New Integer Programming Formulations for the Stable Exchange Problem

Virginia Costa[1] *Xenia Klimentova*[1] Péter Biró[2] Ana Viana[1,3]
João Pedro Pedroso[1,4]

(1) INESC TEC, Porto, Portugal

(2) Institute of Economics, Hungarian Academy of Sciences, Budapest, Hungary

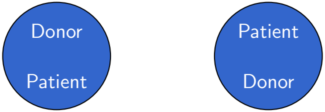
(3) ISEP - School of Engineering, Polytechnic of Porto, Porto, Portugal

(4) Faculdade de Ciências, Universidade do Porto, Porto, Portugal

COMPETE 2020 · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional · FCT Fundação para a Ciência e a Tecnologia MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

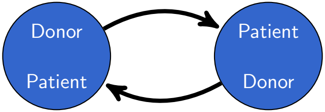
Kidney exchange program

Two incompatible donor-patient pairs.



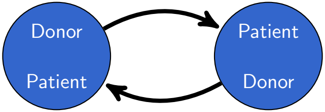
Kidney exchange program

Two incompatible donor-patient pairs.



Kidney exchange program

Two incompatible donor-patient pairs.



There are constraints to consider

- exchanges associated to a cycle must be be performed simultaneously;
- limitation on the number of simultaneous exchanges.

Kidney exchange program

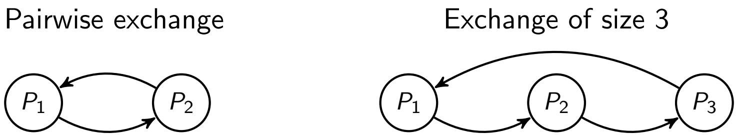


Figure 1: Kidney exchanges.

Consider exchanges of length at most K.

Objective is to define a set of exchanges in order to maximize the number of transplants performed.

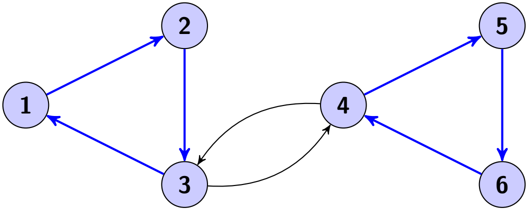
Kidney exchange problem

Let $G = (V, A)$ be a directed graph:

- Each vertex $i \in V$ represents a incompatible patient–donor pair.
- Each arc $(i, j) \in A$ represents compatibility between donor from pair j and patient in pair i.

Kidney exchange problem

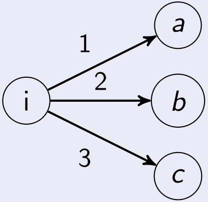
Find a set of vertex-disjoint cycles of length at most K (a feasible exchange) with maximum weight (cardinality).



Preferences

Some kidneys may fit a given patient better then another (due to medical conditions, size etc.)

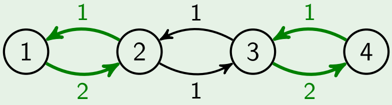
Each vertex $i \in V$ has **preferences** on list of out-neighbour vertices.



- if i *prefers* j to k we write $j <_i k$: $a <_i b <_i c$;
- the preferences are strict, i.e. $j =_i k$ implies that $j = k$;
- a vertex prefers to be matched to any out-going neighbour, rather than be unmatched.

Preferences. Intuition of stability

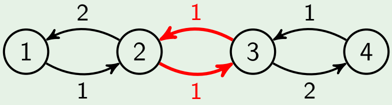
Example



- Consider exchange
  $E = \{(1,2),(3,4)\}$;

Preferences. Intuition of stability

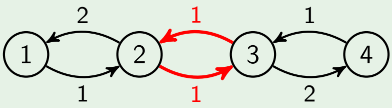
Example



- Consider exchange $E = \{(1,2),(3,4)\}$;
- In cycle $(2,3)$ both patients 2 and 3 could get better kidney.

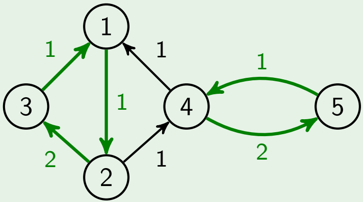
Preferences. Intuition of stability

Example



- Consider exchange $E = \{(1,2),(3,4)\}$;
- In cycle $(2,3)$ both patients 2 and 3 could get better kidney.

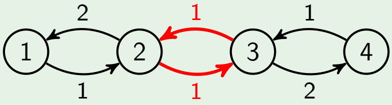
Example



- Consider exchange $E = \{(1,2,3),(4,5)\}$;

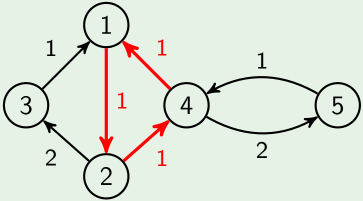
Preferences. Intuition of stability

Example



- Consider exchange $E = \{(1,2),(3,4)\}$;
- In cycle $(2,3)$ both patients 2 and 3 could get better kidney.

Example



- Consider exchange $E = \{(1,2,3),(4,5)\}$;
- In cycle $(1,2,4)$ patient 1 get the same kidney, but patients 2 and 4 get better kidneys.

Blocking and weakly blocking cycles

Blocking cycle

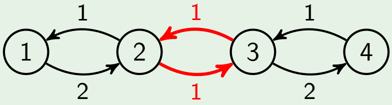
For a given feasible exchange E, a cycle $c \notin E$ is called **blocking** if every vertex i in cycle c is either unmatched or prefers the out-neighbour in c to the one in exchange E.

Blocking and weakly blocking cycles

Blocking cycle

For a given feasible exchange E, a cycle $c \notin E$ is called **blocking** if every vertex i in cycle c is either unmatched or prefers the out-neighbour in c to the one in exchange E.

Example



Cycle $(2, 3)$ is **blocking** for the exchange $E = \{(1, 2), (3, 4)\}$;

Blocking and weakly blocking cycles

Weakly blocking cycle

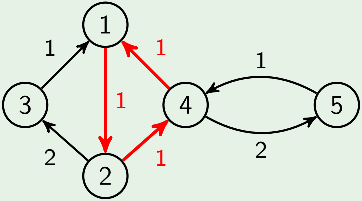
For a given feasible exchange E, a cycle $c \notin E$ is called **weakly blocking** if every vertex i in cycle c is either unmatched or is indifferent to or prefers the out-neighbour in c to the one in exchange E.

Blocking and weakly blocking cycles

Weakly blocking cycle

For a given feasible exchange E, a cycle $c \notin E$ is called **weakly blocking** if every vertex i in cycle c is either unmatched or is indifferent to or prefers the out-neighbour in c to the one in exchange E.

Example



Cycle $(1, 2, 4)$ is **weakly blocking** for the exchange $E = \{(1, 2, 3), (4, 5)\}$.

Stable and Strongly stable exchange

Stable Exchange

An exchange E is called **stable** if there is no blocking cycle $c \notin E$.

Strongly Stable Exchange

An exchange E is called **strongly stable** if there is no weakly blocking cycle $c \notin E$.

Stable and Strongly stable exchange

Stable Exchange

An exchange E is called **stable** if there is no blocking cycle $c \notin E$.

Strongly Stable Exchange

An exchange E is called **strongly stable** if there is no weakly blocking cycle $c \notin E$.

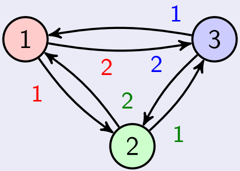
Stable exchange problem

Find a set of vertex-disjoint set of cycles of length at most K, i.e. an exchange, that is stable (strongly stable).

Stable Exchange problem, remarks

For $K = 2$ the stable exchange problem is a room-mate problem.

Stable (strongly stable) exchange may not exist. Let $K = 2$, and consider the following graph:



Edge formulation

Decision variable for each arc $(i,j) \in A$:
$y_{ij} = 1$ if arc (i,j) is chosen in exchange, 0 otherwise.

$$\sum_{j:(j,i)\in A} y_{ji} - \sum_{j:(i,j)\in A} y_{ij} = 0 \qquad \forall i \in V \tag{1}$$

$$\sum_{j:(i,j)\in A} y_{ij} \leq 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{(i,j)\in A(p)} y_{ij} \leq K - 1 \qquad \forall p \in \mathcal{P} \tag{3}$$

where $\mathcal{P}$ is the set of all paths p in G with K arcs, $A(p)$ is a set of arcs p.

Edge formulation, stability constraints

$\mathcal{C}$ is a set of cycles of length at most K; $A(c)$ is the set of arcs in c.

Stable exchange

$$\sum_{(i,j)\in A(c)} \left[y_{ij} + \sum_{r:r<_i j} y_{ir} \right] \geq 1, \quad \forall c \in \mathcal{C}. \tag{4}$$

Strongly stable exchange

$$|c| \cdot \left[\sum_{(ij)\in A(c)} \sum_{r:r<_i j} y_{ir} \right] + \sum_{(i,j)\in A(c)} y_{ij} \geq |c|, \quad \forall c \in \mathcal{C}. \tag{5}$$

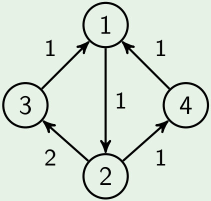
Cycle formulation, notation

- Vertex i prefers cycle c to cycle s, if for $(i, j) \in A(c)$ and $(i, k) \in s$ $j <_i k$, we denote $c \prec_i s$;
- Vertex i is indifferent between c and s if for $(i, j) \in A(c)$ and $(i, j) \in A(s)$, we denote $c =_i s$;

Cycle formulation, notation

- Vertex i prefers cycle c to cycle s, if for $(i,j) \in A(c)$ and $(i,k) \in s$ $j <_i k$, we denote $c \prec_i s$;
- Vertex i is indifferent between c and s if for $(i,j) \in A(c)$ and $(i,j) \in A(s)$, we denote $c =_i s$;

Example



- $(1,2,4) \prec_2 (1,2,3)$;
- $(1,2,4) =_1 (1,2,3)$.

Cycle formulation

Decision variable for each cycle $c \in \mathcal{C}$:
$x_c = 1$ if cycle c is in exchange, 0 otherwise.

$$\sum_{c:i \in V(c)} x_c \leq 1, \qquad \forall i \in V, \tag{6}$$

$V(c)$ is a set of vertices in cycle c.

Cycle formulation, stability constraints

$B_{i,c} = \{s \in \mathcal{C}(i), s \neq c : s \preceq_i c\}$
$S_{i,c} = \{s \in \mathcal{C}(i) : s \prec_i c\}.$

Stable exchange

$$x_c + \sum_{i \in V(c)} \sum_{s \in B_{i,c}} x_s \geq 1, \qquad \forall c \in \mathcal{C}, \tag{7}$$

Strongly stable exchange

$$x_c + \sum_{i \in V(c)} \sum_{s \in S_{i,c}} x_s \geq 1, \forall c \in \mathcal{C}, \tag{8}$$

Edge-cycle formulation

Both decision variables, y_{ij} and x_c.

$$\sum_{j:(j,i)\in A} y_{ji} - \sum_{j:(i,j)\in A} y_{ij} = 0 \qquad \forall i \in V \tag{9}$$

$$\sum_{j:(i,j)\in A} y_{ij} \leq 1 \qquad \forall i \in V \tag{10}$$

$$\sum_{(i,j)\in A(p)} y_{ij} \leq K - 1 \qquad \forall p \in \mathcal{P} \tag{11}$$

Stability constraints for edge formulation

Edge-cycle formulation

Both decision variables, y_{ij} and x_c.

$$\sum_{j:(j,i)\in A} y_{ji} - \sum_{j:(i,j)\in A} y_{ij} = 0 \qquad \forall i \in V \tag{9}$$

$$\sum_{j:(i,j)\in A} y_{ij} \leq 1 \qquad \forall i \in V \tag{10}$$

$$|c| \cdot x_c \leq \sum_{(i,j)\in A(c)} y_{i,j}, \qquad \forall c \in \mathcal{C}, \tag{11}$$

$$\sum_{(i,j)\in A(c)} y_{ij} - |c| + 1 \leq x_c, \qquad \forall c \in \mathcal{C}, \tag{12}$$

$$\sum_{j:(i,j)\in A} y_{ij} \leq \sum_{c:i\in V(c)} x_c, \qquad \forall i \in V \tag{13}$$

Stability constraints for edge formulation

Computational experiments

Objective

Maximisation of number of transplants:

Cycle formulation:

$$\max_{c} \sum_{c:c\in\mathcal{C}} |c| \cdot x_c,$$

Edge and Edge-Cycle formulation:

$$\max_{y} \sum_{(i,j)\in A} y_{ij}$$

Computational experiments

- Generated instances of kidney exchange problem, preferences randomly generated;
- 50 instances of each size;
- C++ language and GUROBI library, with default options, as integer programming solver.
- A computer with 12 cores Intel(R) Xeon(R) CPU X5675/3.07GHz, 50GB of RAM memory, Ubuntu 16.04.3 LTS operation system and g++ version 5.4.0.

Computational experiments, stable exchange problem

$\|V\|$	Av.$\|A\|$	Av.$\|\mathcal{C}\|$	Time			
				Edge f.	Cycle f.	Edge-cycle f.
20	87	32		0.01	0.00	0.00
30	205	102		0.19	0.01	0.01
40	367	228		0.93	0.03	0.03
50	572	429		5.55	0.10	0.07
60	818	688	(1)	35.87	0.26	0.13
70	1102	1027		162.81	0.60	0.23
80	1454	1521		821.29	1.77	0.41
90	1874	2238		6864.43	5.28	0.72
100	2312	2987		-	-	-

Table 1: Average results for the stable exchange problem; in parenthesis we show the number of instances (out of 50) for which stable exchange do not exist.

Computational experiments, strongly stable exchange problem

$\|V\|$	Av.$\|A\|$	Av.$\|\mathcal{C}\|$		Time		
				Edge f.	Cycle f.	Edge-cycle f.
20	87	32	(1)	0.00	0.00	0.00
30	205	102	(1)	0.04	0.00	0.01
40	367	228	(2)	0.15	0.01	0.02
50	572	429	(5)	0.55	0.04	0.04
60	818	688	(5)	0.96	0.10	0.07
70	1102	1027	(4)	3.98	0.26	0.14
80	1454	1521	(4)	20.87	0.78	0.27
90	1874	2238	(3)	39.40	2.04	0.46
100	2312	2987	(7)	136.56	4.65	0.65

Table 2: Average results for the strongly stable exchange problem; in parenthesis we show the number of instances (out of 50) for which strongly stable exchange do not exist.

Conclusion and future work

Conclusion:

- We propose three formulations for the Stable Exchange Problem and carried out computational experiments on the test instance for the kidney exchange problem.
- Edge-cycle formulation clearly outperforms the other formulations for all instances both for stable and strongly stable exchange variants of the problem.

Future work:

- Extend the problem to the case where preferences on vertices are not strict;
- Perform more computational experiments to evaluate how good the formulations scale, and, if necessary, to develop effective methods for solving the problem of larger size.

Conclusion and future work

Conclusion:

- We propose three formulations for the Stable Exchange Problem and carried out computational experiments on the test instance for the kidney exchange problem.
- Edge-cycle formulation clearly outperforms the other formulations for all instances both for stable and strongly stable exchange variants of the problem.

Future work:

- Extend the problem to the case where preferences on vertices are not strict;
- Perform more computational experiments to evaluate how good the formulations scale, and, if necessary, to develop effective methods for solving the problem of larger size.

Acknowledgements

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project "mKEP - Models and optimisation algorithms for multicountry kidney exchange programs" (POCI-01-0145-FEDER-016677), by FCT project SFRH/BPD/101134/2014 and by COST Action CA15210 ENCKEP, supported by COST (European Cooperation in Science and Technology) – http://www.cost.eu/. Biró is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3/2018) and Cooperation of Excellences Grant (KEP-6/2018), and by the Hungarian Scientific Research Fund – OTKA (no. K129086).

Thanks for you attention!