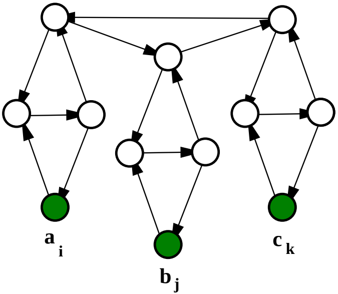
Test results for large instances:

	Pairwise exchange			Unrestricted exchange			
nodes	size	weight	time	size	weight	longest c.	time
100	46	971	0.3s	52	1458	(52)	0.3s
200	86	2662	0.9s	95	3215	(43)	1.0s
300	150	4151	2.0s	169	5459	(136)	2.3s
400	194	6760	3.4s	208	7662	(124)	4.0s
500	256	8161	5.4s	268	9056	(169)	7.1s
600	322	10404	7.9s	343	11606	(213)	9.5s
700	368	12495	10.4s	374	13520	(152)	14.3s
800	418	14447	14.0s	450	15370	(323)	20.0s
900	458	15543	17.2s	487	16703	(230)	24.2s
1000	516	17508	21.3s	530	18552	(191)	32.5s

2- and 3-way exchanges: an NP-hard problem

The problem of finding a maximum size / weight set of 2- and 3-way exchanges is NP-hard (reduction from 3DM):

for each triple $(a_i, b_j, c_k) \in \mathcal{F}$ we create the following gadget:



$\exists$ complete 3D matching $\iff$ $\exists$ complete set of 3-way exchanges

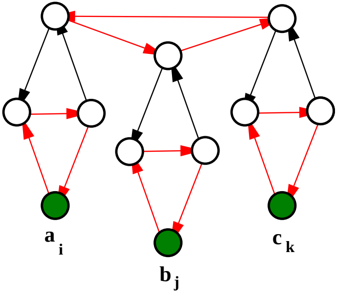
- D. J. Abraham, A. Blum and T. Sandholm, Clearing algorithms for barter-exchange markets: enabling nationwide kidney exchanges, In Proc. EC'07: the Eighth ACM Conference on Electronic Commerce, ACM, pp:295–304, 2007.

2- and 3-way exchanges: an NP-hard problem

The problem of finding a maximum size / weight set of 2- and 3-way exchanges is NP-hard (reduction from 3DM):

for each triple $(a_i, b_j, c_k) \in \mathcal{F}$ we create the following gadget:

$(a_i, b_j, c_k) \in F \iff$



$\exists$ complete 3D matching $\iff$ $\exists$ complete set of 3-way exchanges

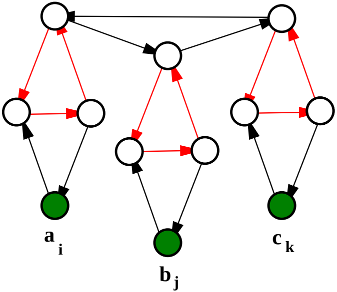
- D. J. Abraham, A. Blum and T. Sandholm, Clearing algorithms for barter-exchange markets: enabling nationwide kidney exchanges, In Proc. EC'07: the Eighth ACM Conference on Electronic Commerce, ACM, pp:295–304, 2007.

2- and 3-way exchanges: an NP-hard problem

The problem of finding a maximum size / weight set of 2- and 3-way exchanges is NP-hard (reduction from $3\textsc{DM}$):

for each triple $(a_i, b_j, c_k) \in \mathcal{F}$ we create the following gadget:

$(a_i, b_j, c_k) \notin F \iff$



$\exists$ complete 3D matching $\iff$ $\exists$ complete set of 3-way exchanges

- D. J. Abraham, A. Blum and T. Sandholm, Clearing algorithms for barter-exchange markets: enabling nationwide kidney exchanges, In Proc. EC'07: the Eighth ACM Conference on Electronic Commerce, ACM, pp:295–304, 2007.

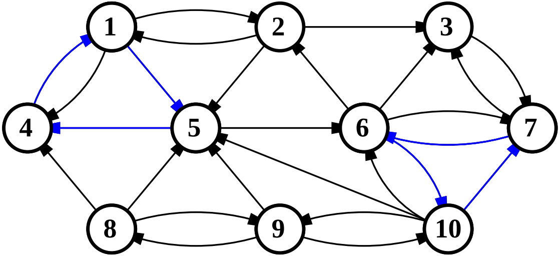
2- and 3-way exchanges: approximation algorithms

The greedy algorithm provides a 3-approximation for the maximum weight problem.

Biró-Manlove-Rizzi (2009): This can be improved to a $(2+\epsilon)$-approximation algorithm for any $\epsilon > 0$.

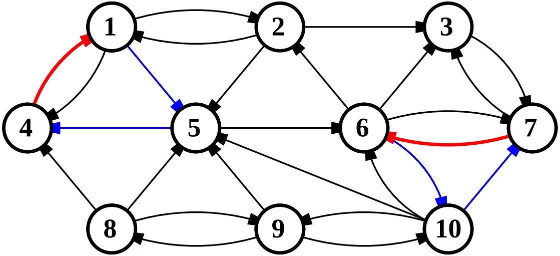
- P. Biró, D.F. Manlove and R. Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. Discrete Mathematics, Algorithms and Applications 1(4), pp:499-517, 2009.

Exact algorithm: reducing the running time 1.



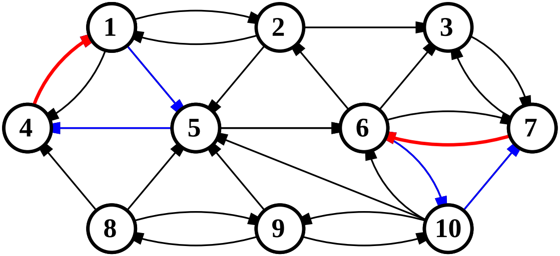
If we knew the set of 3-cycles of an optimal set of 2- and 3-way exchanges,
then we could find an optimal solution (by simply finding a maximum weight matching in the rest of the digraph).

Reducing the running time 2.

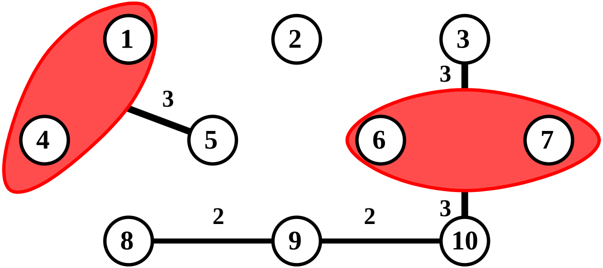


But it is enough to know only one arc from each 3-cycle, since we can find an optimal 2- and 3-way exchange after a transformation!

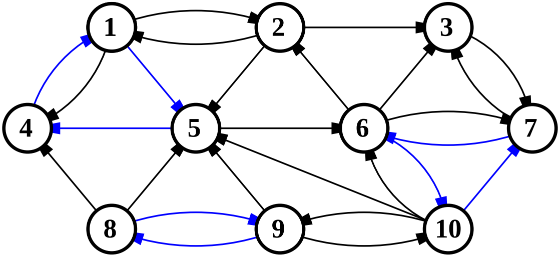
Reducing the running time 2.



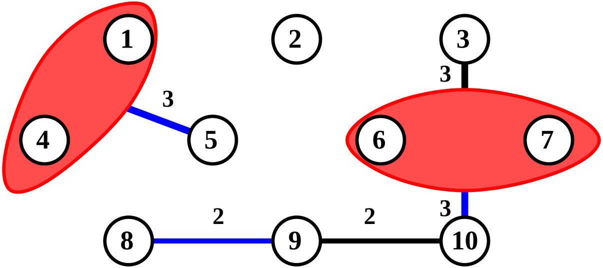
For an arc-set Y,
We create an undirected graph G_Y,



Reducing the running time 2.

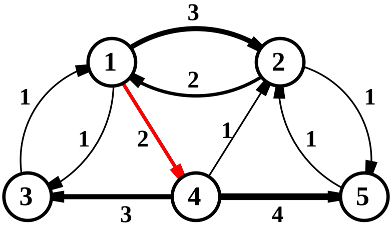


For an arc-set Y, maximum cardinality 2- and 3-way exchanges
We create an undirected graph G_Y, maximum weight matching



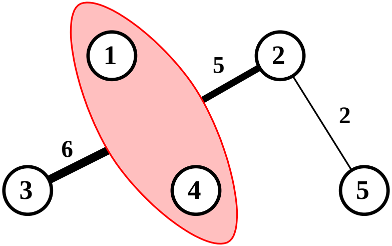
Reducing the running time 3.

In a weighted graph:



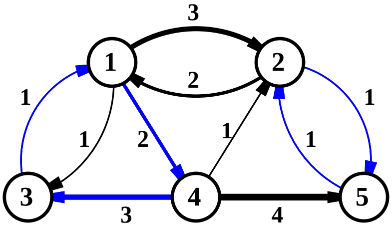
For an arc-set Y,

We create an undirected graph G_Y,



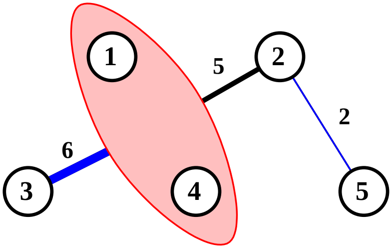
Reducing the running time 3.

In a weighted graph:

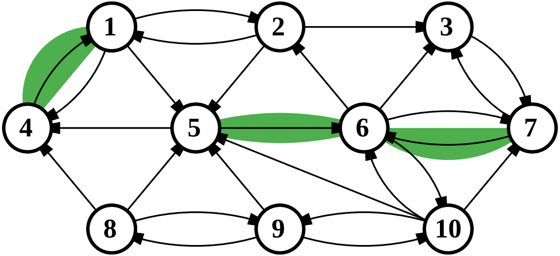


For an arc-set Y, maximum weight 2- and 3-way exchanges

We create an undirected graph G_Y, maximum weight matching

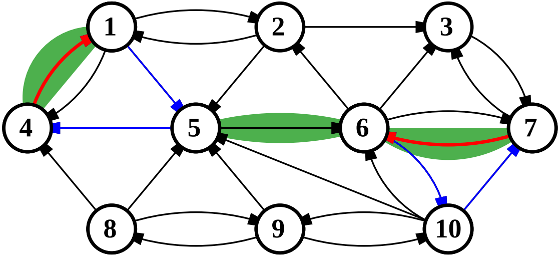


Reducing the running time 4.



Let T be an arc set in D such that after removing T from D no 3-cycle remains.

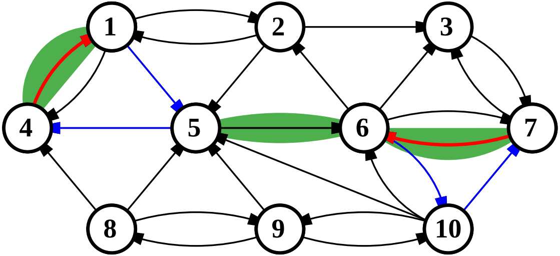
Reducing the running time 4.



Let T be an arc set in D such that after removing T from D no 3-cycle remains.

T intersects every 3-cycle of D, so T intersects also the 3-cycles of an optimal solution, thus Y can be chosen as a subset of T.

Reducing the running time 4.

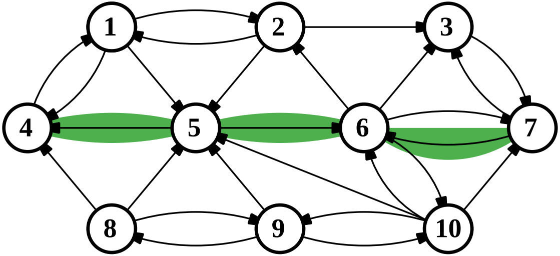


Let T be an arc set in D such that after removing T from D no 3-cycle remains.

T intersects every 3-cycle of D, so T intersects also the 3-cycles of an optimal solution, thus Y can be chosen as a subset of T.

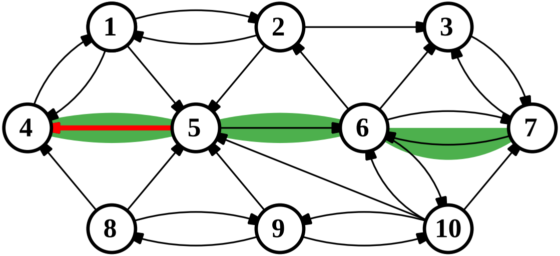
Here, T has **6** disjoint subsets, that we shall probe, so we can find an optimal set of 2- and 3-way exchanges by transforming the graph and running Edmonds' algorithm **6 times**.

Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

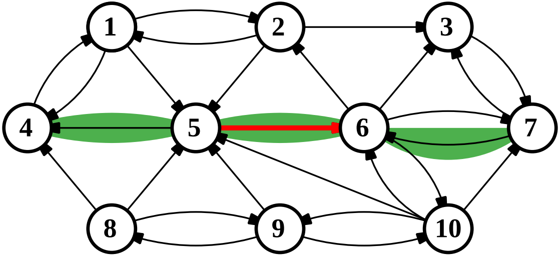
Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1,

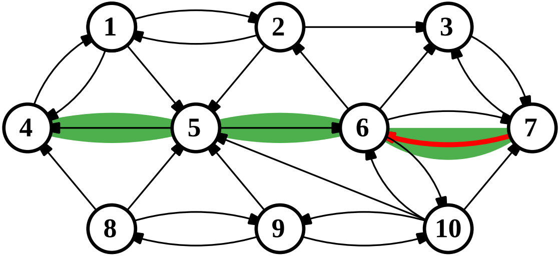
Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1, Y_2,

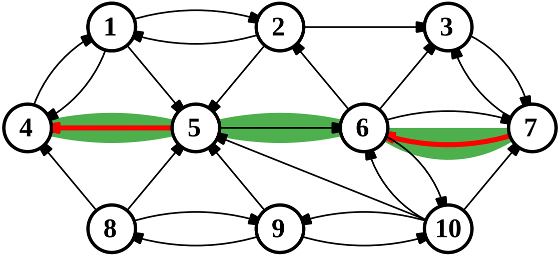
Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1, Y_2, Y_3,

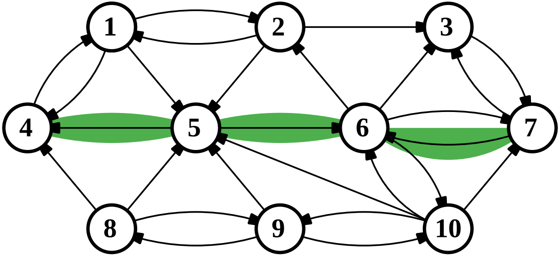
Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1, Y_2, Y_3, Y_4,

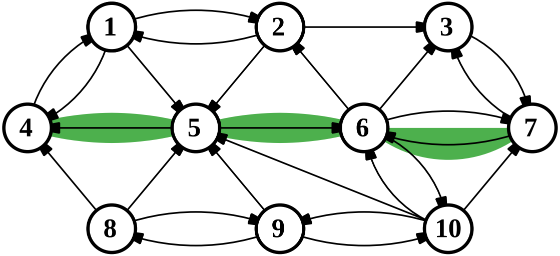
Reducing the running time 5.



We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1, Y_2, Y_3, Y_4, Y_5 (the emptyset).

Reducing the running time 5.

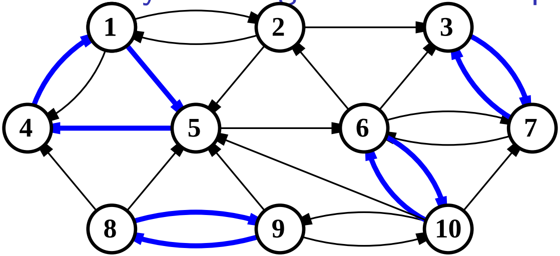


We shall choose a set T for which the number of independent subsets of T is minimal.

Here, T has the following 5 independent subsets:
Y_1, Y_2, Y_3, Y_4, Y_5 (the emptyset).

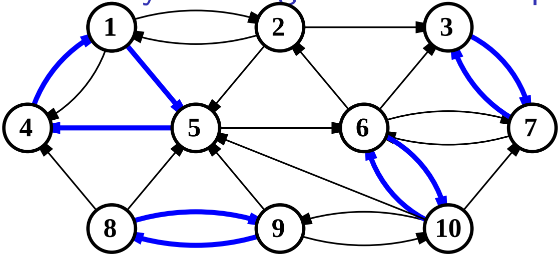
Clearly $|T| \leq m/2$, so the number of subsets that we need to check with Edmonds' algorithm is at most $2^{|T|} \leq 2^{\frac{m}{2}}$.

Optimal 2- and 3-way exchanges in two examples



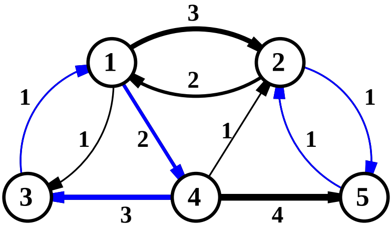
Maximum cardinality 2- and 3-way exchanges

Optimal 2- and 3-way exchanges in two examples

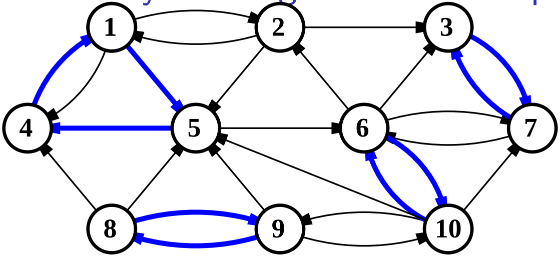


Maximum cardinality 2- and 3-way exchanges

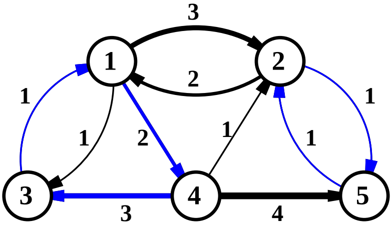
Maximum weight 2- and 3-way exchanges



Optimal 2- and 3-way exchanges in two examples



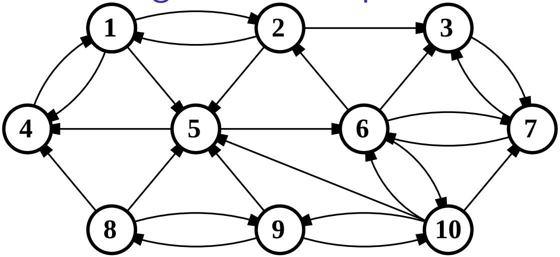
nodes	arcs	2-cycle	3-cycle	$\|T\|$	subsets of T	r. time
10	25	7	5	3	5	0.0s
5	10	3	2	1	2	0.0s



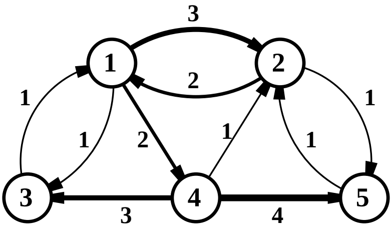
Test results for 2- and 3-way exchanges

nodes	arcs	2-cycle	3-cycle	$\|T\|$	subsets of T	r. time
10	22	2	0	0	0	0.0s
15	45	7	13	3	6	0.1s
20	101	7	5	2	3	0.0s
25	125	16	37	5	6	0.1s
30	239	16	36	8	40	0.4s
35	339	32	111	16	656	7.2s
40	354	25	145	17	296	3.8s
45	541	48	185	22	1792	28.8s
50	502	46	257	21	336	6.2s
55	609	59	151	19	992	18.9s
60	696	51	164	25	5172	121.4s
65	993	89	620	52	1841364	55387.1s
70	1164	133	778	55	555624	17665.4s

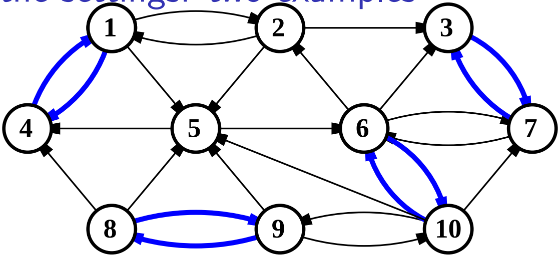
Comparing the settings: two examples



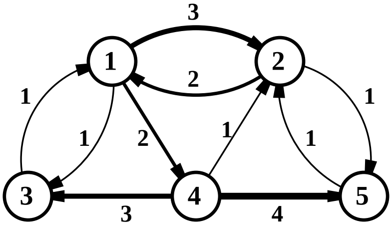
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	9	9	1	10	10	(4)
5	2	5	5	8	1	4	9	(4)



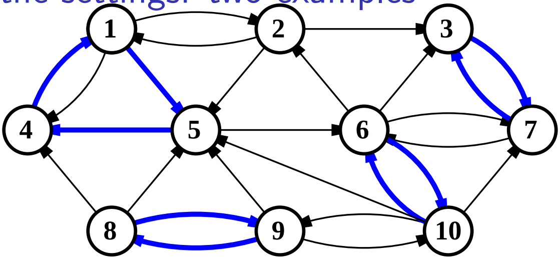
Comparing the settings: two examples



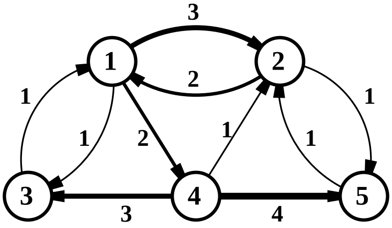
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	**8**	**8**	9	9	1	10	10	(4)
5	2	5	5	8	1	4	9	(4)



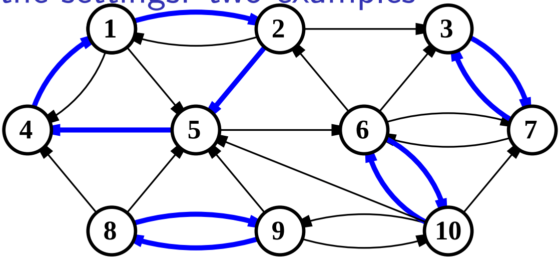
Comparing the settings: two examples



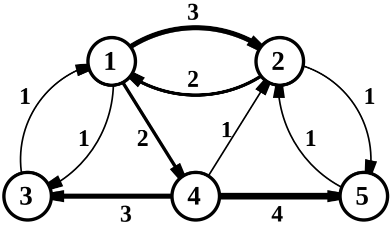
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	**9**	**9**	**1**	10	10	(4)
5	2	5	5	8	1	4	9	(4)



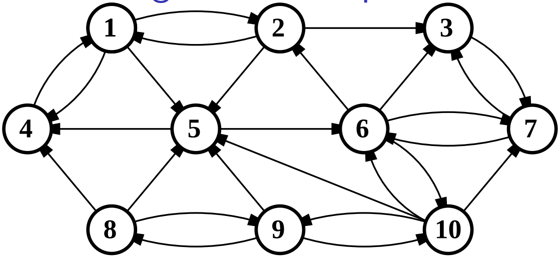
Comparing the settings: two examples



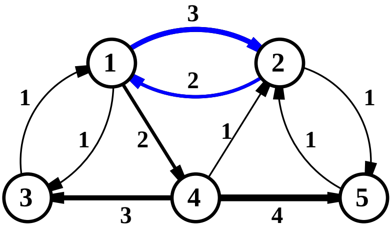
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	9	9	1	**10**	**10**	**(4)**
5	2	5	5	8	1	4	9	(4)



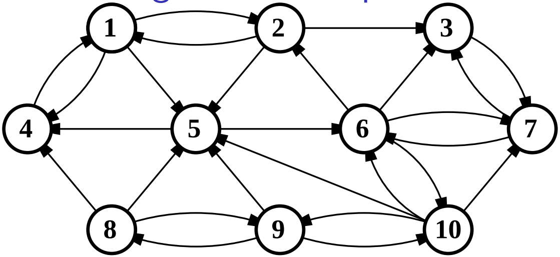
Comparing the settings: two examples



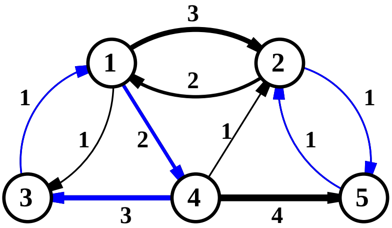
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	9	9	1	10	10	(4)
5	**2**	**5**	5	8	1	4	9	(4)



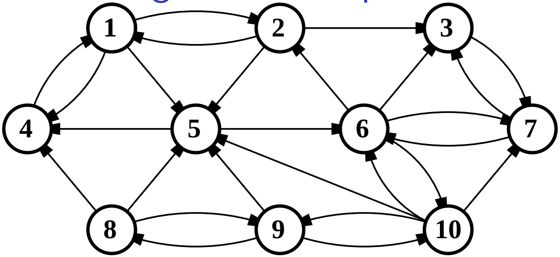
Comparing the settings: two examples



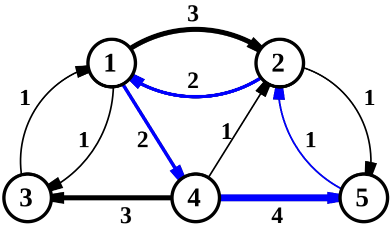
	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	9	9	1	10	10	(4)
5	2	5	**5**	**8**	**1**	4	9	(4)



Comparing the settings: two examples



	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	8	8	9	9	1	10	10	(4)
5	2	5	5	8	1	**4**	**9**	**(4)**



Comparing the settings: test results

	Pairwise		2- and 3-way			Unrestricted		
nodes	size	weight	size	weight	3-c.	size	weight	longest c.
10	2	24	2	24	0	2	24	(2)
15	6	140	6	170	2	6	170	(6)
20	6	230	7	282	1	7	282	(3)
25	6	162	6	162	0	6	162	(4)
30	12	656	15	956	3	15	956	(8)
35	16	562	18	820	2	19	866	(7)
40	12	574	15	960	3	16	1006	(7)
45	20	1092	23	1298	3	23	1298	(19)
50	14	466	17	762	3	20	966	(15)
55	20	1098	23	1334	3	25	1524	(11)
60	18	1216	23	1576	5	23	1722	(21)
65	26	994	29	1402	5	31	1510	(28)
70	26	1174	31	1470	7	31	1470	(31)

dmaa_rev.pdf

Matching run		2008 Apr	2008 Jul	2008 Oct	2009 Jan	2009 Apr	2009 Jul	2009 Oct
# pairs		76	85	123	126	122	95	97
# possible donations		287	235	704	576	760	1212	866
Total #	2-cycles	5	2	14	16	20	54	4
	3 cycles	5	0	109	65	68	164	4
Pairwise exchanges	#2-cycles	2	1	6	5	5	10	2
	size	4	2	12	10	10	20	4
	weight	91	6	499	264	388	739	222
$\leq$3-way exchanges	#2-cycles	2	1	2	1	2	2	0
	#3-cycles	4	0	7	5	5	9	2
	size	16	2	25	17	19	31	6
	weight	620	6	1122	633	757	1300	300
the exact algorithm	size of S	5	0	18	13	14	25	3
	$\#Y \subseteq S$	24	0	3480	588	1440	67824	6
Running time (sec)		0.3	0.0	66.0	7.5	19.2	1494.3	2.0
Unbounded exchanges	size	22	2	33	28	28	40	6
	weight	857	6	1546	1134	1275	1894	300
	longest c.	20	2	27	19	23	28	3
Chosen solution (NHSBT)	#2-cycles	2	1	6	5	5	4	1
	#3-cycles	4	0	3	1	2	7	1
	size	16	2	21	13	16	29	5
	weight	620	6	930	422	618	1168	288

Table 1. Results arising from matching runs from April 2008 to October 2009.

We also used our exact algorithm to find optimal exchanges for NHSBT for the quarterly matching runs of the NMSPD from April 2008 to October 2009 inclusive, and the results corresponding to these input datasets are contained in Table 1. The

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

Daily Mail, Thursday, December 6, 2007

The transplant pact

Two saved as families exchange kidneys

By Luke Salkeld

THEY were both in desperate need of a kidney donor, and both had relatives who were willing to sacrifice an organ.

But without a family match, strangers Donald Planner and Margaret Wearn instead entered into an extraordinary pact.

Mr Planner's daughter donated her kidney to Mrs Wearn, whose husband gave his kidney to Mr Planner.

The operations took place 170 miles apart in synchronised procedures with the organs transported by ambulances travelling in opposite directions between the two hospitals.

'Completely amazing': Donald Planner with his daughter Suzanne

Margaret and Roger Wearn: 'No different to a direct donation'

organ or he would die. His ... ily reliant on the dialysis

Page last updated at 10:41 GMT, Monday, 8 March 2010

Three-way kidney transplant success

By Graham Satchell
BBC News Breakfast reporter

Step back to nine in the morning on 4 December 2009.

Six patients are ready for surgery at three different hospitals across the UK.

It is the culmination of months of preparation and a remarkable event in the history of live organ donation in this country.

This is a three-way kidney swap between couples who've never met.

In Aberdeen, 54-year-old Andrea Mullen suffered sudden kidney failure three years ago.

It had a devastating impact on her life. She had to have dialysis three

Chris Brent with his sister Lisa Burton

> **"It's a threefold thing really so it's a real good feelgood factor all round"**

Lisa Burton, who donated a kidney

SEE ALSO
- Three-way transplant brings hope
  08 Mar 10 | Health

RELATED BBC LINKS
- Kidney transplant

RELATED INTERNET LINKS
- Human Tissue Authority
- Hammersmith hospital
- Guy's and St Thomas' hospital
- Edinburgh Royal Infirmary

The BBC is not responsible for the content of ext internet sites

TOP HEALTH STORIES
- Stem cell method put to the test
- Hospitals 'eyeing private market'
- Low vitamin D 'Parkinson's link'

| News feeds

Alternative method: integer linear programming

We create an integer program as follows:

- we list all the possible exchanges: $C_1, C_2, \ldots, C_m$
- we use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1$ iff C_i is part of optimal solution x
- we build matrix A of dimensions $n \times m$ where $n = |V|$ and $A_{i,j} = 1$ iff v_i is incident to C_j
- let b be $n \times 1$ vector of 1s
- let c be $1 \times m$ vector of values according to what we want to optimise, e.g. c_j could be weight of C_j

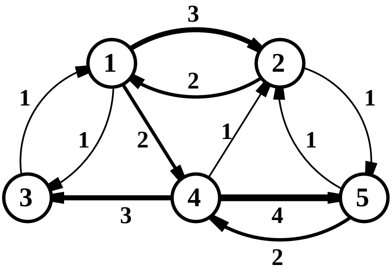
Then solve max cx s.t. $Ax \leq b$

- D. J. Abraham, A. Blum and T. Sandholm, Clearing algorithms for barter-exchange markets: enabling nationwide kidney exchanges, In Proc. EC'07: the Eighth ACM Conference on Electronic Commerce, ACM, pp:295–304, 2007.

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0, 1\}$

where

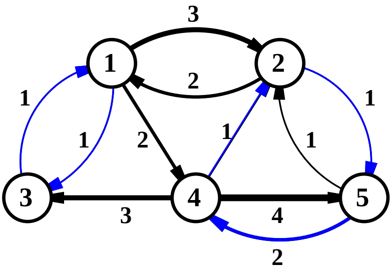


$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right], b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \text{and}$$

$$c_s = \left[\begin{array}{cccc|ccc} 2 & 2 & 2 & 2 & 3 & 3 & 3 \end{array} \right] \text{ if maximum size}$$

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0, 1\}$



where

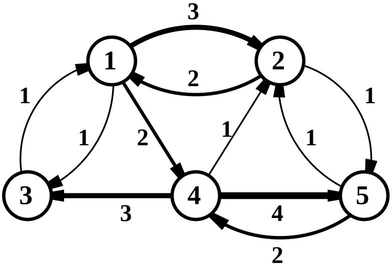
$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right], b = \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}\right], x = \left[\begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{array}\right] \text{ and}$$

$$c_s = \left[\begin{array}{cccc|ccc} 2 & 2 & 2 & 2 & 3 & 3 & 3 \end{array}\right] \text{ if maximum size } \max c_s x = 5$$

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0, 1\}$

where

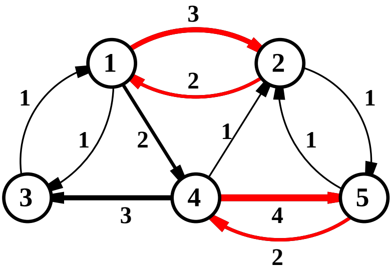


$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right], b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \text{and}$$

$$c_w = \left[\begin{array}{cccc|ccc} 5 & 2 & 2 & 6 & 5 & 6 & 4 \end{array}\right] \text{ if maximum weight}$$

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0,1\}$



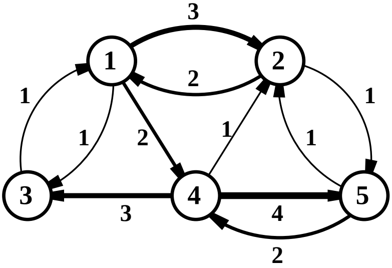
where

$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right], b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and}$$

$$c_w = \left[\begin{array}{cccc|ccc} 5 & 2 & 2 & 6 & 5 & 6 & 4 \end{array}\right] \text{ if maximum weight } \max c_w x = 11$$

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0, 1\}$



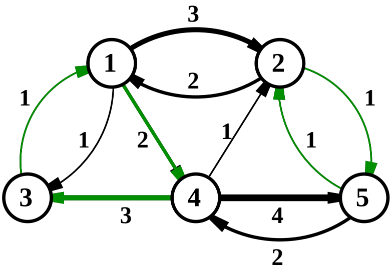
where

$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right], b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \text{and}$$

$c_o = c_s \cdot M + c_w$ if max weight max size

Alternative method: integer linear programming

max cx
s.t. $Ax \leq b$
and $x_i \in \{0, 1\}$

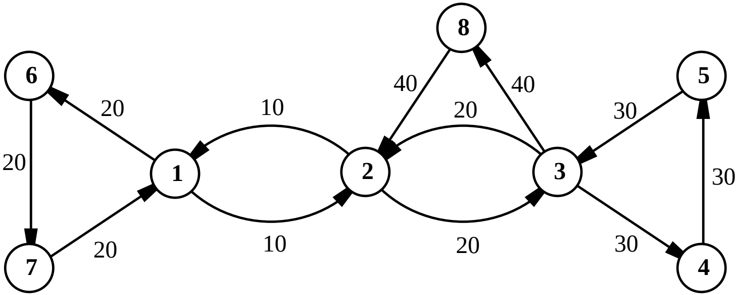


where

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, x = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ and}$$

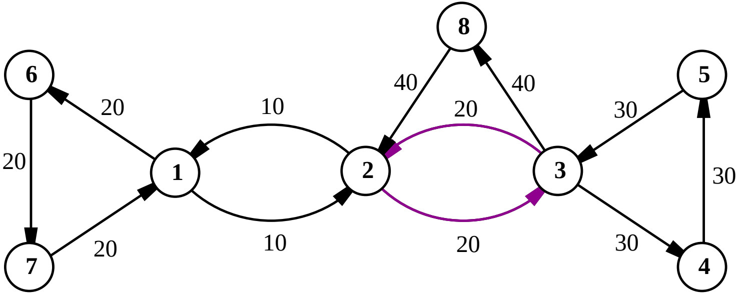
$c_o = c_s \cdot M + c_w$ if max weight max size max $c_o x = 5M + 8$

Changing the optimisation criteria in the UK program



- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012, vol. 7276 of LNCS, pp 271-282.

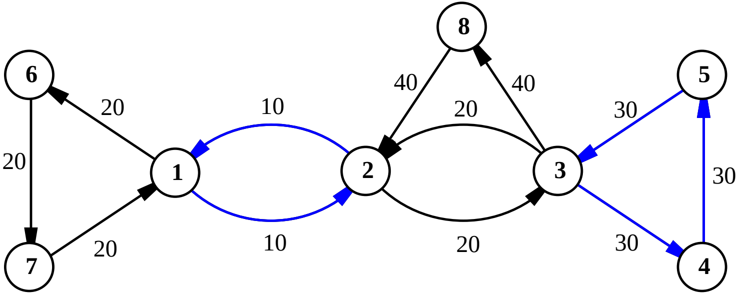
Changing the optimisation criteria in the UK program



best (maximum weight maximum size) set of 2-way exchanges,

- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012, vol. 7276 of LNCS, pp 271-282.

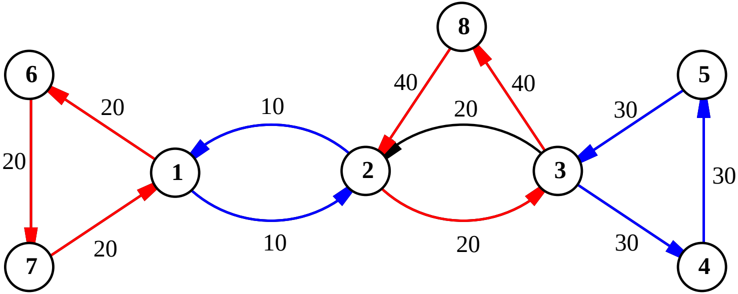
Changing the optimisation criteria in the UK program



best (maximum weight maximum size) set of 2-way exchanges,
best set of 2-way exchanges with **extra** 3-way exchanges

- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012, vol. 7276 of LNCS, pp 271-282.

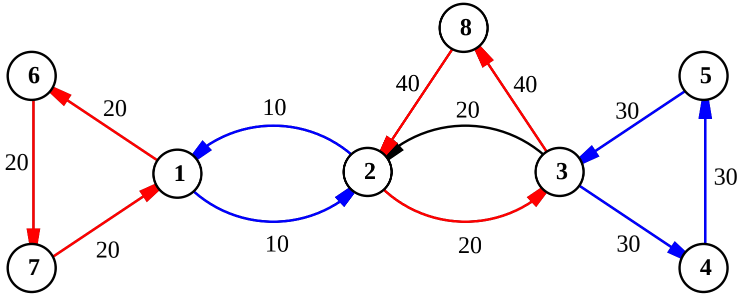
Changing the optimisation criteria in the UK program



best (maximum weight maximum size) set of 2-way exchanges,
best set of 2-way exchanges with **extra** 3-way exchanges
best set of 2-way exchanges and 3-way exchanges with **embedded** 2-way exchanges.

- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012, vol. 7276 of LNCS, pp 271-282.

Changing the optimisation criteria in the UK program

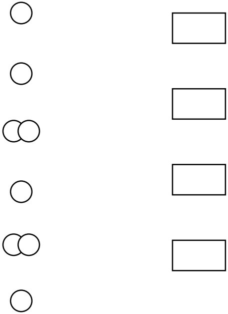


best (maximum weight maximum size) set of 2-way exchanges,
best set of 2-way exchanges with **extra** 3-way exchanges
best set of 2-way exchanges and 3-way exchanges with **embedded** 2-way exchanges. (July 2009: We could replace eight from the ten 2-way exchanges by 3-way exchanges with embedded 2-way exchanges.)

- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012, vol. 7276 of LNCS, pp 271-282.

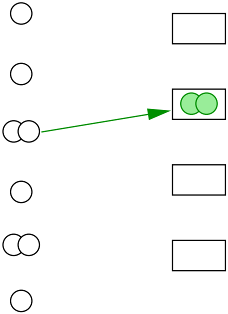
Matching couples with 0-1 preferences

We have $2n$ people, containing some couples, and n double rooms.



Matching couples with 0-1 preferences

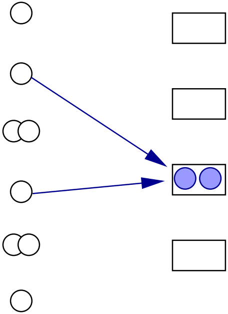
We have $2n$ people, containing some couples, and n double rooms.



- each couple has to be accommodated in a double room

Matching couples with 0-1 preferences

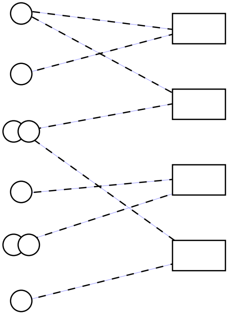
We have $2n$ people, containing some couples, and n double rooms.



- each couple has to be accommodated in a double room
- two single persons can be placed in one double room

Matching couples with 0-1 preferences

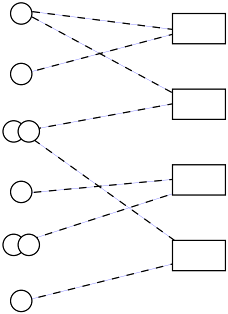
We have $2n$ people, containing some couples, and n double rooms.



- each couple has to be accommodated in a double room
- two single persons can be placed in one double room
- every single person and couple has a list of suitable rooms

Matching couples with 0-1 preferences

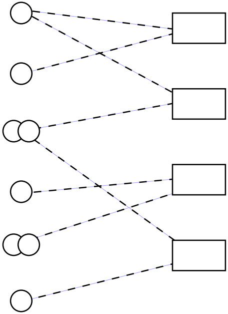
We have $2n$ people, containing some couples, and n double rooms.



- each couple has to be accommodated in a double room
- two single persons can be placed in one double room
- every single person and couple has a list of suitable rooms

Is it possible to accommodate everybody?

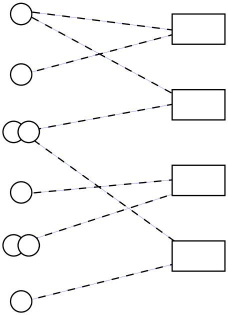
Motivation: matching couples, scheduling jobs



- allocating singles and couples by maximising the size

- P.A. Robards. Applying two-sided matching processes to the United States Navy enlisted assignment process, Master's Thesis, Naval Postgraduate School, Monterey, California, 2001.
- W. Yang, J.A. Giampapa, K. Sycara, Two-sided matching for the US Navy Detailing Process with market complication, Technical Report CMU-RI-TR-03-49, Robotics Institute, Carnegie-Mellon University, 2003.

Motivation: matching couples, scheduling jobs

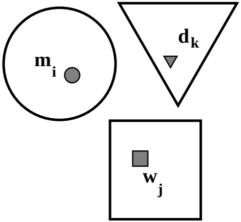


- allocating singles and couples
  by maximising the size
- multiprocessor scheduling: allocating
  jobs (of length 1 or 2) to processors
  by minimising the makespan
- bin packing: allocating
  items of size 0.5 or 1 to bins (of size 1)
  by minimising the number of bins used

- P.A. Robards. Applying two-sided matching processes to the United States Navy enlisted assignment process, Master's Thesis, Naval Postgraduate School, Monterey, California, 2001.
- W. Yang, J.A. Giampapa, K. Sycara, Two-sided matching for the US Navy Detailing Process with market complication, Technical Report CMU-RI-TR-03-49, Robotics Institute, Carnegie-Mellon University, 2003.
- C.A. Glass and H. Kellerer. Parallel machine scheduling with job assignment restrictions, Naval Research Logistics. A Journal Dedicated to Advances in Operations and Logistics Research 54(3), pp:250–257, 2007.
- P. Biró and E. McDermid. Matching with sizes (or scheduling with processing set restrictions). Discrete Applied Mathematics 164(1), pp:61–67, 2014.

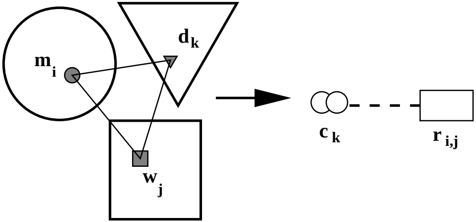
The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:



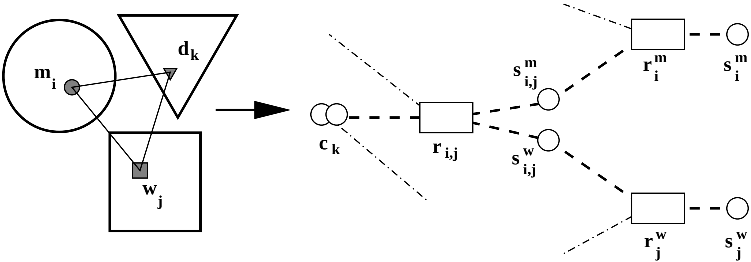
The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:



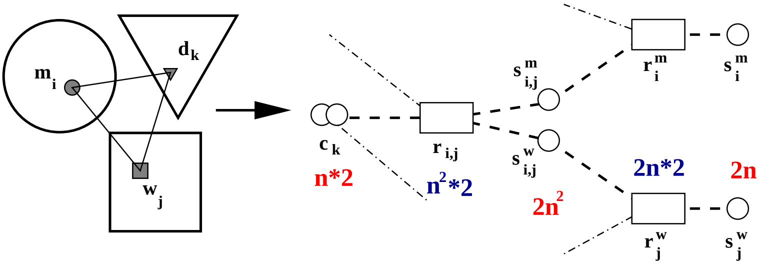
The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:



The NP-hardness proof

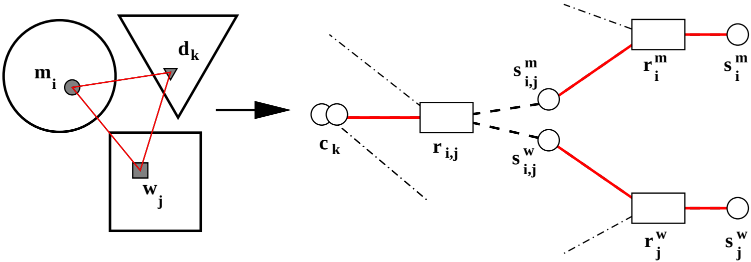
Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:



$\exists$ complete 3D-matching $\iff$ $\exists$ complete matching with couples

The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:

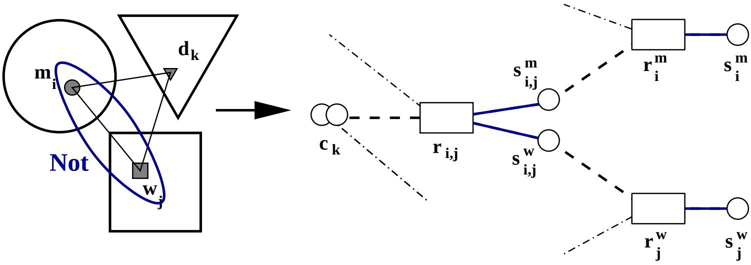


$\exists$ complete 3D-matching $\iff$ $\exists$ complete matching with couples

$\Longrightarrow$ Suppose that we have a complete matching F...

The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:

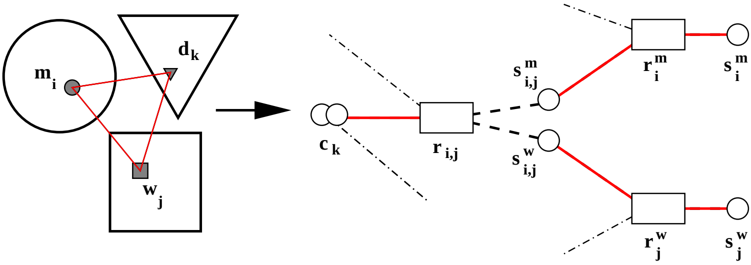


$\exists$ complete 3D-matching $\iff$ $\exists$ complete matching with couples

$\Longrightarrow$ Suppose that we have a complete matching F...

The NP-hardness proof

Glass-Kellerer (2007), Biró-McDermid (2014):
We reduce from 3DM:



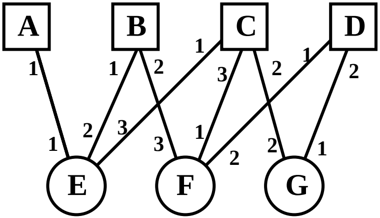
$\exists$ complete 3D-matching $\Longleftrightarrow$ $\exists$ complete matching with couples

$\Longleftarrow$ similarly...

Matching **under preferences**

Stable marriage problem by Gale and Shapley [1962]

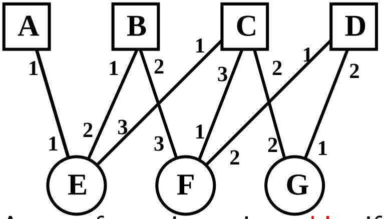
"College admission and the stability of marriage"



"Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner."

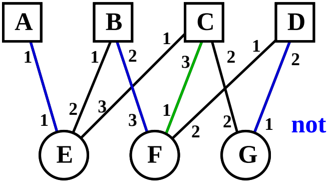
Stable marriage problem by Gale and Shapley [1962]

"College admission and the stability of marriage"



"Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner."

A set of marriages is stable, if there is no "blocking pair": a man and a woman who are not married to each other but prefer each other to their actual mates.

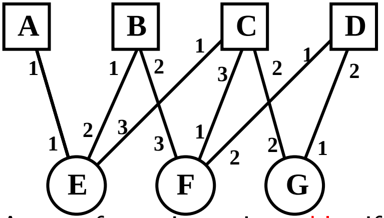


not stable

(C,F) blocking pair

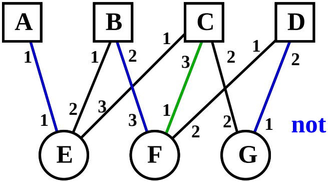
Stable marriage problem by Gale and Shapley [1962]

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is stable, if there is no “blocking pair”: a man and a woman who are not married to each other but prefer each other to their actual mates.



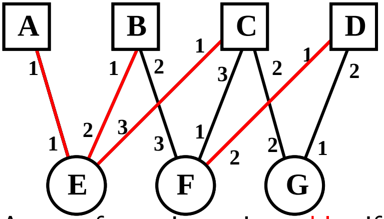
not stable

(C,F) blocking pair

Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching.

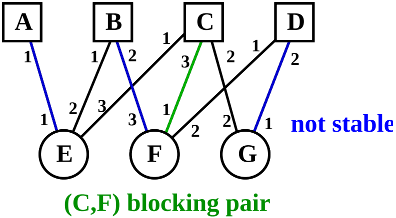
Stable marriage problem by Gale and Shapley [1962]

"College admission and the stability of marriage"



"Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner."

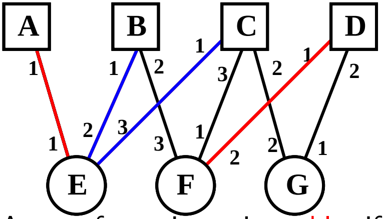
A set of marriages is stable, if there is no "blocking pair": a man and a woman who are not married to each other but prefer each other to their actual mates.



Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching.

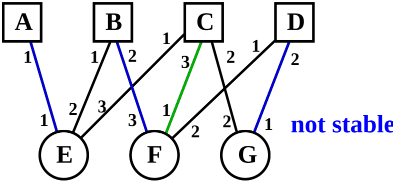
Stable marriage problem by Gale and Shapley [1962]

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is stable, if there is no “blocking pair”: a man and a woman who are not married to each other but prefer each other to their actual mates.

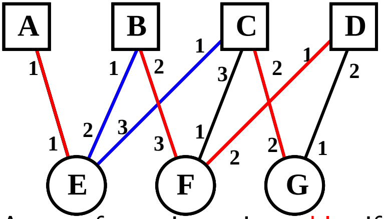


(C,F) blocking pair

Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching.

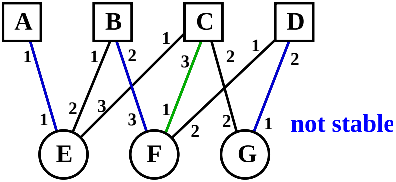
Stable marriage problem by Gale and Shapley [1962]

“College admission and the stability of marriage”



“Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner.”

A set of marriages is stable, if there is no “blocking pair”: a man and a woman who are not married to each other but prefer each other to their actual mates.

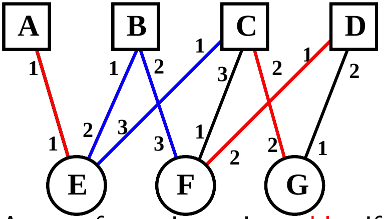


(C,F) blocking pair

Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching.

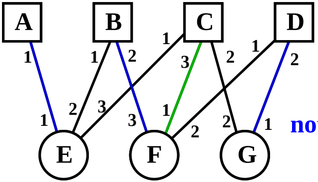
Stable marriage problem by Gale and Shapley [1962]

"College admission and the stability of marriage"



"Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner."

A set of marriages is stable, if there is no "blocking pair": a man and a woman who are not married to each other but prefer each other to their actual mates.



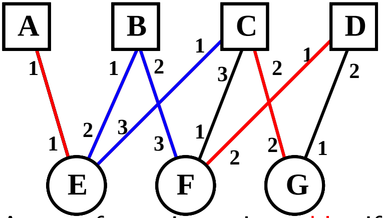
not stable

(C,F) blocking pair

Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching.

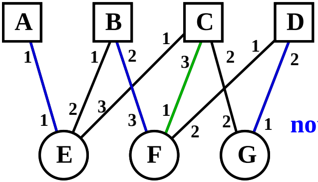
Stable marriage problem by Gale and Shapley [1962]

"College admission and the stability of marriage"



"Each person ranks those of the opposite sex in accordance with his or her preferences for a marriage partner."

A set of marriages is stable, if there is no "blocking pair": a man and a woman who are not married to each other but prefer each other to their actual mates.



not stable

(C,F) blocking pair

Gale-Shapley 1962: The deferred-acceptance algorithm finds a stable matching. This matching is *man-optimal*.

SM + quotas: College Admissions (CA)

The solution by the Gale-Shapley mechanism is

- fair: an application is rejected by a college only if its quota is filled with better applicants (i.e., the matching is stable).
- student-optimal: no student could be admitted to a better college in any other fair solution.

SM + quotas: College Admissions (CA)

The solution by the Gale-Shapley mechanism is

- fair: an application is rejected by a college only if its quota is filled with better applicants (i.e., the matching is stable).
- student-optimal: no student could be admitted to a better college in any other fair solution.

The automated procedure based on the Gale-Shapley algorithm is

- fast: the running time is linear in the number of applications (10 seconds in Hungary, would be $\sim$1 minutes in the UK and $\sim$15 minutes in China).
- strategy-proof: no student can be better off by cheating.

The Gale–Shapley algorithm in practice

Allocating residents to positions:

- National Resident Matching Program since 1952!
- and many other professions in the US and other countries...
  (e.g., Scottish Foundation Allocation Scheme)

The Gale–Shapley algorithm in practice

Allocating residents to positions:

- National Resident Matching Program since 1952!
- and many other professions in the US and other countries...
  (e.g., Scottish Foundation Allocation Scheme)

Admission systems in education:

- New York high schools since 2004,
  Boston high schools since 2005
- Higher education admissions in Spain (1998)
- Higher education admissions in Hungary since 1996
- Secondary school admissions in Hungary since 2000
  (Original Gale–Shapley model and algorithm!)

Matching under preferences...

List of hard problems to be discussed:

- finding weakly stable matchings **as large as possible**
- finding large matchings **as stable as possible**
- finding a matching that is the **most likely** to be stable
- stable cyclic **3D-matchings**, stable **exchanges**
- special features in college admissions: **paired applications, lower and common quotas**
- resident allocation problem with **couples**

Finding maximum size weakly stable matchings

Scottish Foundation Allocation Scheme

Hospitals can have **ties** in their rankings...

Applicants:	**Adam**	**Bill**
1st application:	Glasgow	Glasgow
2nd application:	Edinburgh	

the ranking of SG Glasgow Hospital: [Adam, Bill]
the ranking of Royal Edinburgh Hospital: Adam

Finding maximum size weakly stable matchings

Scottish Foundation Allocation Scheme

Hospitals can have **ties** in their rankings...

Applicants:	**Adam**	**Bill**
1st application:	Glasgow	Glasgow
2nd application:	Edinburgh	

the ranking of SG Glasgow Hospital: [Adam, Bill]

the ranking of Royal Edinburgh Hospital: Adam

Finding maximum size weakly stable matchings

Scottish Foundation Allocation Scheme

Hospitals can have **ties** in their rankings...

Applicants:	**Adam**	**Bill**
1st application:	Glasgow	Glasgow
2nd application:	Edinburgh	

the ranking of SG Glasgow Hospital: [Adam, Bill]
the ranking of Royal Edinburgh Hospital: Adam

Weakly stable matchings can have different sizes.

Finding maximum size weakly stable matchings

Scottish Foundation Allocation Scheme

Hospitals can have **ties** in their rankings...

Applicants:	**Adam**	**Bill**
1st application:	Glasgow	Glasgow
2nd application:	Edinburgh	

the ranking of SG Glasgow Hospital: [Adam, Bill]
the ranking of Royal Edinburgh Hospital: Adam

Weakly stable matchings can have different sizes.

Iwama, Manlove et. al. (1999): Finding a maximum size weakly stable matching is NP-hard (reduction from EXACT-MM: finding a maximal matching of given size).

Restrictions, approximability, inapproximability

The problem is NP-hard even if ties occur on one side only, each preference list is strictly ordered or is a single tie, and

- Manlove et al. (2002): each tie is of length 2
- Irving-Manlove-O'Malley (2009): length of pref. lists ≤ 3
- Irving-Manlove-Scott (2008): master lists on both sides

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

Restrictions, approximability, inapproximability

The problem is NP-hard even if ties occur on one side only, each preference list is strictly ordered or is a single tie, and

- Manlove et al. (2002): each tie is of length 2
- Irving-Manlove-O'Malley (2009): length of pref. lists ≤ 3
- Irving-Manlove-Scott (2008): master lists on both sides

McDermid (2009): MAX SMTI is approximable within $\frac{3}{2}$.

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

Restrictions, approximability, inapproximability

The problem is NP-hard even if ties occur on one side only, each preference list is strictly ordered or is a single tie, and

- Manlove et al. (2002): each tie is of length 2
- Irving-Manlove-O'Malley (2009): length of pref. lists ≤ 3
- Irving-Manlove-Scott (2008): master lists on both sides

McDermid (2009): MAX SMTI is approximable within $\frac{3}{2}$.

Yanagisawa (2007): MAX SMTI is not approximable within $\frac{33}{29}$ unless P=NP.

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

Restrictions, approximability, inapproximability

The problem is NP-hard even if ties occur on one side only, each preference list is strictly ordered or is a single tie, and

- Manlove et al. (2002): each tie is of length 2
- Irving-Manlove-O'Malley (2009): length of pref. lists ≤ 3
- Irving-Manlove-Scott (2008): master lists on both sides

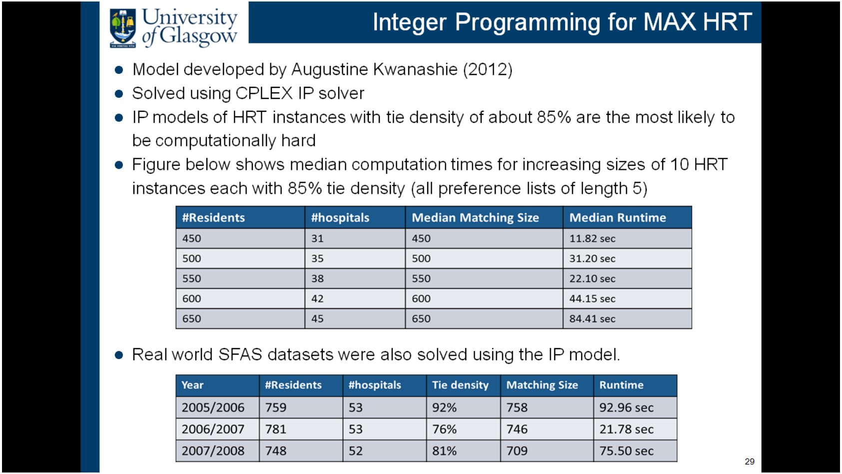
McDermid (2009): MAX SMTI is approximable within $\frac{3}{2}$.

Yanagisawa (2007): MAX SMTI is not approximable within $\frac{33}{29}$ unless P=NP.

Manlove-Irving (2009): Experiments with heuristics for random and real instances.

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

IPs on MAX-SMTI (David Manlove's talk)



University of Glasgow

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

29

- A. Kwanashie and D.F. Manlove. An Integer Programming approach to the Hospitals / Residents problem with Ties. To appear in Proceedings of OR 2013: the International Conference on Operations Research, Springer, 2014.

Finding 'almost stable' maximum size matchings

In many practical applications the first objective is to find a maximum size or complete matchings, and then they are concern with stability. e.g. for:

- US Navy
- United Nations World Food Programme

- P.A. Robards, Applying two-sided matching processes to the United States Navy enlisted assignment process, Master's Thesis, Naval Postgraduate School, Monterey, California, 2001.
- W. Yang, J.A. Giampapa, K. Sycara, Two-sided matching for the US Navy Detailing Process with market complication, Technical Report CMU-RI-TR-03-49, Robotics Institute, Carnegie-Mellon University, 2003.
- M. Soldner. Optimization and measurement in humanitarian operations: addressing practical needs. PhD Dissertation, 2014-07-02, Georgia Institute of Technology.

Finding 'almost stable' maximum size matchings

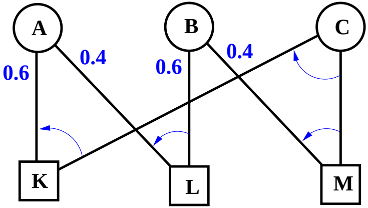
Biró-Manlove-Mittal (2010):

- Given a instance of stable marriage problem, finding a complete matching where the number of blocking pairs is minimised is NP-hard, and it is not approximable within $n^{1-\epsilon}$ for any $\epsilon > 0$ unless P=NP.
- For preference lists of length at most 3 on both sides, the problem is not approximable within $\frac{3557}{3556+2032\epsilon}$ for any ϵ, $(0 < \epsilon < \frac{1}{2032})$ unless P=NP.
- In the agents on one side has preference lists of size at most two then the problem is solvable in $O(n)$ time, where n is the number of men in the market.

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

Matching under uncertain preferences

Suppose that the preferences of the agents are uncertain.



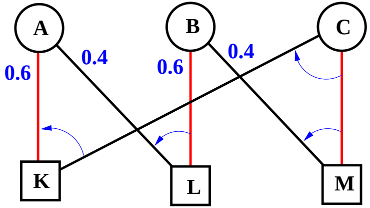
We may want to find a matching

- that is most likely to be stable
- where the expected number of blocking pairs is minimised

- P. Biró and B. Rastegari. Matching under uncertain preference. Working paper, 2014.

Matching under uncertain preferences

Suppose that the preferences of the agents are uncertain.



$P(\{AK, BL, CM\}$ is stable$)=0.36$

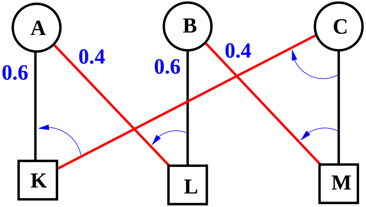
We may want to find a matching

- that is most likely to be stable
- where the expected number of blocking pairs is minimised

- P. Biró and B. Rastegari. Matching under uncertain preference. Working paper, 2014.

Matching under uncertain preferences

Suppose that the preferences of the agents are uncertain.



$P(\{AK, BL, CM\}$ is stable$)=0.36$
$P(\{AL, BM, CK\}$ is stable$)=0.4$

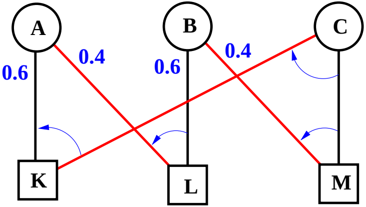
We may want to find a matching

- that is most likely to be stable
- where the expected number of blocking pairs is minimised

- P. Biró and B. Rastegari. Matching under uncertain preference. Working paper, 2014.

Matching under uncertain preferences

Suppose that the preferences of the agents are uncertain.



$P(\{AK, BL, CM\}$ is stable$)=0.36$
$P(\{AL, BM, CK\}$ is stable$)=0.4$

We may want to find a matching

- that is most likely to be stable
- where the expected number of blocking pairs is minimised

Biró-Rastegari (2014): Finding a matching that is most likely to be stable is NP-hard, even is uncertainty is resolved with uniform tie-breakings. (Implied by the inapproximability of MAX SMTI.)

- P. Biró and B. Rastegari. Matching under uncertain preference. Working paper, 2014.

3D Stable Matching problem (3DSM)

Knuth (1976):
"Problem 11. Can the stable-matching problem be generalized to three sets of objects (for example men, women and dogs)?"

3D Stable Matching problem (3DSM)

Knuth (1976):
"Problem 11. Can the stable-matching problem be generalized to three sets of objects (for example men, women and dogs)?"

Problem description:
- each agent has preference over all pairs from the two other sets.
- a **matching** is a set of disjoint families
- a matching is stable is there exists no blocking family
(that is preferred by all of its members to their current families)

3D Stable Matching problem (3DSM)

Knuth (1976):
"Problem 11. Can the stable-matching problem be generalized to three sets of objects (for example men, women and dogs)?"

Problem description:
- each agent has preference over all pairs from the two other sets.
- a **matching** is a set of disjoint families
- a matching is stable is there exists no blocking family
(that is preferred by all of its members to their current families)

Alkan (1988): Stable matching may not exist.

Ng and Hirschberg (1991): This problem is NP-complete.

Cyclic 3DSM

Ng and Hirschberg (1991): "cyclic preferences"
Men only care about women,
women only care about dogs and
dogs only care about men.

Cyclic 3DSM

Ng and Hirschberg (1991): "cyclic preferences"
Men only care about women,
women only care about dogs and
dogs only care about men.

Conjecture: If $|M| = |W| = |D|$ and **the lists are complete**, then stable matching always exists.

Cyclic 3DSM

Ng and Hirschberg (1991): "cyclic preferences"
Men only care about women,
women only care about dogs and
dogs only care about men.

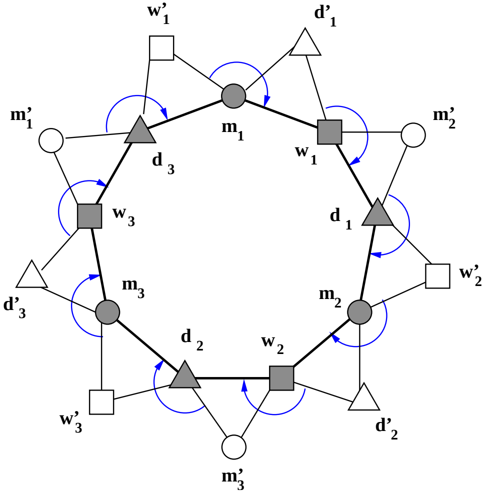
Conjecture: If $|M| = |W| = |D|$ and **the lists are complete**, then stable matching always exists.

Boros *et al.* (2004): This is true for 3×3 players.

Eriksson *et al.* (2006): True for 3×4 players as well...

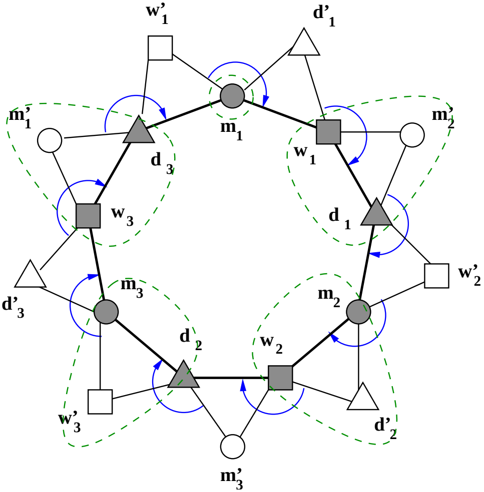
Cyclic 3DSMI: cyclic 3DSM with incomplete lists

Stable matching may not exist!
A counterexample for 3×6 players: $R6$



Cyclic 3DSMI: cyclic 3DSM with incomplete lists

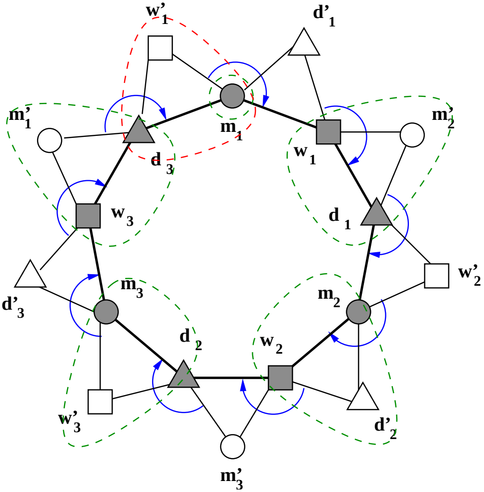
Stable matching may not exist!
A counterexample for 3 × 6 players: *R6*



- At least one inner player is unmatched

Cyclic 3DSMI: cyclic 3DSM with incomplete lists

Stable matching may not exist!
A counterexample for 3×6 players: *R6*



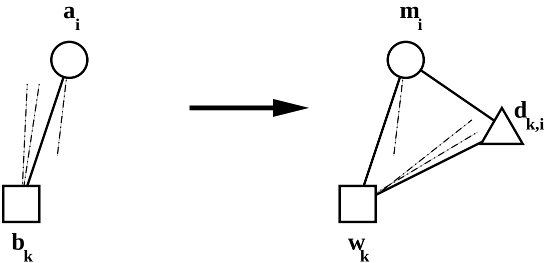
- At least one inner player is unmatched
- and is involved in a blocking cycle.

Cyclic 3DSMI is NP-complete

Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI

Cyclic 3DSMI is NP-complete

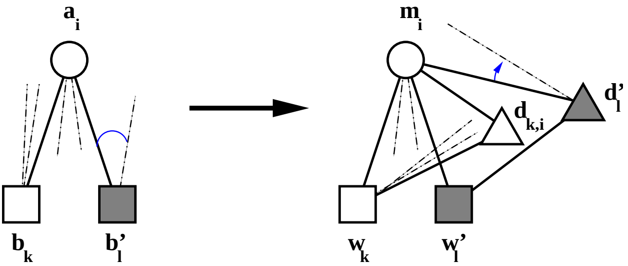
Sketch of the proof: COM SMTI $\implies$ cyclic 3DSMI



$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$

Cyclic 3DSMI is NP-complete

Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI

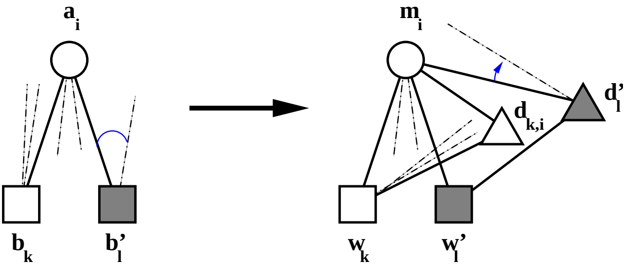


$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$

$$\{a_i, b'_l\} \in E(G) \iff (m_i, w'_l, d'_l) \in \mathcal{F}$$

Cyclic 3DSMI is NP-complete

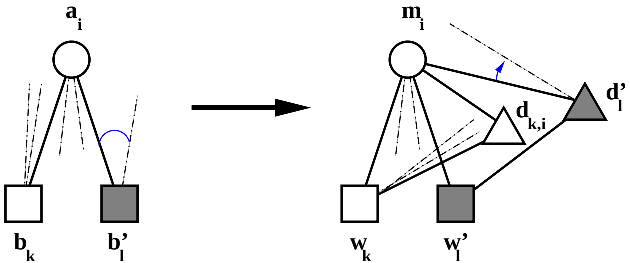
Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI



$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$
$$\{a_i, b'_l\} \in E(G) \iff (m_i, w'_l, d'_l) \in \mathcal{F}$$
$$M \subseteq E(G) \text{ matching} \iff F \subseteq \mathcal{F} \text{ 3D matching}$$

Cyclic 3DSMI is NP-complete

Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI

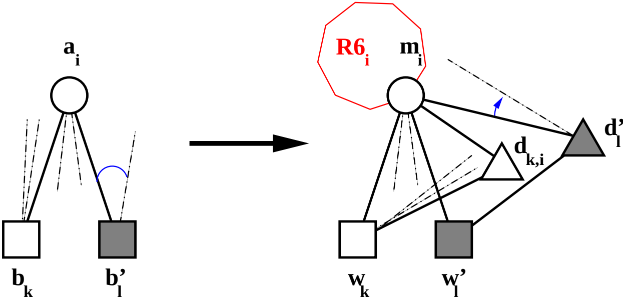


$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$
$$\{a_i, b'_l\} \in E(G) \iff (m_i, w'_l, d'_l) \in \mathcal{F}$$
$M \subseteq E(G)$ matching $\iff$ $F \subseteq \mathcal{F}$ 3D matching

weakly stable $\iff$ stable

Cyclic 3DSMI is NP-complete

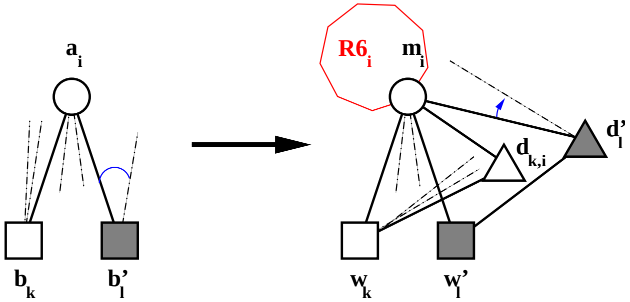
Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI



$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$
$$\{a_i, b'_l\} \in E(G) \iff (m_i, w'_l, d'_l) \in \mathcal{F}$$
$$M \subseteq E(G) \text{ matching} \iff F \subseteq \mathcal{F} \text{ 3D matching}$$
$$\text{weakly stable} \iff \text{stable}$$

Cyclic 3DSMI is NP-complete

Sketch of the proof: COM SMTI $\Longrightarrow$ cyclic 3DSMI



$$\{a_i, b_k\} \in E(G) \iff (m_i, w_k, d_{k,i}) \in \mathcal{F}$$

$$\{a_i, b'_l\} \in E(G) \iff (m_i, w'_l, d'_l) \in \mathcal{F}$$

$M \subseteq E(G)$ matching $\iff$ $F \subseteq \mathcal{F}$ 3D matching

weakly stable $\iff$ stable

M weakly stable and complete $\iff$ F stable

Summary of results

Biró-McDermid (2010): CYCLIC 3DSMI is NP-complete.

- P. Biró and E. McDermid. Three-sided stable matchings with cyclic preferences. Algorithmica 58, pp: 5–18, 2010.

Summary of results

Biró-McDermid (2010): CYCLIC 3DSMI is NP-complete.

A matching is strongly stable, if there exists no weakly blocking family (one player is strictly better off and nobody is worse off).

Biró-McDermid (2010): CYCLIC 3DSM is NP-complete under strong stability.

- P. Biró and E. McDermid. Three-sided stable matchings with cyclic preferences. Algorithmica 58, pp: 5–18, 2010.

Summary of results

Biró-McDermid (2010): CYCLIC 3DSMI is NP-complete.

A matching is strongly stable, if there exists no weakly blocking family (one player is strictly better off and nobody is worse off).

Biró-McDermid (2010): CYCLIC 3DSM is NP-complete under strong stability.

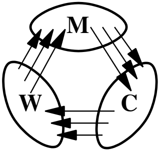
Summary of results:

	complete lists	incomplete lists
(weak) stability	???	NP-complete
strong stability	NP-complete	(NP-complete)

- P. Biró and E. McDermid. Three-sided stable matchings with cyclic preferences. Algorithmica 58, pp: 5–18, 2010.

Stable 3-way exchanges problem is NP-complete

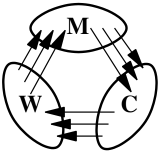
stable 3-way exchanges for a tripartite graph = CYCLIC 3DSMI



$V = M \cup W \cup D$ (i.e. men, women and dogs)
every arc $(i, j) \in A$ is from either
$W \times M$ or $D \times W$ or $M \times D$.

Stable 3-way exchanges problem is NP-complete

stable 3-way exchanges for a tripartite graph = CYCLIC 3DSMI

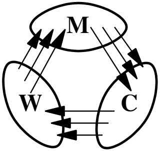


$V = M \cup W \cup D$ (i.e. men, women and dogs)
every arc $(i, j) \in A$ is from either
$W \times M$ or $D \times W$ or $M \times D$.

So the stable 2- and 3-way exchanges problem is also NP-complete.

Stable 3-way exchanges problem is NP-complete

stable 3-way exchanges for a tripartite graph = CYCLIC 3DSMI



$V = M \cup W \cup D$ (i.e. men, women and dogs)
every arc $(i, j) \in A$ is from either
$W \times M$ or $D \times W$ or $M \times D$.

So the stable 2- and 3-way exchanges problem is also NP-complete.

This situation can occur in the application: The set of M, W and D can correspond to patient-donor pairs with blood groups B-A, A-O and O-B, respectively.

Complexity of exchange problems: summary

		exchanges		
		pairwise		
maximum size/weight	does exist?	**yes**		
	hard to find?			
stable	does exist?			
	hard to find?			

Complexity of exchange problems: summary

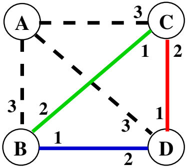
		exchanges		
		pairwise		
maximum size/weight	does exist?	yes		
	hard to find?	**P**		
stable	does exist?			
	hard to find?			

Edmonds (1967): Polynomial time algorithms for maximum size / maximum weight matching problem.

Complexity of exchange problems: summary

		exchanges		
		pairwise		
maximum size/weight	does exist?	yes		
	hard to find?	P		
stable	does exist?	**may not**		
	hard to find?			

stable pairwise exchange = stable roommates



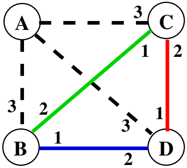
Gale and Shapley (1962):

Stable matching may not exist!

Complexity of exchange problems: summary

		exchanges		
		pairwise		
maximum size/weight	does exist?	yes		
	hard to find?	P		
stable	does exist?	may not		
	hard to find?	**P**		

stable pairwise exchange = stable roommates

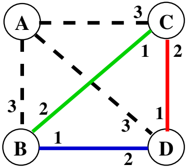


Gale and Shapley (1962):
Stable matching may not exist!
Irving (1985): A stable matching can be found in linear time, if one exists.

Complexity of exchange problems: summary

		exchanges		
		pairwise		
maximum size/weight	does exist?	yes		
	hard to find?	P		
stable	does exist?	may not		
	hard to find?	P		

stable pairwise exchange = stable roommates



Gale and Shapley (1962):
Stable matching may not exist!
Irving (1985): A stable matching can be found in linear time, if one exists.

Abraham-Biró-Manlove (2006): The problem of minimising the number of blocking pairs is NP-hard.

Complexity of exchange problems: summary

		exchanges		
		pairwise	**2-3-way**	
maximum size/weight	does exist?	yes	**yes**	
	hard to find?	P		
stable	does exist?	may not		
	hard to find?	P		

Complexity of exchange problems: summary

		exchanges		
		pairwise	2-3-way	
maximum size/weight	does exist?	yes	yes	
	hard to find?	P	**NP-hard**	
stable	does exist?	may not		
	hard to find?	P		

Abraham et al.; B.-Manlove-Rizzi: The problem of finding a maximum size/weight 2-3-way exchange is NP-complete.

Biró-Manlove-Rizzi: An $O(2^{\frac{m}{2}})$-time exact algorithm.
Implemented for UK Transplant.

Complexity of exchange problems: summary

		exchanges		
		pairwise	2-3-way	
maximum size/weight	does exist?	yes	yes	
	hard to find?	P	NP-hard	
stable	does exist?	may not	**may not**	
	hard to find?	P	**NPc**	

Abraham et al.; B.-Manlove-Rizzi: The problem of finding a maximum size/weight 2-3-way exchange is NP-complete.

Biró-Manlove-Rizzi: An $O(2^{\frac{m}{2}})$-time exact algorithm.
Implemented for UK Transplant.

B.-McDermid (2010): Stable 2-3-way exchange may not exist, and the related problem is NP-complete, even for tripartite graphs.

Complexity of exchange problems: summary

		exchanges		
		pairwise	2-3-way	**unbounded**
maximum size/weight	does exist?	yes	yes	**yes**
	hard to find?	P	NPc	
stable	does exist?	may not	may not	
	hard to find?	P	NPc	

Complexity of exchange problems: summary

		exchanges		
		pairwise	2-3-way	unbounded
maximum size/weight	does exist?	yes	yes	yes
	hard to find?	P	NPc	**P**
stable	does exist?	may not	may not	
	hard to find?	P	NPc	

Graph Theory folklore: The problem of finding a maximum size/weight (unbounded) exchange is P-time solvable.

Complexity of exchange problems: summary

		exchanges		
		pairwise	2-3-way	unbounded
maximum size/weight	does exist?	yes	yes	yes
	hard to find?	P	NPc	P
stable	does exist?	may not	may not	**yes**
	hard to find?	P	NPc	**P**

Graph Theory folklore: The problem of finding a maximum size/weight (unbounded) exchange is P-time solvable.

Scarf-Shapley (1972): Stable exchange always exists. A solution can be found by the Top Trading Cycle algorithm of Gale.

Hungarian higher education matching scheme

Special features:

1. ties
2. lower quotas
3. common quotas
4. paired applications

Theory: Each of the 2.-4. features makes the problem of finding a 'good' solution NP-hard, so heuristics are used...

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).
- P. Biró and S. Kiselgof. College admissions with stable score-limits. To appear in Central European Journal of Operations Research, 2015.
- P. Biró, and I. McBride. Integer programming methods for special college admissions problems. In Proceedings of COCOA 2014: the 8th Annual International Conference on Combinatorial Optimization and Applications, volume 8881 of LNCS, pages 429-443, Springer, 2014.

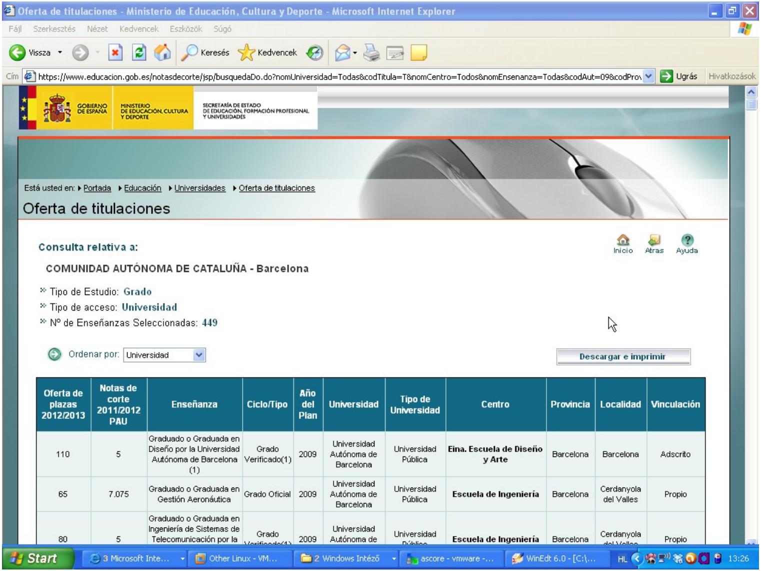
Stable matchings and score-limits

Basic admission mechanism (used in many countries):

- colleges set their quotas (over their programmes)
- applicants submit their strict preferences over the colleges
- colleges rank their applicants according to their scores
- central coordinator announces the **score-limits**
- **induced matching**: each student is admitted to the first college in her list where she achieved the score-limit

A set of score-limits is **stable** IFF the induced matching is stable

Score-limits in Spain



Oferta de titulaciones - Ministerio de Educación, Cultura y Deporte - Microsoft Internet Explorer

https://www.educacion.gob.es/notasdecorte/jsp/busquedaDo.do?nomUniversidad=Todas&codTitula=T&nomCentro=Todos&nomEnsenanza=Todas&codAut=09&codPro

GOBIERNO DE ESPAÑA | MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE | SECRETARÍA DE ESTADO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y UNIVERSIDADES

Está usted en: Portada › Educación › Universidades › Oferta de titulaciones

Oferta de titulaciones

Inicio | Atras | Ayuda

Consulta relativa a:

COMUNIDAD AUTÓNOMA DE CATALUÑA - Barcelona

- Tipo de Estudio: **Grado**
- Tipo de acceso: **Universidad**
- Nº de Enseñanzas Seleccionadas: **449**

Ordenar por: Universidad

Descargar e imprimir

Oferta de plazas 2012/2013	Notas de corte 2011/2012 PAU	Enseñanza	Ciclo/Tipo	Año del Plan	Universidad	Tipo de Universidad	Centro	Provincia	Localidad	Vinculación
110	5	Graduado o Graduada en Diseño por la Universidad Autónoma de Barcelona (1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Eina. Escuela de Diseño y Arte**	Barcelona	Barcelona	Adscrito
65	7.075	Graduado o Graduada en Gestión Aeronáutica	Grado Oficial	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela de Ingeniería**	Barcelona	Cerdanyola del Valles	Propio
80	5	Graduado o Graduada en Ingeniería de Sistemas de Telecomunicación por la	Grado	2009	Universidad Autónoma de	Universidad	**Escuela de Ingeniería**	Barcelona	Cerdanyola	Propio

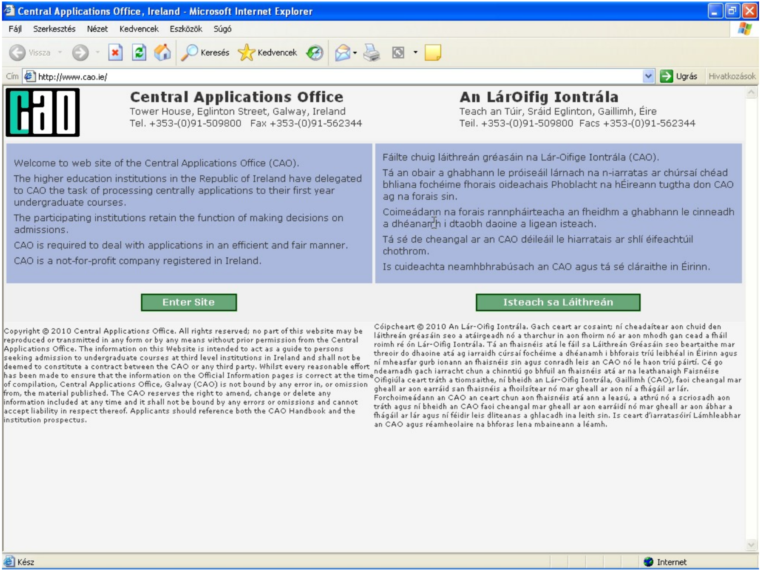
Score-limits in Spain

Oferta de titulaciones - Ministerio de Educación, Cultura y Deporte - Microsoft Internet Explorer

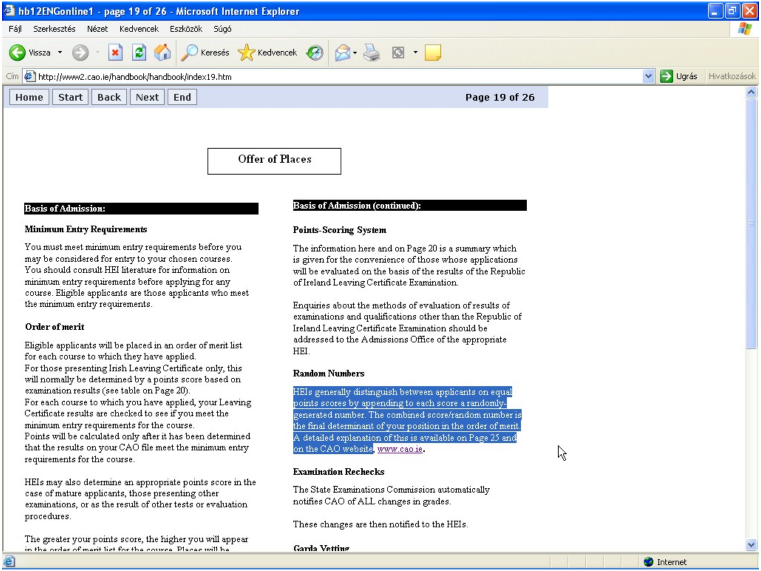
https://www.educacion.gob.es/notasdecorte/jsp/busquedaDo.do?nomUniversidad=Todas&codTitula=T&nomCentro=Todos&nomEnsenanza=Todas&codAut=09&codProv

		Universidad Autónoma de Barcelona(1)			Barcelona					
80	5	Graduado o Graduada en Ingeniería Electrónica de Telecomunicación por la Universidad Autónoma de Barcelona(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela de Ingeniería**	Barcelona	Cerdanyola del Valles	Propio
300	5	Graduado o Graduada en Ingeniería Informática(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela de Ingeniería**	Barcelona	Cerdanyola del Valles	Propio
80	5.07	Graduado o Graduada en Ingeniería Química(1)	Grado Verificado(1)	2011	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela de Ingeniería**	Barcelona	Cerdanyola del Valles	Propio
80	5	Graduado o Graduada en Prevención y Seguridad Integral	Grado Oficial	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela de Prevención y Seguridad Integral (EPSI)**	Barcelona	Cerdanyola del Valles	Adscrito
90	5.098	Graduado o Graduada en Artes y Diseño por la Universidad Autónoma de Barcelona(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Massana. Centro Municipal de Arte y Diseño**	Barcelona	Barcelona	Adscrito
120	5.022	Graduado o Graduada en Enfermería por la Universidad Autónoma de Barcelona	Grado Oficial	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias de la Salud de Manresa**	Barcelona	Manresa	Adscrito
60	5	Graduado o Graduada en Logopedia por la Universidad Autónoma de Barcelona	Grado Oficial	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias de la Salud de Manresa**	Barcelona	Manresa	Adscrito
160	5	Graduado o Graduada en Fisioterapia por la Universidad Autónoma de Barcelona(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias de la Salud de Manresa**	Barcelona	Manresa	Adscrito
60	5	Graduado o Graduada en Podología(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias de la Salud de Manresa**	Barcelona	Manresa	Adscrito
80	5	Graduado o Graduada en Educación Infantil	Grado Oficial	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias Sociales**	Barcelona	Manresa	Adscrito
80	5	Graduado o Graduada en Gestión de Empresas(1)	Grado Verificado(1)	2009	Universidad Autónoma de Barcelona	Universidad Pública	**Escuela Universitaria de Ciencias Sociales**	Barcelona	Manresa	Adscrito

Score-limits in Ireland



Score-limits in Ireland



Offer of Places

Basis of Admission:

Minimum Entry Requirements

You must meet minimum entry requirements before you may be considered for entry to your chosen courses. You should consult HEI literature for information on minimum entry requirements before applying for any course. Eligible applicants are those applicants who meet the minimum entry requirements.

Order of merit

Eligible applicants will be placed in an order of merit list for each course to which they have applied. For those presenting Irish Leaving Certificate only, this will normally be determined by a points score based on examination results (see table on Page 20). For each course to which you have applied, your Leaving Certificate results are checked to see if you meet the minimum entry requirements for the course. Points will be calculated only after it has been determined that the results on your CAO file meet the minimum entry requirements for the course.

HEIs may also determine an appropriate points score in the case of mature applicants, those presenting other examinations, or as the result of other tests or evaluation procedures.

The greater your points score, the higher you will appear in the order of merit list for the course. Places will be

Basis of Admission (continued):

Points-Scoring System

The information here and on Page 20 is a summary which is given for the convenience of those whose applications will be evaluated on the basis of the results of the Republic of Ireland Leaving Certificate Examination.

Enquiries about the methods of evaluation of results of examinations and qualifications other than the Republic of Ireland Leaving Certificate Examination should be addressed to the Admissions Office of the appropriate HEI.

Random Numbers

HEIs generally distinguish between applicants on equal points scores by appending to each score a randomly-generated number. The combined score/random number is the final determinant of your position in the order of merit. A detailed explanation of this is available on Page 25 and on the CAO website, www.cao.ie.

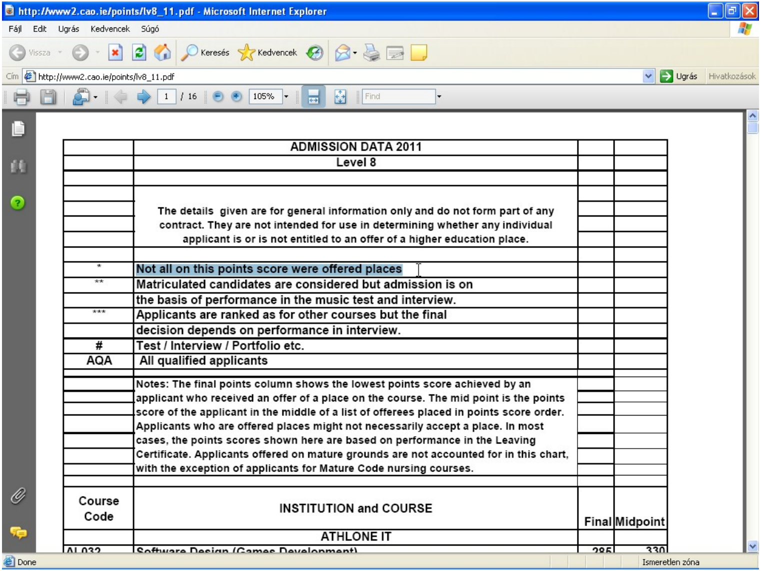
Examination Rechecks

The State Examinations Commission automatically notifies CAO of ALL changes in grades.

These changes are then notified to the HEIs.

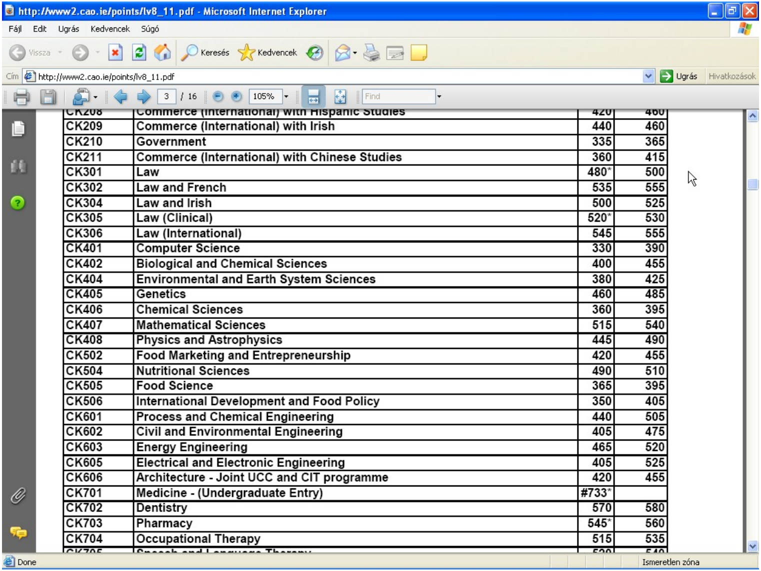
Garda Vetting

Score-limits in Ireland



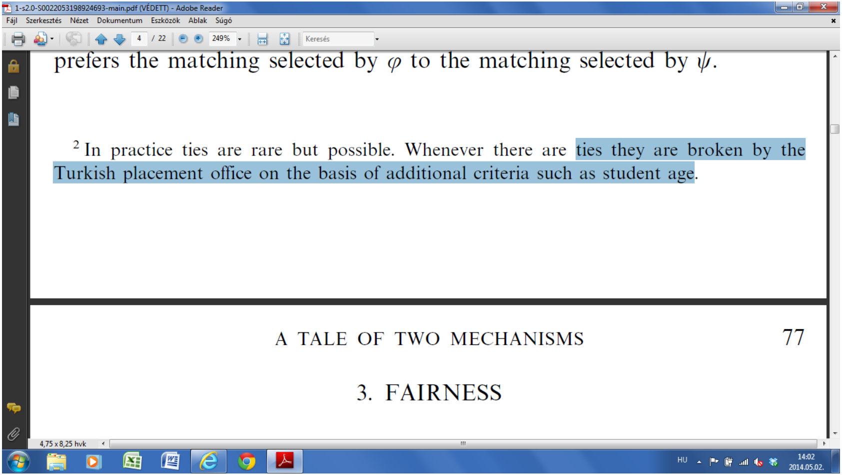
	ADMISSION DATA 2011		
	Level 8		
	The details given are for general information only and do not form part of any contract. They are not intended for use in determining whether any individual applicant is or is not entitled to an offer of a higher education place.		
*	Not all on this points score were offered places		
**	Matriculated candidates are considered but admission is on the basis of performance in the music test and interview.		
***	Applicants are ranked as for other courses but the final decision depends on performance in interview.		
#	Test / Interview / Portfolio etc.		
AQA	All qualified applicants		
	Notes: The final points column shows the lowest points score achieved by an applicant who received an offer of a place on the course. The mid point is the points score of the applicant in the middle of a list of offerees placed in points score order. Applicants who are offered places might not necessarily accept a place. In most cases, the points scores shown here are based on performance in the Leaving Certificate. Applicants offered on mature grounds are not accounted for in this chart, with the exception of applicants for Mature Code nursing courses.		
Course Code	INSTITUTION and COURSE	Final	Midpoint
	ATHLONE IT		
AL032	Software Design (Games Development)	285	330

Score-limits in Ireland



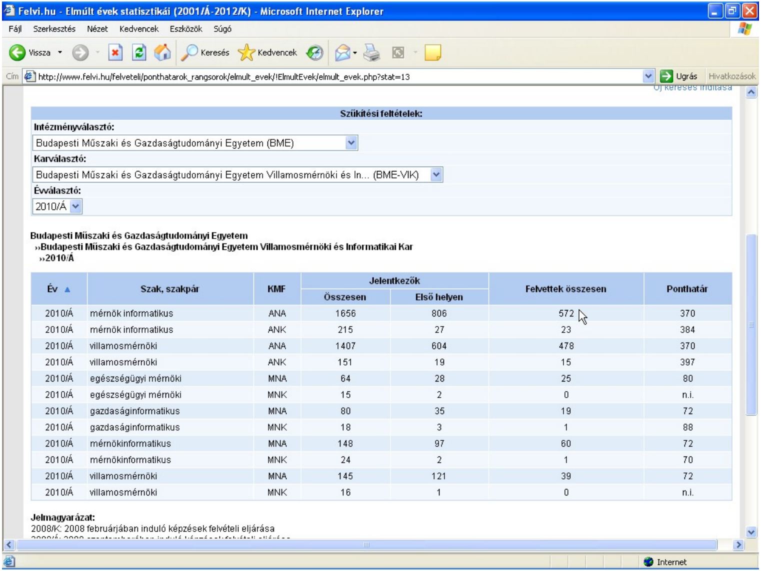
CK208	Commerce (International) with Hispanic Studies	420	460
CK209	Commerce (International) with Irish	440	460
CK210	Government	335	365
CK211	Commerce (International) with Chinese Studies	360	415
CK301	Law	480*	500
CK302	Law and French	535	555
CK304	Law and Irish	500	525
CK305	Law (Clinical)	520*	530
CK306	Law (International)	545	555
CK401	Computer Science	330	390
CK402	Biological and Chemical Sciences	400	455
CK404	Environmental and Earth System Sciences	380	425
CK405	Genetics	460	485
CK406	Chemical Sciences	360	395
CK407	Mathematical Sciences	515	540
CK408	Physics and Astrophysics	445	490
CK502	Food Marketing and Entrepreneurship	420	455
CK504	Nutritional Sciences	490	510
CK505	Food Science	365	395
CK506	International Development and Food Policy	350	405
CK601	Process and Chemical Engineering	440	505
CK602	Civil and Environmental Engineering	405	475
CK603	Energy Engineering	465	520
CK605	Electrical and Electronic Engineering	405	525
CK606	Architecture - Joint UCC and CIT programme	420	455
CK701	Medicine - (Undergraduate Entry)	#733*	
CK702	Dentistry	570	580
CK703	Pharmacy	545*	560
CK704	Occupational Therapy	515	535
CK705	Speech and Language Therapy	520	540

Score-limits in Turkey



- M. Balinski and T. Sönmez. A Tale of Two Mechanisms: Student Placement. Journal of Economic Theory 84, 73-94 (1999)

Score-limits in Hungary



Szűkítési feltételek:

Intézményválasztó: Budapesti Műszaki és Gazdaságtudományi Egyetem (BME)

Karválasztó: Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és In... (BME-VIK)

Évválasztó: 2010/Á

Budapesti Műszaki és Gazdaságtudományi Egyetem
»Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar
»2010/Á

Év ▲	Szak, szakpár	KMF	Jelentkezők		Felvettek összesen	Ponthatár
			Összesen	Első helyen		
2010/Á	mérnök informatikus	ANA	1656	806	572	370
2010/Á	mérnök informatikus	ANK	215	27	23	384
2010/Á	villamosmérnöki	ANA	1407	604	478	370
2010/Á	villamosmérnöki	ANK	151	19	15	397
2010/Á	egészségügyi mérnöki	MNA	64	28	25	80
2010/Á	egészségügyi mérnöki	MNK	15	2	0	n.i.
2010/Á	gazdaságinformatikus	MNA	80	35	19	72
2010/Á	gazdaságinformatikus	MNK	18	3	1	88
2010/Á	mérnökinformatikus	MNA	148	97	60	72
2010/Á	mérnökinformatikus	MNK	24	2	1	70
2010/Á	villamosmérnöki	MNA	145	121	39	72
2010/Á	villamosmérnöki	MNK	16	1	0	n.i.

Jelmagyarázat:
2008/K: 2008 februárjában induló képzések felvételi eljárása

Basic IP model for the College Admissions problem

Feasibility constraints:

$$\sum_{j:(a_i,c_j)\in E} x_{ij} \leq 1 \text{ for each } a_i \in A$$

$$\sum_{i:(a_i,c_j)\in E} x_{ij} \leq u_j \text{ for each } c_j \in C$$

Stability constraints:

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right) \cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}>s_{ij}} x_{hj} \geq u_j \text{ for each } (a_i, c_j) \in E$$

Where x_{ij} is a binary variable representing the application (a_i, c_j),
r_{ij} is the rank of the application to c_j in a_i's list, and
s_{ij} is the score of a_i at c_j.

Basic IP for the College Admissions problem

Remark 1: We can get an applicant-optimal (resp. an applicant-pessimal) stable solution by setting the objective function of the IP as the minimum (resp. maximum) of the following term:

$$\sum_{(a_i,c_j)\in E} r_{ij} \cdot x_{ij}$$

Remark 2: When we have ties in the priorities (due to equal scores), then the following modified stability constraints (together with the feasibility constraints) lead to *weakly stable* matchings:

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right) \cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}\geq s_{ij}} x_{hj} \geq u_j \text{ for each } (a_i,c_j) \in E$$

Alternative stability conditions with score-limits

In addition to the feasibility constraints, we define a **score-limit** $0 \leq t_j \leq \bar{s} + 1$ for each college c_j, and we link these score-limits to the matching with the following constraints:

$$t_j \leq (1 - x_{ij}) \cdot (\bar{s} + 1) + s_{ij} \text{ for each } (a_i, c_j) \in E$$

and

$$s_{ij} + 1 \leq t_j + \left(\sum_{k: r_{ik} \leq r_{ij}} x_{ik} \right) \cdot (\bar{s} + 1) \text{ for each } (a_i, c_j) \in E$$

Implying that **each applicant is assigned to the best college where she achieved the score-limit**

Alternative stability conditions with score-limits

In addition to the feasibility constraints, we define a **score-limit** $0 \le t_j \le \bar{s}+1$ for each college c_j, and we link these score-limits to the matching with the following constraints:

$$t_j \le (1 - x_{ij}) \cdot (\bar{s}+1) + s_{ij} \text{ for each } (a_i, c_j) \in E$$

and

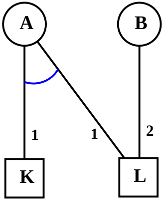
$$s_{ij} + 1 \le t_j + \left(\sum_{k: r_{ik} \le r_{ij}} x_{ik} \right) \cdot (\bar{s}+1) \text{ for each } (a_i, c_j) \in E$$

The **stability condition** can be replaced by either of the followings:

1. each unfilled college has score-limit zero
2. no college can decrease its score-limit without violating its quota
3. adding the following objective function:

$$\min \sum_{j=1...m} t_j$$

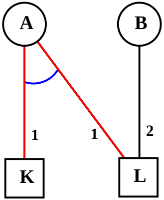
Special feature 1: ties with equal treatment policy.



- Students with the same score at some college
- Either all or none of them are admitted

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

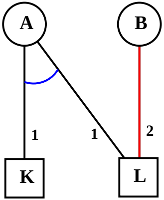
Special feature 1: ties with equal treatment policy.



- Students with the same score at some college
- Either all or none of them are admitted

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

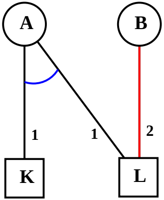
Special feature 1: ties with equal treatment policy.



- Students with the same score at some college
- Either all or none of them are admitted

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

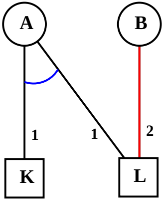
Special feature 1: ties with equal treatment policy.



- Students with the same score at some college
- Either all or none of them are admitted
- **Stable score-limits:** No score-limit can be decreased at any college without violating its quota. (So the last tied group is rejected!)

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

Special feature 1: ties with equal treatment policy.

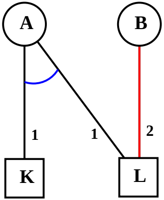


- Students with the same score at some college
- Either all or none of them are admitted
- **Stable score-limits:** No score-limit can be decreased at any college without violating its quota. (So the last tied group is rejected!)

Biró (2007): The generalised student / college-oriented GS algorithms produce student-optimal / pessimal stable score-limits efficiently.

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

Special feature 1: ties with equal treatment policy.



- Students with the same score at some college
- Either all or none of them are admitted
- **Stable score-limits:** No score-limit can be decreased at any college without violating its quota. (So the last tied group is rejected!)

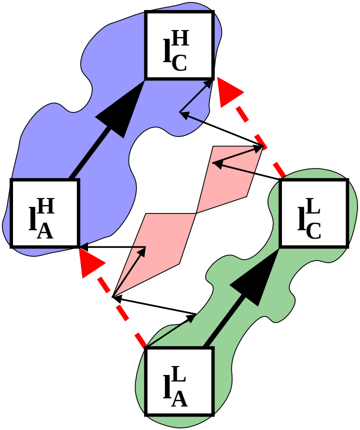
Biró (2007): The generalised student / college-oriented GS algorithms produce student-optimal / pessimal stable score-limits efficiently.

In Hungary the college-oriented version has been replaced by the applicant-oriented version in 2007.

- P. Biró. Student Admissions in Hungary as Gale and Shapley Envisaged. Technical Report. Dept of Computing Science, University of Glasgow, TR-2008-291.

Stable score-limits under different policies

- higher stable: equal treatment, where no quota is violated (used in Hungary)
- breaking ties with lottery
- lower stable: equal treatment, where the quota may be violated with the last tied group (used in Chile)



- P. Biró and S. Kiselgof. College admissions with stable score-limits. To appear in Central European Journal of Operations Research, 2015.
- I. Rios, T. Larroucau, G. Parra and R. Cominetti. College Admissions Problem with Ties and Flexible Quotas. Working paper, 2014.
- T. Fleiner and Zs. Jankó. Choice Function-Based Two-Sided Markets: Stability, Lattice Property, Path Independence and Algorithms. Algorithms 7(1), 32-59 (2014)

College Admissions with ties: stable score-limits

In addition to the feasibility constraints, we define a score-limit $0 \leq t_j \leq \bar{s} + 1$ for each college c_j, and the following constraints:

$$t_j \leq (1 - x_{ij}) \cdot (\bar{s} + 1) + s_{ij} \text{ for each } (a_i, c_j) \in E$$

and

$$s_{ij} + 1 \leq t_j + \left(\sum_{k: r_{ik} \leq r_{ij}} x_{ik} \right) \cdot (\bar{s} + 1) \text{ for each } (a_i, c_j) \in E$$

together with a set of constraints implying that **no college can decrease its score-limit without violating its quota**.

College Admissions with ties: stable score-limits

In addition to the feasibility constraints, we define a score-limit $0 \leq t_j \leq \bar{s} + 1$ for each college c_j, and the following constraints:

$$t_j \leq (1 - x_{ij}) \cdot (\bar{s} + 1) + s_{ij} \text{ for each } (a_i, c_j) \in E$$

and

$$s_{ij} + 1 \leq t_j + \left(\sum_{k: r_{ik} \leq r_{ij}} x_{ik} \right) \cdot (\bar{s} + 1) \text{ for each } (a_i, c_j) \in E$$

together with a set of constraints implying that **no college can decrease its score-limit without violating its quota**.

OR with the following objective function:

$$\min \sum_{j=1 \ldots m} t_j$$

Special feature 2: lower quotas

Suppose that college c_j has lower quota l_j and upper quota u_j.

A solution is a matching, where each college c_j has either
- no assignees ("closed college") or
- at least l_j and at most u_j assignees ("open college").

Special feature 2: lower quotas

Suppose that college c_j has lower quota l_j and upper quota u_j.

A solution is a matching, where each college c_j has either
- no assignees ("closed college") or
- at least l_j and at most u_j assignees ("open college").

A matching is stable is there exist no
- "blocking pair", consisting of an open college and an unsatisfied applicant,
- "blocking coalition", consisting of a closed college c_j and l_j unsatisfied applicants.

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	Saxophone	Trumpet
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant: 2nd applicant:	Adam Bill	Adam Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

B.-Fleiner-Irving-Manlove (2010): Stable matching may not exist, and the related decision problem is NP-complete.

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

Special feature 2: lower quotas

Studies:	**Saxophone**	**Trumpet**
lower and upper quotas	$1 \leq \cdots \leq 1$	$2 \leq \cdots \leq 2$
1st applicant:	Adam	Adam
2nd applicant:	Bill	Bill

Adam's list: Trumpet, Saxophone
Bill's list: Saxophone, Trumpet

B.-Fleiner-Irving-Manlove (2010): Stable matching may not exist, and the related decision problem is NP-complete.

A natural heuristic is used in Hungary.

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

College Admissions with lower quotas: IP model

$o_j \in \{0,1\}$ is the indicator variable showing whether c_j is open.
New feasibility constraint:

$$o_j \cdot l_j \leq \sum_{i:(a_i,c_j)\in E} x_{ij} \leq o_j \cdot u_j \text{ for each } c_j \in C$$

Pairwise stability for open colleges:

$$\left(\sum_{k:r_{ik}\leq r_{ij}} x_{ik}\right) \cdot u_j + \sum_{h:(a_h,c_j)\in E, s_{hj}>s_{ij}} x_{hj} \geq o_j \cdot u_j \text{ for each } (a_i,c_j) \in E$$

group-stability for closed colleges:

$$\sum_{i:(a_i,c_j)\in E} \left[1 - \sum_{k:r_{ik}<r_{ij}} x_{ik}\right] \leq (1-o_j)\cdot(l_j-1) + o_j \cdot n \text{ for each } c_j \in C$$

Some lemmas that can speed up the solver

$c_1, c_2, \ldots, c_{m-k}, \; c_{m-k+1}, \ldots, c_m$

Lemma 1: The colleges that reach their lower quotas in the stable solutions of a College Admissions problem with no lower quotas must be open in every stable solution where lower quotas are respected.

Lemma 2: Suppose that X is the set of colleges that do not reach their lower quotas in the stable solutions with no lower quotas. Given a college c_j of X, if all the colleges in X but c_j are closed and c_j still does not achieve its lower quota then c_j must be closed in any stable solution with lower quotas.

and then **we can repeat this filtering process...**

Special feature 3: common quotas

Some set of colleges may have a common quota.
No common quota may be exceeded in a feasible matching.

Special feature 3: common quotas

Some set of colleges may have a common quota.
No common quota may be exceeded in a feasible matching.

The stability of a matching:
If an applicant a_i is not matched to a college c_j, then
– either a_i is matched to a better college
– or c_j has filled its quota with better applicants than a_i
– or there is a set of colleges C_p such that $c_j \in C_p$ and C_p filled its quota with better applicants.

Special feature 3: common quotas

Studies:	**p. CS_{BME}**	**s. CS_{BME}**	...	**s. CS_{GD}**	...
c. quotas:		CS national quota: $\leq$3000			
quotas:	$\lesssim 50$	$\lesssim 450$	...	$\lesssim 400$	...
2004:	49 (78p)	474 (113p)	...	336 (74p)	...
2005:	51 (90p)	423 (126p)	...	369 (77p)	...
2006:	41 (80p)	443 (125p)	...	321 (78p)	...
2007:	51 (100p)	478 (120p)	...	246 (79p)	...

Special feature 3: common quotas

Studies:	p. CS_{BME}	s. CS_{BME}	...	s. CS_{GD}	...
c. quotas:		CS national quota: $\leq$3000			
quotas:	$\leq$ 50	$\leq$ 450	...	$\leq$ 400	...
2004:	49 (78p)	474 (113p)	...	336 (74p)	...
2005:	51 (90p)	423 (126p)	...	369 (77p)	...
2006:	41 (80p)	443 (125p)	...	321 (78p)	...
2007:	51 (100p)	478 (120p)	...	246 (79p)	...

Studies:	p. CS_{BME}	s. CS_{BME}	...	s. CS_{GD}	...
c. quotas:		CS national quota: $\leq$3000			
c. quotas:	faculty quota: $\leq$500		...	$\leq$ 400	...
2008:	8 (365p)	492 (366p)	...	165 (160p)	...
2009:	16 (365p)	583 (373p)	...	183 (224p)	...
2010:	23 (384p)	572 (370p)	...	241 (206p)	...
2011:	24 (372p)	573 (370p)	...	356 (200p)	...
2012:	35 (396p)	578 (370p)	...	40 (240p)	...
2013:	42 (382p)	519 (370p)	...	33 (240p)	...

CA with common quotas: theoretical findings

B.-Fleiner-Irving-Manlove (2010): For **nested set systems**, stable matching always exists and it can be obtained by generalised Gale-Shapley type algorithms. Moreover, the applicant / college -oriented versions produce the best / worst possible stable matchings for the applicants.

Otherwise, stable matching may not exist, and the related decision problem is NP-complete.

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

CA with common quotas: theoretical findings

B.-Fleiner-Irving-Manlove (2010): For **nested set systems**, stable matching always exists and it can be obtained by generalised Gale-Shapley type algorithms. Moreover, the applicant / college -oriented versions produce the best / worst possible stable matchings for the applicants.

Otherwise, stable matching may not exist, and the related decision problem is NP-complete.

The set system had been nested in Hungary until 2007, but became non-nested in 2008 with the possibility that no stable solution exists, and the related decision problem being NP-hard. So, heuristics are used...

- P. Biró, T.Fleiner, R.W. Irving and D.F. Manlove. The College Admissions problem with lower and common quotas. Theoretical Computer Science 411, 3136-3153 (2010).

IP for CA with common quotas

Let u_p be a common upper quota for C_p and t_p a corresponding score-limit. Additional feasibility constraint:

$$\sum_{i:(a_i,c_j)\in E, c_j \in C_p} x_{ij} \le u_p \text{ for each } C_p \subseteq C$$

Stability:

$$t_p \le (1 - x_{ij}) \cdot (\bar{s} + 1) + s_{ij} \text{ for each } (a_i, c_j) \in E \text{ and } c_j \in C_p$$

and

$$s_{ij} + 1 \le t_p + \left(\sum_{k:r_{ik} \le r_{ij}} x_{ik} + y_i^p \right) \cdot (\bar{s} + 1) \text{ for each } (a_i, c_j) \in E \text{ and } c_j \in C_p$$

with

$$\sum_{p:c_j \in C_p} y_i^p \le q_j - 1 \text{ for each } (a_i, c_j) \in E$$

where $y_i^p \in \{0, 1\}$ and q_j is the number of sets c_j is involved in.

Special feature 4: paired applications

Students may apply for pair of programmes (these are special programmes for teachers). In 2010: 5,578 students applied for teachers' programmes, and 2,091 of them applied for pair of programmes...

This is like the Hospitals Residents problems with couples! Ronn's 1990 theorem implies NP-hardness here as well.

Integer programming techniques used for market design

Many papers on auctions and allocation problems

- N. Nisan. Bidding and allocation in combinatorial auctions. In Proceedings of ACM-EC 2000.
- E. Budish, A. Othman and T. Sandholm. Finding Approximate Competitive Equilibria: Efficient and Fair Course Allocation. In Proceedings of AAMAS 2010.
- N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn, and J. Mestre. Assigning Papers to Referees. Algorithmica, 58(1):119-136 (2010).

Most kidney exchange applications are based on IP techniques

- A.E. Roth, T. Sönmez and M.U. Ünver. Efficient Kidney Exchange: Coincidence of Wants in Markets with Compatibility-Based Preferences. American Economic Review, 97(3), 828-851 (2007).
- D. Abraham, A. Blum and T. Sandholm. Clearing Algorithms for Barter-Exchange Markets: Enabling Nationwide Kidney Exchanges. In Proceedings of ACM-EC 2007.
- D.F. Manlove and G. O'Malley. Paired and altruistic kidney donation in the UK: Algorithms and experimentation. In Proceedings of SEA 2012.

Recent papers on IP methods for stable matching problems

- A. Kwanashie and D.F. Manlove. An Integer Programming approach to the Hospitals / Residents problem with Ties. Proceedings of OR 2013, Springer, pp: 263–269, 2014.
- P. Biró, I. McBride and D.F. Manlove. The Hospitals / Residents problem with Couples: Complexity and Integer Programming models. Proceedings of SEA 2014, vol. 8504 of LNCS, pp: 10–21, 2014.

Integer programming for solving the Hungarian case

What we have done in this paper:

- We formulated IPs to solve the problems for each of the four special features
- We investigated some combination of these special features
- We established new lemmas to speed up the solutions

Future plans:

- To integrate the IPs into a single one that can be used to solve the real application
- Implement and test the IPs on a real data from 2008, Hungary
- Other applications? E.g.
  - resident allocation with regional caps
  - controlled school choice

- P. Biró, and I. McBride. Integer programming methods for special college admissions problems. In Proceedings of COCOA 2014: the 8th Annual International Conference on Combinatorial Optimization and Applications, volume 8881 of LNCS, pages 429-443, Springer, 2014

Computational complexity in mechanism design

Why is this aspect interesting?

- because the computational complexity of the underlying matching problems is crucial in the solvability of **practical applications**
- sometimes we can **avoid** the computationally hard problems when designing the market
- if we cannot avoid the hard problems, **algorithm/optimisation theory** still provides many tools to analyse and solve them...

Further references

New book on the algorithmic aspects:
David F. Manlove: Algorithmics of matching under preferences. World Scientific, 2013.

Summer school talks by Manlove and others:
http://econ.core.hu/english/res/MatchingSchool.html

COST Action on Computational Social Choice:
http://www.illc.uva.nl/COST-IC1205/

The Matching in Practice network website:
http://www.matching-in-practice.eu/

My research website:
http://www.cs.bme.hu/∼pbiro/research.html