Optimising paired and pooled kidney exchanges

Péter Biró, Kirstin MacDonald and David Manlove

Department of Computing Science
University of Glasgow

Supported by EPSRC grant EP/E011993/1

Acknowledgements

- Péter Biró
- Katarína Cechlárová
- Rachel Johnson and Joanne Allen, UK Transplant (UKT)

Acknowledgements

- Péter Biró
- Katarína Cechlárová
- Rachel Johnson and Joanne Allen, UK Transplant (UKT)

Kidney failure

- Treatment
  - Dialysis
  - Transplantation

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03
    - *Source: UKT*

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03
    - *Source: UKT*
  - Living donors

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03
    - *Source: UKT*
  - Living donors
    - 2007: 36% of all kidney transplants from living donors (*Source: UKT*), but...

Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03
    - *Source: UKT*
  - Living donors
    - 2007: 36% of all kidney transplants from living donors (*Source: UKT*), but...
    - Blood type incompatibility (e.g., A $\not\longrightarrow$ B)

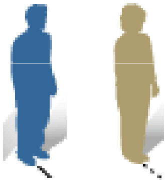
Kidney failure

- Treatment
  - Dialysis
  - Transplantation
- Need for donors
  - Deceased donors
    - UK: 6980 patients on transplant list, 1453 transplants carried out (2007-08)
    - Median waiting time: 841 days (adults), 164 days (children) based on patient registrations during 2000-03
    - *Source: UKT*
  - Living donors
    - 2007: 36% of all kidney transplants from living donors (*Source: UKT*), but...
    - Blood type incompatibility (e.g., A $\not\longrightarrow$ B)
    - Positive crossmatch (tissue-type incompatibility)

Pairwise exchange: Portsmouth / Plymouth, December 2007

Donald Planner, 61

Suzanne Wills, 43

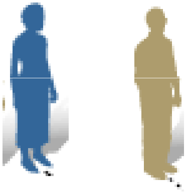


Father / daughter

Incompatible blood type

Margaret Wearn ,56

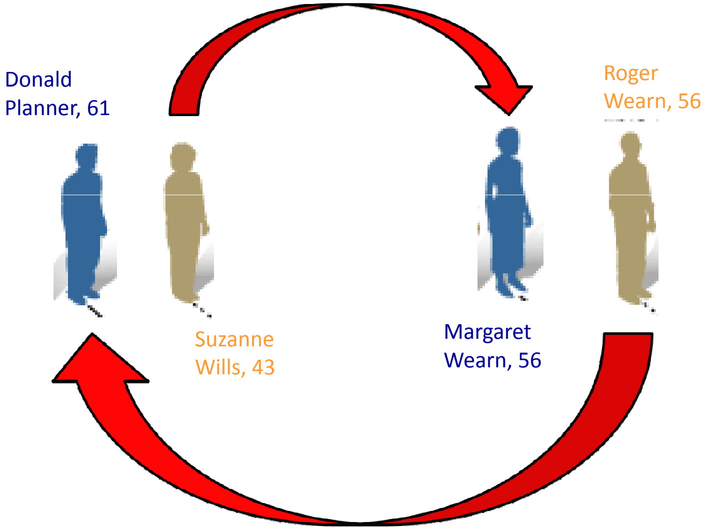
Roger Wearn, 56



Married

Positive crossmatch

Pairwise exchange: Portsmouth / Plymouth, December 2007



Publicity

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 lly reliant on the dialysis

Exchange between three pairs: Johns Hopkins Hospital, July 2003

Julia, 56 Jeremy, 12 Paul, 30 Germaine,30 Connie, 41 Tracy, 39

Friends:

Positive crossmatch

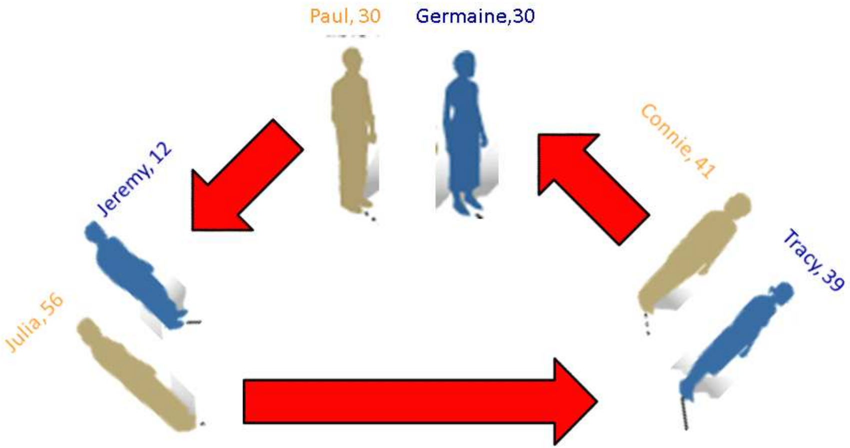
Engaged:

incompatible blood type

Sisters:

Positive crossmatch

Exchange between three pairs: Johns Hopkins Hospital, July 2003



Kidney exchange programs around the world

US Programs:

- New England Program for Kidney Exchange since 2004
- Alliance for Paired Donation
- Paired Donation Network
  - Roth, Sönmez and Ünver, 2004, 2005

Mostly involving pairwise and 3-way exchanges, but sometimes even longer (a 6-way exchange was performed in April 2008)

Kidney exchange programs around the world

US Programs:

- New England Program for Kidney Exchange since 2004
- Alliance for Paired Donation
- Paired Donation Network
  - Roth, Sönmez and Ünver, 2004, 2005

Mostly involving pairwise and 3-way exchanges, but sometimes even longer (a 6-way exchange was performed in April 2008)

Other countries:

- The Netherlands
  - Keizer et al. 2005
- South Korea
- Romania
  - Lucan et al. 2003
- UK
  - National Matching Scheme for Paired Donation (UKT)

Kidney exchange programs around the world

US Programs:

- New England Program for Kidney Exchange since 2004
- Alliance for Paired Donation
- Paired Donation Network
    - Roth, Sönmez and Ünver, 2004, 2005

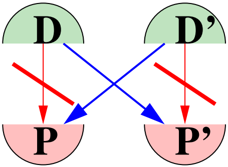
Mostly involving pairwise and 3-way exchanges, but sometimes even longer (a 6-way exchange was performed in April 2008)

Other countries:

- The Netherlands
    - Keizer et al. 2005
- South Korea
- Romania
    - Lucan et al. 2003
- UK
    - National Matching Scheme for Paired Donation (UKT)

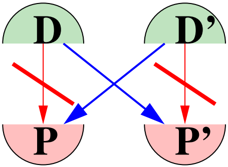
Cycles should be as short as possible

Description of the basic model



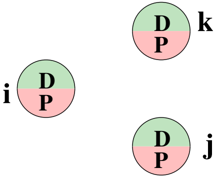
Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a *pairwise exchange* is possible between them.

Description of the basic model

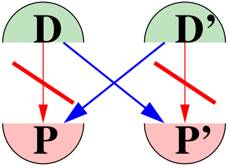


Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a *pairwise exchange* is possible between them.

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



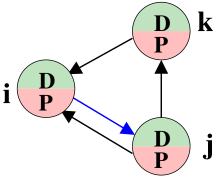
Description of the basic model



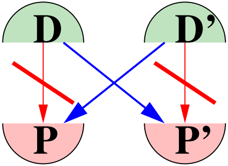
Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a *pairwise exchange* is possible between them.

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

$(i, j) \in A$ if and only if donor i is compatible with patient j.



Description of the basic model

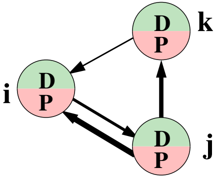


Given two incompatible patient-donor pairs (blood-type or tissue-type incompatibility). If they are compatible across, then a *pairwise exchange* is possible between them.

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

$(i, j) \in A$ if and only if donor i is compatible with patient j.

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



UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

- Location points (0 or 20)
  - 20 if d_i and p_j are in the same "area", 0 otherwise

UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

- Location points (0 or 20)
  - 20 if d_i and p_j are in the same "area", 0 otherwise
- Sensitisation points (0-50)
  - Based on calculated sensitisation ("panel reactive antibody" test) % divided by 2

UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

- Location points (0 or 20)
  - 20 if d_i and p_j are in the same "area", 0 otherwise
- Sensitisation points (0-50)
  - Based on calculated sensitisation ("panel reactive antibody" test) % divided by 2
- HLA mismatch points (0, 5, 10 or 15)
  - HLA ("Human Leukocyte Antigen") mismatch levels determine tissue-type incompatibility

UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

- Location points (0 or 20)
  - 20 if d_i and p_j are in the same "area", 0 otherwise
- Sensitisation points (0-50)
  - Based on calculated sensitisation ("panel reactive antibody" test) % divided by 2
- HLA mismatch points (0, 5, 10 or 15)
  - HLA ("Human Leukocyte Antigen") mismatch levels determine tissue-type incompatibility
- Donor-donor age difference (0 or 3)
  - 3 points if donor-donor age difference $\leq$ 20 years, 0 otherwise

UK Transplant's scoring system

A score (0-89) is given to each arc (i, j):

- Location points (0 or 20)
  - 20 if d_i and p_j are in the same "area", 0 otherwise
- Sensitisation points (0-50)
  - Based on calculated sensitisation ("panel reactive antibody" test) % divided by 2
- HLA mismatch points (0, 5, 10 or 15)
  - HLA ("Human Leukocyte Antigen") mismatch levels determine tissue-type incompatibility
- Donor-donor age difference (0 or 3)
  - 3 points if donor-donor age difference $\leq$ 20 years, 0 otherwise
- "Final discriminator" involving *actual* donor-donor age difference

The 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 optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

1. the number of vertices covered by π is maximum;

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

1. the number of vertices covered by π is maximum;
2. subject to (1), the sum of the weights is maximum (i.e., the total score is maximum).

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

1. the number of vertices covered by π is maximum;
2. subject to (1), the sum of the weights is maximum (i.e., the total score is maximum).

We study 3 cases:

- Only 2-cycles (pairwise exchanges) are possible.

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

1. the number of vertices covered by π is maximum;
2. subject to (1), the sum of the weights is maximum (i.e., the total score is maximum).

We study 3 cases:

- Only 2-cycles (pairwise exchanges) are possible.
- The cycle lengths are unrestricted.

The optimisation problems

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

A vertex $i \in V$ is *covered* by π if $\pi(i) \neq i$.

A set of exchanges is *optimal* if:

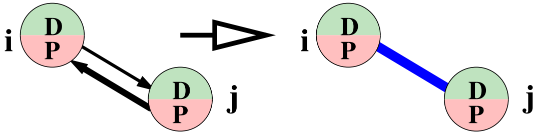
1. the number of vertices covered by π is maximum;
2. subject to (1), the sum of the weights is maximum (i.e., the total score is maximum).

We study 3 cases:

- Only 2-cycles (pairwise exchanges) are possible.
- The cycle lengths are unrestricted.
- 2- and 3-cycles (pairwise and *3-way exchanges*) are allowed.

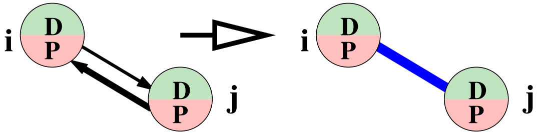
Pairwise exchanges ⟹ matching problem

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



Pairwise exchanges $\Longrightarrow$ matching problem

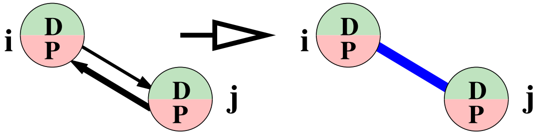
We transform the **directed graph** D to an undirected graph G.



A set of **pairwise 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.

Pairwise exchanges $\Longrightarrow$ matching problem

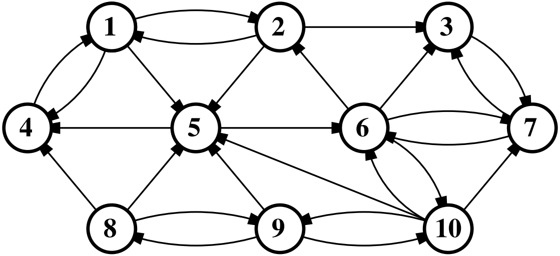
We transform the **directed graph** D to an undirected graph G.



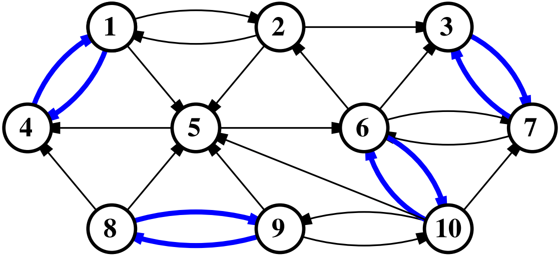
A set of **pairwise exchanges in** D corresponds to a matching in G with the same weight, since $w(\{i,j\}) = w(i,j) + w(j,i)$ for every edge $\{i,j\}$ of G.

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

Optimal pairwise exchanges: two examples

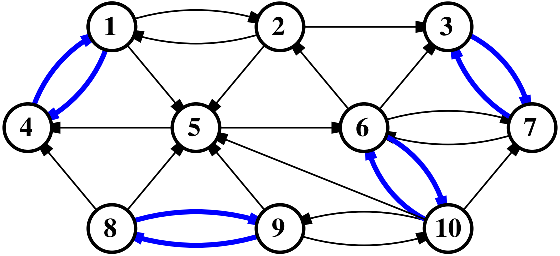


Optimal pairwise exchanges: two examples

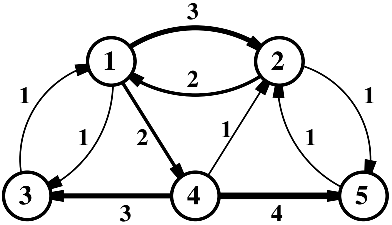


Maximum cardinality set of pairwise exchanges

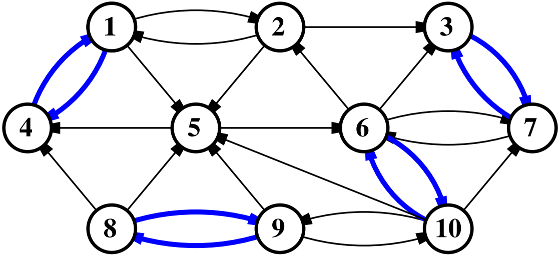
Optimal pairwise exchanges: two examples



Maximum cardinality set of pairwise exchanges

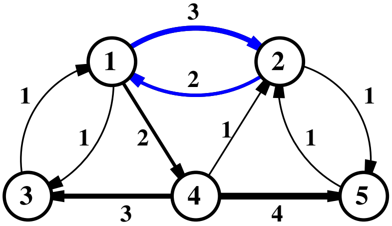


Optimal pairwise exchanges: two examples

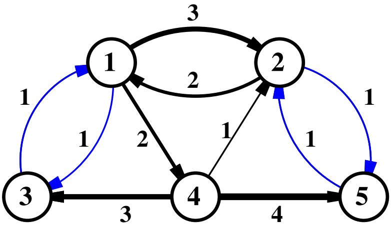


Maximum cardinality set of pairwise exchanges

Maximum weight set of pairwise exchanges

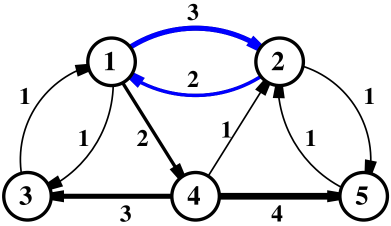


Optimal pairwise exchanges: two examples



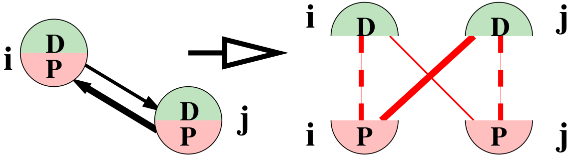
Optimal set of pairwise exchanges

Maximum weight set of pairwise exchanges



Unrestricted exchanges $\Longrightarrow$ matching problem

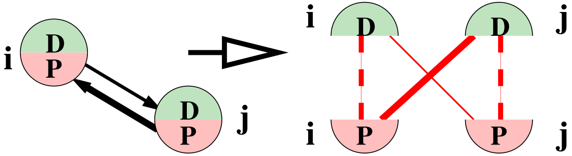
We transform the **directed graph** D to a bipartite graph G.



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

Unrestricted exchanges $\Longrightarrow$ matching problem

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

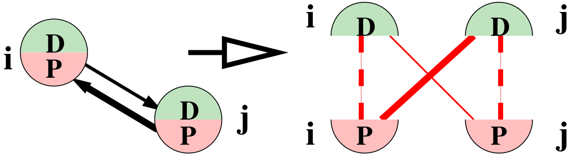


With an edge of weight 0, between each patient and his/her donor.

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

Unrestricted exchanges $\Longrightarrow$ matching problem

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

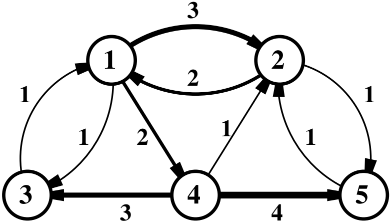


With an edge of weight 0, between each patient and his/her donor.

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

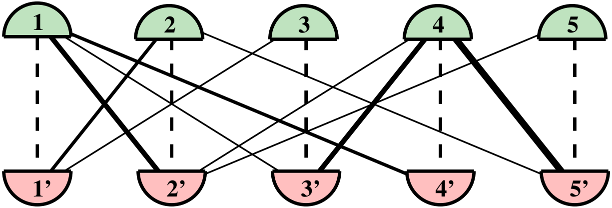
The problem of finding a maximum weight perfect matching in G can be solved in polynomial time.

The transformation in an example

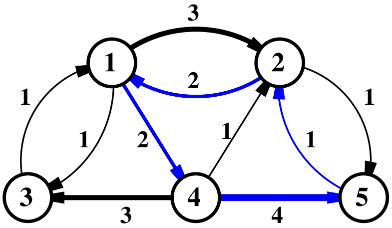


From a directed graph D,

we create a bipartite graph G,

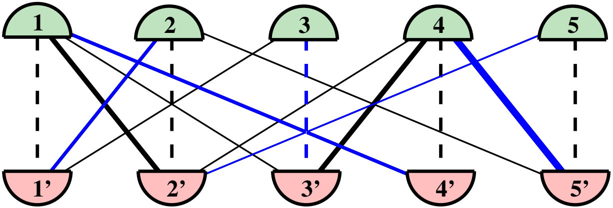


The transformation in an example

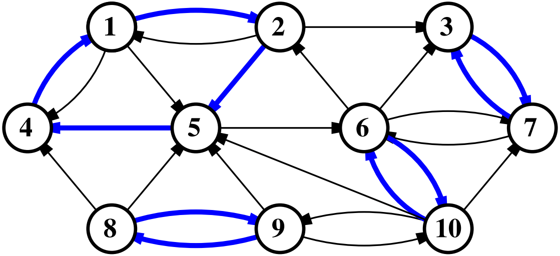


From a directed graph D, maximum weight unrestricted set of exchanges

we create a bipartite graph G, maximum weight perfect matching

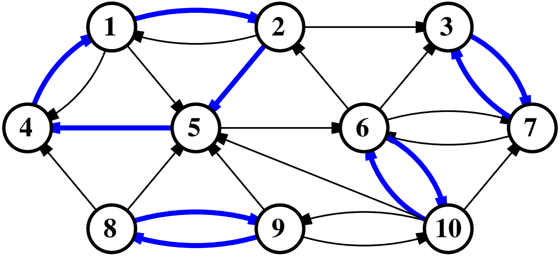


Optimal unrestricted exchanges in two examples



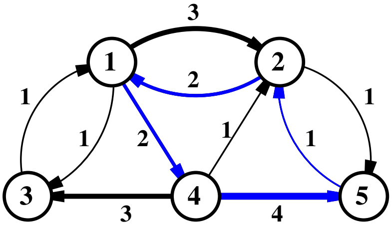
Maximum cardinality unrestricted set of exchanges

Optimal unrestricted exchanges in two examples



Maximum cardinality unrestricted set of exchanges

Maximum weight unrestricted set of exchanges



Test results for large instances:

	Pairwise exchanges			Unrestricted exchanges			
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

Pairwise and 3-way exchanges: a hard problem

The problem of finding a maximum weight set of exchanges involving only 2- and 3-cycles is NP-hard

- Abraham et al, 2007

Pairwise and 3-way exchanges: a hard problem

The problem of finding a maximum weight set of exchanges involving only 2- and 3-cycles is NP-hard

- Abraham et al, 2007

and APX-hard also

- Biró, Manlove and Rizzi, 2009

so. . .

Pairwise and 3-way exchanges: a hard problem

The problem of finding a maximum weight set of exchanges involving only 2- and 3-cycles is NP-hard

- Abraham et al, 2007

and APX-hard also

- Biró, Manlove and Rizzi, 2009

so. . .

- Either we use some polynomial-time heuristics, but we cannot guarantee to find the optimum.

Pairwise and 3-way exchanges: a hard problem

The problem of finding a maximum weight set of exchanges involving only 2- and 3-cycles is NP-hard

- Abraham et al, 2007

and APX-hard also

- Biró, Manlove and Rizzi, 2009

so. . .

- Either we use some polynomial-time heuristics, but we cannot guarantee to find the optimum.
- Or we find an exact solution by an exponential algorithm.

Pairwise and 3-way exchanges: a hard problem

The problem of finding a maximum weight set of exchanges involving only 2- and 3-cycles is NP-hard

- Abraham et al, 2007

and APX-hard also

- Biró, Manlove and Rizzi, 2009

so. . .

- Either we use some polynomial-time heuristics, but we cannot guarantee to find the optimum.
- Or we find an exact solution by an exponential algorithm.

But, in the latter case, instead of checking each possible exchange we can reduce the running time by using some ideas...

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution
- build an $n \times m$ matrix A where $n = |V|$ and $A_{i,j} = 1 \Leftrightarrow v_i$ is incident to C_j

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution
- build an $n \times m$ matrix A where $n = |V|$ and $A_{i,j} = 1 \Leftrightarrow v_i$ is incident to C_j
- let b be an $n \times 1$ vector of 1s

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$ where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution
- build an $n \times m$ matrix A where $n = |V|$ and $A_{i,j} = 1 \Leftrightarrow v_i$ is incident to C_j
- let b be an $n \times 1$ vector of 1s
- let c be a $1 \times m$ vector of values corresponding to the optimisation criterion, e.g., c_j could be weight of C_j

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution
- build an $n \times m$ matrix A where $n = |V|$ and $A_{i,j} = 1 \Leftrightarrow v_i$ is incident to C_j
- let b be an $n \times 1$ vector of 1s
- let c be a $1 \times m$ vector of values corresponding to the optimisation criterion, e.g., c_j could be weight of C_j

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

Integer linear program implemented in Matlab

We create an integer program as follows:

- list all the possible cycles (exchanges) of lengths 2 and 3 in the directed graph as $C_1, C_2, \ldots, C_m$
- use binary variables $x_1, x_2, \ldots, x_m$
  where $x_i = 1 \Leftrightarrow C_i$ belongs to an optimal solution
- build an $n \times m$ matrix A where $n = |V|$ and $A_{i,j} = 1 \Leftrightarrow v_i$ is incident to C_j
- let b be an $n \times 1$ vector of 1s
- let c be a $1 \times m$ vector of values corresponding to the optimisation criterion, e.g., c_j could be weight of C_j

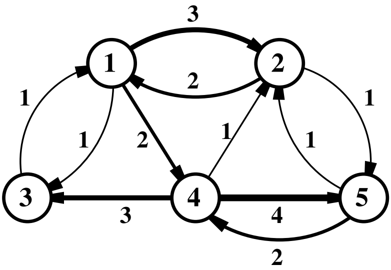
Then solve max cx s.t. $Ax \leq b$

- Roth, Sönmez and Ünver, 2007
- Abraham et al., 2007

Integer linear program implemented in Matlab

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

where

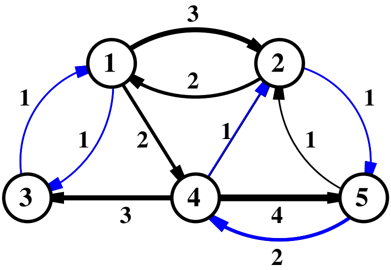


$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right], \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\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

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

Integer linear program implemented in Matlab

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



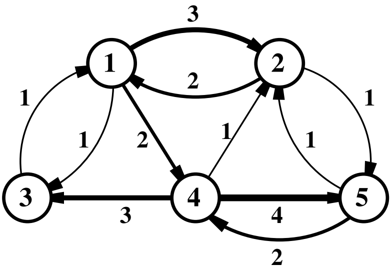
where

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

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

Integer linear program implemented in Matlab

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



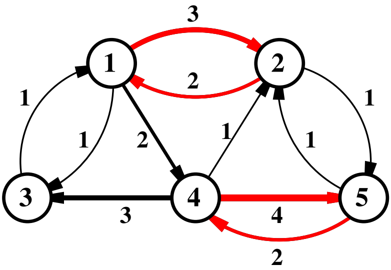
where

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

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

Integer linear program implemented in Matlab

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



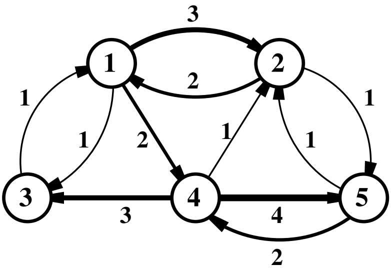
where

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

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

Integer linear program implemented in Matlab

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



where

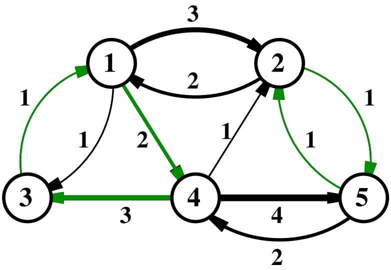
$$A = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right], \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\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

$$c_o = c_s \cdot M + c_w \text{ if optimal}$$

Integer linear program implemented in Matlab

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

where



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

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

Comparing the models: test results

	Pairwise		2- and 3-way				
nodes	size	weight	size	weight	3-c.	size inc.	weight inc.
30	11	627	16	979	4	5	352
35	11	554	16	1041	4	5	487
40	14	882	21	1414	6	7	532
45	16	1036	22	1554	6	6	518
50	17	1091	25	1709	6	8	618

Results from UKT matching runs

Matching run		Apr 08	Jul 08	Oct 08	Jan 09	Apr 09
Number of pairs		76	85	123	126	122
Pairwise exchanges	#2 cycles	2	1	6	5	5
	size	4	2	12	10	10
	weight	91	6	499	264	388
Pairwise and 3-way exchanges	#2 cycles	2	1	2	1	2
	#3 cycles	4	0	7	5	5
	size	16	2	25	17	19
	weight	620	6	1122	633	757
Unbounded exchanges	size	22	2	33	28	28
	weight	857	6	1546	1134	1275
	longest c.	20	2	27	19	23
Chosen solution (UKT)	#2 cycles	2	1	6	5	5
	#3 cycles	4	0	3	1	2
	size	16	2	21	13	16
	weight	620	6	930	422	618

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles
- maximum cardinality maximum weight set of exchanges

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles
- maximum cardinality maximum weight set of exchanges
- exchanges with altruistic donors

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles
- maximum cardinality maximum weight set of exchanges
- exchanges with altruistic donors

Future work

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles
- maximum cardinality maximum weight set of exchanges
- exchanges with altruistic donors

Future work

- Cycles of length 4 and greater

Extensions to the basic model

These models can be easily modified to find

- an optimal set of pairwise and 3-way exchanges with the fewest number of 3-cycles
- maximum cardinality maximum weight set of exchanges
- exchanges with altruistic donors

Future work

- Cycles of length 4 and greater
- Larger size of datasets