European Journal of Operational Research 231 (2013) 57–68

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

New insights on integer-programming models for the kidney exchange problem

Miguel Constantino [b,*,1,2], Xenia Klimentova [a,1], Ana Viana [a,c,1], Abdur Rais [d,1]

[a] INESC TEC (formerly INESC Porto), Campus da FEUP, Rua Dr. Roberto Frias, 378, 4200-465 Porto, Portugal
[b] Centro de Investigação Operacional, Departamento de Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa, Portugal
[c] ISEP – School of Engineering, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal
[d] Centro Algoritmi, Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal

ARTICLE INFO

Article history:
Received 3 August 2012
Accepted 15 May 2013
Available online 30 May 2013

Keywords:
Kidney exchange program
Integer programming
Combinatorial optimization
Healthcare

ABSTRACT

In recent years several countries have set up policies that allow exchange of kidneys between two or more incompatible patient–donor pairs. These policies lead to what is commonly known as kidney exchange programs.

The underlying optimization problems can be formulated as integer programming models. Previously proposed models for kidney exchange programs have exponential numbers of constraints or variables, which makes them fairly difficult to solve when the problem size is large. In this work we propose two compact formulations for the problem, explain how these formulations can be adapted to address some problem variants, and provide results on the dominance of some models over others. Finally we present a systematic comparison between our models and two previously proposed ones via thorough computational analysis. Results show that compact formulations have advantages over non-compact ones when the problem size is large.

1. Introduction

Kidney transplants are essential for the survival of many patients suffering from kidney failure, but finding suitable kidneys can be difficult because of their scarcity as well as blood or tissue incompatibility between donors and patients. For a long time, deceased donors were typically the most acceptable source of kidneys for transplantation. However, they only met a tiny fraction of the demand and alternative transplantation policies considering living donors progressively stepped forward. Within these policies, if a patient had someone willing to donate a kidney and the patient–donor pair was compatible, then the transplant could be done. However, if a patient and the prospective donor were not physiologically compatible, then transplantation could not be performed.

In recent years kidney exchange programs brought new hope for many kidney patients. These programs involve patient–donor pairs in which donors are incompatible with their recipients. The key aspect is to organize exchanges between a number of such pairs so that patient P in one pair receives a kidney from donor D in the other pair. Fig. 1 (left) illustrates the simplest case with only two pairs (P_1,D_1) and (P_2,D_2), where patient and donor in each pair are incompatible (dotted lines represent incompatibilities). However, P_1 is compatible with D_2 and P_2 is compatible with D_1. Previously, when exchanges between pairs were not allowed, no transplants could be performed in this situation. Within the evolving frameworks of new programs, exchanges between such pairs are allowed and the two transplants can be performed (arrows represent the exchange in the figure).

Kidney exchange programs have already been introduced in many countries, including South Korea [20], Switzerland [37], Turkey [15], Romania [21], The Netherlands [8,9,19], UK [7,17,23] and the US [35,36,2,39]. Very recently, similar programs have also been set up in other countries: in 2010, Canada, Portugal, Australia, and New Zealand kicked-off their own programs while Spain initiated its program in 2011.

The objective for optimization in a kidney exchange program is generally to maximize the collective benefit for a given pool of

* Corresponding author. Tel.: +351 214500402; fax: +351 217500022.
E-mail addresses: mfconstantino@fc.ul.pt (M. Constantino), xenia.klimentova@inescporto.pt (X. Klimentova), aviana@inescporto.pt (A. Viana), abdur.rais@dps.uminho.pt (A. Rais).

[1] This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programm for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "KEP – New models for enhancing the kidney transplantation process./FCT ref: PTDC/EGE-GES/110940/2009".
[2] FCT – Fundação para a Ciência e a Tecnologia, under the project PEst-OE/MAT/UI0152.

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.05.025

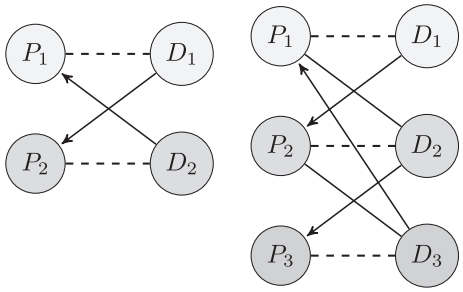

Fig. 1. A 2-way exchange (left) and a 3-way exchange (right).

incompatible pairs, usually measured by the number of possible kidney exchanges [36,8] — in the entirety of the paper we will refer to this optimization problem as the *Kidney Exchange Problem* (KEP). Although such an optimal solution is typically desirable, there are other factors which may also be considered in some situations; e.g. maximize the weighted sum of kidney exchanges [23] and/or the quality-adjusted life expectancy of transplant candidates [42].

One of the crucial questions for the KEP is the definition of a bound on the number of pairs that can be involved in an exchange. When a kidney exchange involves only two donor–recipient pairs as illustrated in Fig. 1 (left) it is commonly known as a 2-*way* or 2-*cycle* exchange. Basically this is an alternating directed cycle of two donors and two recipients in which the donor from one incompatible pair gives one kidney to the recipient in the other pair and vice versa. One can note that the size of the exchange cycle can be increased. For example, the 3-*way* exchange presented in Fig. 1 (right) allows three patients to get transplants instead of two; solid lines here represent compatibilities and arrows represent the actual exchanges that derive maximum collective benefit.

Generally *k-cycle* exchanges with $k \geqslant 3$ can be better for optimization as they have the potential for increasing the options for involving more incompatible pairs in an "exchange cycle". If there is no bound on the number of pairs in an exchange cycle, i.e., k is not fixed, the problem turns into an assignment problem and can be solved in polynomial time [2]. Ideally all operations involved in a cycle should be performed simultaneously so that donors remain committed when the incompatible partners receive other donors' kidneys. Therefore for a solution to be practical and manageable, the length of the cycles should be restricted for at least two main reasons. First, the number of personnel and facilities needed for simultaneous operations of donors and patients raise several logistic issues that can make it prohibitively inconvenient to handle too many operations simultaneously [2]. Second, because last-minute tests on donors and patients can bring out new incompatibility issues that can cause a kidney donation and related exchanges in the cycle to be canceled, it is preferable for the cycles to be shorter.

For a given pool of donor–recipient pairs, a 2-cycle exchange can be seen as a task of pairwise compatibility matching, and Edmond's maximum cardinality matching algorithm [11] can provide an optimal solution in polynomial time. The problem with k-cycle exchange certainly is a generalized model and much more interesting for practical applications. However, the associated problem is known to be *NP*-complete for $k \geqslant 3$ and difficult to solve efficiently when a problem instance is large [2].

Current work on solving the KEP focuses mostly on Integer Programming (IP) formulations. Two IP models, referred to in this paper as the "edge formulation" and the "cycle formulation", were proposed in [34]. Despite the very good results reported for the cycle formulation in [2], the question of finding a compact formulation that has a number of variables and constraints bounded by a polynomial in the size of the problem (i.e., on the total number of pairs in a donor–recipient pool), has, up until now, remained open: the cycle formulation presents an exponential number of variables, while the edge formulation has an exponential number of constraints.

This paper focuses on mathematical modeling aspects of the KEP: we propose two new compact formulations for the problem. Moreover we investigate the relationships between different formulations and provide some proofs of dominance of one formulation over the other in the sense of values of upper bounds for optimal solutions obtained with the linear relaxations (LP relaxations) of each formulation. Finally, a systematic comparison of these formulations with the two previously reported ones is presented by thorough computational analysis.

The paper is organized as follows. Following this introduction we review in Section 2 relevant literature with respect to variants of the KEP and solution methods. In Section 3 the problem statement and the known IP models are presented. The new compact formulations for the KEP are introduced in Section 4. In Section 5 the adaptation of formulations for variants of the KEP is discussed. The interrelations of upper bounds of linear relaxations for the presented IP models are investigated in Section 6. Section 7 reports the computational analysis and conclusions on the effectiveness of each formulation. Finally Section 8 provides conclusions and directions for future work.

2. Literature review

The concept of kidney exchange program for incompatible patient–donor pairs was first promoted in 1986 in [27] as an alternative to deceased donor programs. Since then, several models for the KEP have been proposed that differ mostly on type of exchanges allowed, matching requirements and optimization objectives. For ethical issues concerning the programs, readers may see [30,31]; an overview of contemporary ideas and challenges can be found in [40,12]. In this section we survey KEP variants as well as optimization solution methods used to attack the problem.

2.1. Problem variants

The basic variant of the KEP is a 2-*exchange* mechanism involving two patients in two distinct pairs such that each patient is incompatible with the associated donor [22,18,36] (see Fig. 1 (left)). The notion can be generalized to a *k-exchange* ($k \geqslant 3$) in which up to k pairs can be involved in the exchange cycle [8,2,7].

Variants of the *k-exchange* problem can include *altruistic donors*; i.e., donors that are not associated to any patient, but willing to donate a kidney to someone in need. *Non-directed* (ND) exchanges occur when an altruistic donor gives a kidney to a patient in a kidney exchange program and the recipient's donor is "dominoed" to the next compatible patient on the deceased donor waiting list, or is used to add another incompatible pair to the chain [25,13,32]. The maximum size of a chain is determined by national or regional programs.

Contrary to the above mentioned problems where simultaneity of exchanges is considered, *Non-simultaneous Extended Altruistic Donor* (NEAD) chains allow non-simultaneity of exchanges [10,28,4,12]. Unlike the conventional form of non-direct donation, where the size of the chain is limited, the cascade in NEAD may theoretically never end. The first donor who is incompatible, and whose related patient receives a kidney from the altruistic donor, gives his kidney to someone else with whom he is compatible. The recipient's incompatible donor can then do the same, and so on. By not assigning a kidney to a patient in the deceased donor list the cascading donor chain may continue indefinitely, unless a donor whose related recipient has already been transplanted drops out of the program.

The inclusion of *compatible pairs* in kidney exchange programs defines one more variant of the KEP [14]. In this variant the compatible pair can be involved in an exchange only in the case that the patient of this pair benefits from being in the pool (e.g. receive a "better" kidney than he/she would have received from his/her compatible donor).

Another variant is the *multiple donors* case, when one or more patients have multiple donors associated. For such patients only one of their multiple donors can be selected to create an exchange cycle [2]. The donor selected is the one that would lead to maximum collective benefit.

All variants of KEP outlined above consider the problem as a *static* or offline problem. But the problem can also be *dynamic* (online) when isolated patients, patient–donor pairs, and altruistic donors appear and expire over time [38,5,42,41,3].

2.2. Solution methods

The complexity of the k-exchange problem was investigated in [1,2,7]. In [1,2] it is shown that for a given graph G and $k \geqslant 3$ the problem of deciding if G admits a perfect cycle cover containing cycles of length at most k is *NP*-complete. The proof uses a reduction from the 3D–matching problem which is *NP*-complete. In [7] the authors proved the *APX*-completeness of the problem of finding a maximum size exchange involving only 2- and 3-cycles. In other words they claim that the 3-exchange KEP do not admit any polynomial-time approximation scheme, unless $P = NP$.

Edmonds' algorithm [11] for maximum cardinality matching was followed by [36] and [33] for the 2-exchange problem. When $k \geqslant 3$ a natural and perspective way for attacking the KEP is through IP models. Two different integer programs were used in [2] and [34]: the *edge* and the *cycle* formulations. In [2] a sketch of a proof is presented showing that the cycle formulation provides a better upper bound of the optimal solution with LP relaxation than the edge formulation. In the same work a cutting plane method was implemented for the edge formulation while a column generation method with branch-and-bound was designed for the cycle formulation. The cycle formulation is used in [23] to solve the KEP in UK. The program in UK allows direct and altruistic exchanges and considers a set of criteria that should be pursued in a hierarchical way.

Results related to the potential for developing NEAD strategy were presented in [28]. In [13] the authors created a pool of incompatible pairs based on the statistical data for blood-type, positive cross-matching probabilities, and others, and looked for 2-exchanges. They implemented Monte Carlo simulations and calculated the maximum number of transplants under different scenarios when including ND donors and NEAD chains and concluded that NEAD chains are not clearly superior in terms of the number of transplants achieved. The authors of [4] presented some simulations similar to [13], but allowing long chain segments. With this additional flexibility they concluded that NEAD chains lead to more transplants. Some theoretical and computational analysis of the efficacy of chains initiated by altruistic donation are provided by [10].

The possibility of opening the pool of kidney exchange programs to compatible pairs has been addressed by, e.g. [29] and [14]. A mixed pool of compatible and incompatible pairs was simulated in [14], the results show the benefits of such policy in terms of the increase in probability of matching incompatible patients that might otherwise not get a compatible donor. But the inclusion of compatible pairs in a pool of incompatible pairs is a controversial topic. Some of the ethical aspects associated with it are pointed out in [29,32].

Some results on the dynamic variant of KEP are reported in [38,5,42]. In [38] the author study how exchanges should be conducted through a centralized mechanism in a dynamically evolving agent pool with time and compatibility based preferences. They derive dynamically efficient 2-way and multi-way exchange mechanisms that maximize total discounted exchange surplus. In [5], KEP is considered an online problem in which patient–donor pairs and altruistic donors appear and expire over time. The authors studied trajectory-based online stochastic optimization algorithms for this problem. They identified tradeoffs between different parameters and developed an experimental methodology for setting them. The work in [42] considers KEP as a dynamic resource allocation problem with three objectives: maximize the quality-adjusted life expectancy of transplant candidates (clinical efficiency), and minimize two measures of inequity – the first measure is a linear function of the likelihood of transplantation of the various types of patients and the second is a quadratic function that quantifies the differences in mean waiting times across patient types. The dynamic status of patients is modeled by a set of linear differential equations.

3. Problem definition and formulation

Graph theory can provide a natural framework for representing the KEP models. Let $G(V,A)$ be a directed graph with the set of vertices $V = \{1, \ldots, |V|\}$ consisting of all incompatible patient–donor pairs and the set of arcs A designating compatibilities between the vertices. Two vertices $i, j \in V$ are connected by arc (i,j) if the patient in pair j is compatible with the donor in pair i. To each arc is associated a weight w_{ij}, $(i,j) \in A$. If the objective is to maximize the total number of transplants $w_{ij} = 1$, $\forall (i,j) \in A$.

Fig. 2 illustrates an example where four incompatible pairs are considered, the compatibility between pairs being represented by weighted arcs. An exchange is defined by a set of disjoint cycles in the whole graph and it is feasible if every cycle length does not exceed a given limit k. In Fig. 2 for $k = 3$ the possible cycles are 1–2–3–1 and 3–4–3. However these cycles are not vertex-disjoint as they have vertex 3 in common.

If only 2-way exchanges can be considered, the maximum number of transplants in this pool will be two (between pairs 3 and 4). However, if up to three pairs can be involved in an exchange cycle the optimal matching for this example will be three (donor 1 gives a kidney to patient 2, donor 2 to patient 3, and donor 3 to patient 1).

Definition 1. The *Kidney Exchange Problem* can be defined as follows:

Find a maximum weight set of vertex-disjoint cycles having length at most k.

One of the most effective ways to deal with it is using Integer Programming. Below we introduce the two formulations previously proposed in the literature for this problem: the edge and cycle formulations.

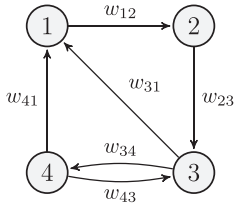

Fig. 2. KEP graph.

3.1. Edge formulation

In the edge formulation, a variable x_{ij} is associated with each arc $(i,j) \in A$ in the graph $G(V,A)$, defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if patient of pair } j \text{ gets a kidney from donor of pair } i, \\ 0 & \text{otherwise.} \end{cases}$$

The edge formulation is given by:

$$\text{Maximize} \sum_{(i,j)\in A} w_{ij}x_{ij} \tag{1a}$$

$$\text{Subject to} \sum_{j:(j,i)\in A} x_{ji} = \sum_{j:(i,j)\in A} x_{ij} \quad \forall i \in V \tag{1b}$$

$$\sum_{j:(i,j)\in A} x_{ij} \leqslant 1 \quad \forall i \in V \tag{1c}$$

$$\sum_{1\leqslant p\leqslant k} x_{i_p i_{p+1}} \leqslant k-1 \quad \forall \text{paths } (i_1, i_2, \ldots, i_k, i_{k+1}) \tag{1d}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{1e}$$

The objective function (1a) maximizes the weighted sum of the exchange – in the case of unitary weights, it corresponds to maximizing the total number of transplants. Constraints (1b) assure that patient i receives a kidney iff donor i donates a kidney. Constraints (1c) guarantee that a donor can only donate one kidney and constraints (1d) enforce the cycle-length: to exclude cycles larger than k, we need to make sure that every path of length k arcs does not have more than $k-1$ arcs in a feasible exchange. This constraint requires all paths of length k to be considered explicitly in the model. In general the number of such paths can grow exponentially with k.

3.2. Cycle formulation

An alternative IP model for the edge formulation is the so called cycle formulation. Let $\mathcal{C}(k)$ be the set of all cycles in G with length at most k. We assume that a cycle is an ordered set of arcs. Define a variable z_c for each cycle $c \in \mathcal{C}(k)$:

$$z_c = \begin{cases} 1 & \text{if cycle } c \text{ is selected for the exchange,} \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $V(c) \subseteq V$ the set of vertices which belong to cycle c. The model can be written as follows (where $w_c = \sum_{(i,j)\in c} w_{ij}$):

$$\text{Maximize} \sum_{c\in\mathcal{C}(k)} w_c z_c \tag{2a}$$

$$\text{Subject to} \sum_{c:i\in V(c)} z_c \leqslant 1 \quad \forall i \in V \tag{2b}$$

$$z_c \in \{0,1\} \quad \forall c \in \mathcal{C}(k). \tag{2c}$$

In the case of unitary weights, w_c equals the number of edges in c, i.e., the number of transplants associated with cycle c. The objective function (2a) maximizes the weighted number of transplants. Constraints (2b) ensure that every vertex is in at most one of the selected cycles (i.e., each donor may donate, and each patient may receive only one kidney). Compared to the edge formulation, the difficulty with this formulation is induced by the exponential number of variables. Indeed, the number of cycles can grow exponentially with k.

4. New compact formulations

As previously noted the number of constraints or variables in the formulations that had been proposed up to now for the KEP can grow exponentially with k. It is known that such formulations can sometimes provide better bounds with linear relaxation than "compact" ones [26] but computationally the size of the problem may become a bottleneck as solution procedures can take long time for solving large problem instances.

We present two new formulations for the problem: the *edge-assignment formulation* and the *extended edge formulation*. Each of these formulations is compact, i.e., both the number of variables and constraints are bounded by a polynomial in the size of the problem, given by the number of pairs. In the edge-assignment formulation the path constraints represented by (1d) are reformulated using additional assignment variables. In the extended edge formulation an extra index will be introduced in the variables x_{ij} for allowing the cycle cardinality constraints to be created in a simple way. In the next couple of sections we present the IP models. A discussion on their downsizing into "reduced" formulations is also provided.

4.1. Edge-assignment formulation

Let L be an upper bound on the number of cycles in any solution. For instance, a simple upper bound is $L = |V|$ as the number of cycles cannot exceed the number of vertices. Let each cycle in the solution be indexed by l, $1 \leqslant l \leqslant L$ and denote by $V(l) \subseteq V$ the set of vertices in cycle l. Define the following assignment variables:

$$y_i^l = \begin{cases} 1 & \text{if node } i \text{ belongs to cycle } l, \\ 0 & \text{otherwise.} \end{cases}$$

With these additional variables, we can write the cycle cardinality constraints as:

$$\sum_i y_i^l \leqslant k \quad \forall l \in 1, \ldots, L. \tag{3a}$$

It is necessary now to ensure that each node i is properly assigned to a cycle l, and this is done using the following constraints:

$$\sum_l y_i^l = \sum_{j:(i,j)\in A} x_{ij} \quad \forall i \in V \tag{4a}$$

$$y_i^l + x_{ij} \leqslant 1 + y_j^l \quad \forall (i,j) \in A, \quad \forall l \in 1, \ldots, L \tag{4b}$$

$$y_i^l \in \{0,1\} \quad \forall i \in V, \quad \forall l \in 1, \ldots, L. \tag{4c}$$

Constraints (4a) ensure that node i is in a cycle $\left(\sum_{j:(i,j)\in A} x_{ij} = 1\right)$ if and only if there is an assignment of i to some $l\left(\sum_l y_i^l = 1\right)$. Constraints (4b) state that if node i is in cycle $l\left(y_i^l = 1\right)$ and donor i gives a kidney to recipient j ($x_{ij} = 1$) then node j must also be in cycle $l\left(y_j^l = 1\right)$.

The edge-assignment formulation is composed of constraints from the edge formulation – (1a), (1b), (1c) and (1e) – together with the constraints (3a), (4a), (4b) and (4c).

For the formulation above, for a given index l, the set of nodes $i \in V(l)$ giving $y_i^l = 1$ could in fact belong to more than one vertex-disjoint simple cycle (see e.g. Fig. 3), but the number of arcs in all of these cycles are guaranteed to have no more than k arcs.

Furthermore, the edge-assignment formulation allows for multiple equivalent solutions, i.e. has symmetry. If there is a solution having p cycles represented by indices $l_1, \ldots, l_p$, other orderings of p indices may correspond to the same solution as exemplified in Fig. 4. A solution consisting of cycles between pairs 2, 3 and 4 and pairs 1 and 5 can be obtained with different indexing l for cycles 1 and 2.

The symmetry can be a problem because, as stated in [24], "Even for relatively modestly sized problems, ILPs with large symmetry groups are difficult to solve using traditional branch-and-bound or branch-and-cut algorithms". One way to avoid the symmetry of the edge-assignment formulation is by representing each cycle by its node with the lowest index. In other words, for $L = |V|$, a cycle having nodes $i_1, \ldots, i_r$ is represented by index $l = \min\{i_1,$

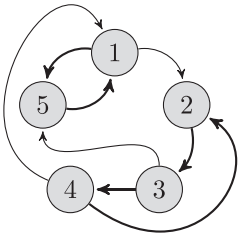

Fig. 3. Index l can represent more than one cycle. For e.g. $l = 1$, the solution with $y_1^1 = y_5^1 = y_2^1 = y_3^1 = y_4^1 = 1$, $x_{15} = x_{51} = x_{23} = x_{34} = x_{42} = 1$ and all other variables equal to 0 satisfies all the constraints of the edge-assignment formulation for $k = 5$.

$\ldots, i_r\}$. In this case only variables y_i^l with $i \geqslant l$ are necessary and this can be enforced by restricting the variables y_i^l to indices $1 \leqslant l \leqslant i \leqslant L$ and by adding constraints:

$$y_i^l \leqslant y_l^l \quad \forall i \in V, \quad l = 1, \ldots, L, \quad i > l \tag{5a}$$

Note that here and below we eliminate variables by fixing them to 0 (i.e. $y_i^l = 0, \forall i \in V, l = 1, \ldots, L, i \leqslant l$).

4.1.1. Reduced edge-assignment formulation

To tighten the model further, in some situations the variable y_i^l can be eliminated if l and i cannot be in the same cycle. Let $\widetilde{V}^l = \{i \in V : i \geqslant l\}$ and d_{ij}^l denote the shortest path distance in terms of number of arcs in graph G from i to j for $i, j \in \widetilde{V}^l$ such that the path passes only through vertices of set $\widetilde{V}^l$. Let $d_{ij}^l = +\infty$ if there is no such path from i to j. For a given i if $d_{li}^l + d_{il}^l > k$, then there is no cycle with length k or less containing both nodes l and i. In this case the variables y_i^l need not be considered in the model. More precisely, for each vertex $l \in V$, let us build the set of vertices $V^l = \left\{i \in V | i \geqslant l \text{ and } d_{li}^l + d_{il}^l \leqslant k\right\}$. It can happen that for some $l \in \{1, \ldots, L\}$ $V^l = \emptyset$. Denote by $\mathcal{L} \subseteq \{1, \ldots, L\}$ the set of indices l such that $V^l \neq \emptyset$. The reduced edge-assignment formulation can now be represented by Eqs. 6a, (6b)–(6i).

$$\text{maximize} \sum_{(i,j)\in A} w_{ij} x_{ij} \tag{6a}$$

$$\text{subject to} \sum_{j:(j,i)\in A} x_{ji} = \sum_{j:(i,j)\in A} x_{ij} \quad \forall i \in V \tag{6b}$$

$$\sum_{j:(i,j)\in A} x_{ij} \leqslant 1 \quad \forall i \in V \tag{6c}$$

$$\sum_{i\in V^l} y_i^l \leqslant k \quad \forall l \in \mathcal{L} \tag{6d}$$

$$\sum_{l\in\mathcal{L}: i\in V^l} y_i^l = \sum_{j:(i,j)\in A} x_{ij} \quad \forall i \in V \tag{6e}$$

$$y_i^l + x_{ij} \leqslant 1 + y_j^l \quad \forall (i,j) \in A, \quad i \in V^l, \quad \forall l \in \mathcal{L} \tag{6f}$$

$$y_i^l \leqslant y_l^l \quad \forall i \in V^l, l \in \mathcal{L} \tag{6g}$$

$$y_i^l \in \{0,1\} \quad \forall i \in V^l, \forall l \in \mathcal{L} \tag{6h}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{6i}$$

4.2. Extended edge formulation

Consider L copies of the graph G, and let l be the index of a copy. Recall that L is an upper bound on the number of cycles in a solution. In each copy l at most k arcs can create a cycle and each node $i \in V$ can belong to at most one such cycle. This can be captured by "cycle cardinality constraints" in a new model using the variables x_{ij}^l. Let

$$x_{ij}^l = \begin{cases} 1 & \text{if arc } (i,j) \text{ is selected to be in copy } l \text{ of the graph,} \\ 0 & \text{otherwise.} \end{cases}$$

The extended edge formulation is given by:

$$\text{maximize} \sum_{l\in\{1,\ldots,L\}} \sum_{(i,j)\in A} w_{ij} x_{ij}^l \tag{7a}$$

$$\text{subject to} \sum_{j:(j,i)\in A} x_{ji}^l = \sum_{j:(i,j)\in A} x_{ij}^l \quad \forall i \in V, \forall l \in \{1, \ldots L\} \tag{7b}$$

$$\sum_{l} \sum_{j:(i,j)\in A} x_{ij}^l \leqslant 1 \quad \forall i \in V \tag{7c}$$

$$\sum_{(i,j)\in A} x_{ij}^l \leqslant k \quad \forall l \in \{1, \ldots L\} \tag{7d}$$

$$x_{ij}^l \in \{0,1\}. \quad \forall (i,j) \in A, \forall l \in \{1, \ldots L\} \tag{7e}$$

The objective (7a) is to maximize the total weight of the arcs taken from all copies of the graph. Constraints (7b) state that in each copy l of the graph, the number of kidneys received by patient i is equal to the number of kidneys given by donor i. To make sure that a donor/patient intervene only once, constraints (7c) ensure that a node can only be selected in at most one copy of the graph. Constraints (7d) state that at most k edges can be used from each copy of the graph. This essentially prevents the cycles from becoming larger than k as each copy of the graph allows only cycles of length k or less. There can be more than one cycle in a copy of the graph but the total number of edges in all the cycles is at most k.

As with the edge-assignment formulation the extended edge formulation also has symmetry. To avoid multiplicity of solutions a similar approach can be used here as well. If a copy l of the graph provides a cycle for some solution, then node l must be in this cycle and all other nodes must have indices larger than l. Hence, all variables x_{ij}^l such that $i < l$ or $j < l$ may be discarded from the model, and constraints similar to (5a) can be added to set l as the lowest index of all nodes in the cycle:

$$\sum_j x_{ij}^l \leqslant \sum_j x_{lj}^l \quad \forall i > l. \tag{8}$$

4.2.1. Reduced extended edge formulation

Along the same lines as the elimination procedures for the variables y_i^l proposed for the edge-assignment formulation (see Section 4.1), one may also be able to eliminate variables x_{ij}^l in the extended edge formulation. If there is no cycle of size at most k

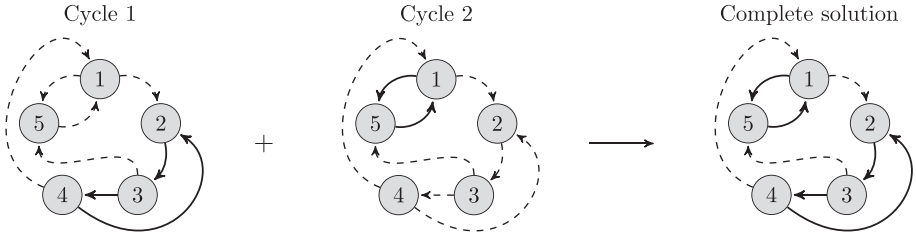

Fig. 4. Symmetry in edge-assignment formulation. The complete solution for $k = 3$ with $x_{15} = x_{51} = x_{23} = x_{34} = x_{42} = 1$ and all other $x_{ij} = 0$ can be obtained with different indexing l. For e.g. $l = 1$ for Cycle 1 and $l = 2$ for Cycle 2 with $y_1^1 = y_5^1 = y_2^2 = y_3^2 = y_4^2 = 1$ and vice versa $l = 2$ for Cycle 1 and $l = 1$ for Cycle 2 with $y_1^2 = y_5^2 = y_2^1 = y_3^1 = y_4^1 = 1$, all other $y_i^l = 0$.

containing both node l and an arc (i,j) with $i \geqslant l, j \geqslant l$, then variable x_{ij}^l can be set to zero or simply eliminated from the model.

Summarizing, the application of the elimination procedures leads to the construction of a subgraph $G^l = (V^l, A^l)$ for each index $l \in \mathcal{L}$, with $V^l, \mathcal{L}$ and d_{ij}^l as defined in Section 4.1 and $A^l = \left\{(i,j) \in A | i,j \in V^l \text{ and } d_{li}^l + 1 + d_{jl}^l \leqslant k\right\}$. With this notation the reduced extended edge formulation is given as follows.

$$\text{maximize} \sum_{l \in \mathcal{L}} \sum_{(i,j) \in A^l} w_{ij} x_{ij}^l, \tag{9a}$$

$$\text{subject to} \sum_{j:(j,i) \in A^l} x_{ji}^l = \sum_{j:(i,j) \in A^l} x_{ij}^l \quad \forall i \in V^l, \forall l \in \mathcal{L}, \tag{9b}$$

$$\sum_{l \in \mathcal{L}} \sum_{j:(i,j) \in A^l} x_{ij}^l \leqslant 1 \quad \forall i \in \bigcup_{l \in \mathcal{L}} V^l, \tag{9c}$$

$$\sum_{(i,j) \in A^l} x_{ij}^l \leqslant k \quad \forall l \in \mathcal{L} \tag{9d}$$

$$\sum_{j:(i,j) \in A^l} x_{ij}^l \leqslant \sum_{j:(l,j) \in A^l} x_{lj}^l \quad \forall i \in V^l, \forall l \in \mathcal{L} \tag{9e}$$

$$x_{ij}^l \in \{0,1\}. \quad \forall (i,j) \in A, \forall l \in \mathcal{L}$$

5. Adaptation of each model to include problem variants

In the following text we discuss how formulations for the KEP can be adapted to three variants: the problem with altruistic donors participating, the problem where compatible pairs are included into the pool, and the problem with one or more patients having multiple donors associated.

5.1. Inclusion of altruistic donors

The inclusion of altruistic donors in KEP IP formulations is discussed in [23]. Exchanges involving altruistic donors are labeled as *domino paired chains* (DPC) and, due to program regulations, the authors do only consider chains of lengths 1 and 2 (when one or two incompatible pairs are involved in the exchange), but their approach can be used for chains of any size. The problem is again modeled as a weighted directed graph with $n+m$ nodes: $V = \{1, \ldots, n+m\}$, n being the number of incompatible patient–donor pairs and m the number of altruistic donors. Let the nodes $\{1, \ldots, m\}$ represent the altruistic donors, nodes $\{m+1, \ldots, m+n\}$ represent the incompatible pairs, and a dummy patient compatible with all donors $j \in \{m+1, \ldots, m+n\}$ be associated with each altruistic donor. An example is given in Fig. 5.

Within this description denote by k' the maximum length of a chain (in terms of nodes) started by an altruistic donor and let $\mathcal{C}(k,k')$ define the set of all cycles in G with length at most k involving only incompatible pairs, and of all cycles with length at most k' with one altruistic donor. Replacing $\mathcal{C}(k)$ by $\mathcal{C}(k,k')$ the cycle formulation can be directly used to solve the problem.

For the other formulations the main impact is on the cardinality constraints that define the maximum size of the cycle. They will now have to be separated in two sets: one for cycles including one altruistic donor, and another for cycles that only include incompatible pairs.

For the edge formulation in Section 3.1 assume also that $k' \leqslant k$. The following additional set of constraints must be added:

$$\sum_{1 \leqslant j \leqslant k'} x_{i_j i_{j+1}} \leqslant k' - 1 \quad \forall (i_1, i_2, \ldots, i_{k'}, i_{k'+1}) \in \mathit{Paths\_a} \tag{10}$$

where $\mathit{Paths\_a}$ is the set of simple paths of length k' with $i_1 \in \{1, \ldots, m\}$ and $i_2, \ldots, i_{k'+1} \in \{m+1, \ldots, n+m\}$.

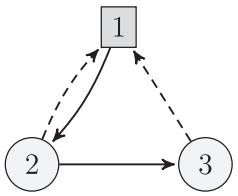

Fig. 5. Inclusion of altruistic donor with $m = 1$ and $n = 2$. Node 1 is the altruistic donor, and dashed arcs (2,1) and (3,1) are the arcs to the altruistic donor's dummy patient.

Similarly, for the edge-assignment formulation in Section 4.1 we define $L = n + m$. In addition to variables $y_i^l, l, i = m+1, \ldots, L$ for the incompatible pairs consider variables y_l^l for $l = 1, \ldots, m$ and y_i^l for $l = 1, \ldots, m$, $i = m+1, \ldots, L$. Then Eq. (3a) limiting the size of cycles is divided into the two following inequalities:

$$\sum_{i \in \{l\} \cup \{m+1, \ldots, L\}} y_i^l \leqslant k' \quad \forall l \in 1, \ldots, m \tag{11a}$$

$$\sum_{i \geqslant m+1} y_i^l \leqslant k \quad \forall l \in m+1, \ldots, L \tag{11b}$$

The same idea is applied to Eq. (7d), resulting in:

$$\sum_{(i,j) \in A: i,j \in \{l\} \cup \{m+1, \ldots, L\}} x_{ij}^l \leqslant k' \quad \forall l \in 1, \ldots, m \tag{12a}$$

$$\sum_{(i,j) \in A: i,j \geqslant m+1} x_{ij}^l \leqslant k \quad \forall l \in m+1, \ldots, L \tag{12b}$$

5.2. Inclusion of compatible pairs

Opening kidney exchange programs to compatible pairs will require slight modifications in the IP models discussed in this paper. The set of vertices V will now represent both compatible an incompatible pairs, $V_C \subset V$ denoting the subset of compatible pairs. For the cycle formulation, one will have to consider the existence of loops ii of size 1 for all vertices of set V_C. All the other formulations will have to consider that it is now possible that patient i gets a kidney from donor i. Therefore variables x_{ij} must be extended to include variables $x_{ii} \forall i \in V_C$.

5.3. Multiple donors

Within kidney exchange programs it is possible that instead of a single donor a patient has multiple donors associated. If that is the case, and if the patient is selected for kidney transplantation, a donor that would allow the cycle where the corresponding patient appears to be created will be selected. In such cases, depending on the scoring system used, two situations can occur: (i) the weight of an arc from a vertex i to a vertex j has a component that depends on which donor is chosen in each pair i and j (for instance, in the UK system this weight has a component that relates to the difference in age between the donors in i and j [23]): (ii) the weight of arc $(i,j) \in A$ does not depend on the donor selected in pair i.

For the first case, a possible way of modeling the problem is by considering different vertices for all donors of a patient P_i, each vertex representing the pair $\left(P_i, D_i^j\right), j = 1 \ldots ND_i$, where ND_i is the number of donors of patient P_i. More precisely denote by $\mathcal{P}$ the set of patients in the pool. For each patient $p \in \mathcal{P}$ let us define set $\mathcal{D}(p)$ as the set of vertices associated with patient p, the set of sets $\{D(p), p \in \mathcal{P}\}$ defines a partition of the set of vertices V.

For the edge formulation (see Section 3.1) the Eq. (1c) need to be replaced by:

$$\sum_{k\in\mathcal{D}(p)}\sum_{j:(k,j)\in A} x_{kj} \leqslant 1 \quad \forall p \in \mathcal{P} \tag{13}$$

These constraints guarantee that only one of the (possible) multiple donors associated with a patient will be selected. The same principle will have to be followed in the edge assignment formulation by replacing Eq. (6c) by the ones above. And similarly for the extended edge formulation by replacing Eq. (9c) by equations

$$\sum_{l\in\mathcal{L}}\sum_{k\in\mathcal{D}(p)}\sum_{j:(k,j)\in A} x^l_{kj} \leqslant 1 \quad \forall i \in \mathcal{P}. \tag{14}$$

Finally, for the cycle formulation two considerations must be made: all cycles in $\mathcal{C}(k)$ can contain at most one vertex with patient p, and Eq. (2b) must be replaced by:

$$\sum_{k\in\mathcal{D}(p)}\sum_{c:k\in V(c)} z_c \leqslant 1 \quad \forall p \in \mathcal{P} \tag{15}$$

For the second case the IP models discussed do not suffer any structural changes and are handled as follows: for all patients i with multiple donors, an arc (i,j) will exist if there is at least one donor in pair i compatible with patient j. If the arcs are weighted and two or more donors in i are compatible with patient j the largest weight associated to a transplant from i to j is assigned to the arc between nodes i and j. The associated IP models will be the same as described in previous sections but a final straightforward procedure will be required, if a multiple donor node appears in the optimal solution to determine which donor within this node should be selected. If only one donor is compatible with the associated recipient, he/she is selected. Otherwise for weighted arcs the donor associated with the maximum weight is selected; for unweighted graphs if more than one donor is compatible with the patient, additional criteria are required for the selection of a donor.

6. Linear relaxations and comparison of the bounds for different models

It is well known that the strength of the LP relaxation is one of the most important factors for a formulation to be effective when a LP based branch-and-bound algorithm is used as a resolution method. Let $\mathcal{IP}$ be some optimization program and A and B be two different integer linear formulations of $\mathcal{IP}$, which define linear relaxation upper bounds (for the maximization problem) $UB(A)$ and $UB(B)$. We say that formulation A *dominates* formulation B if $UB(A) \leqslant UB(B)$. In this section we present a comparison, from a theoretical point of view, of the strength of the formulations described in the previous section. A first result on interaction of the upper bounds of optimal solutions provided by linear relaxations of edge and cycle formulations (Theorem 1) was presented and proven in [2].

Theorem 1. *The cycle formulation dominates the edge formulation.*

As mentioned above the IP formulations with exponential number of variables or constraints sometimes provide better bounds with linear relaxations than compact formulations. We now show that the cycle formulation also dominates the extended edge formulation and that the extended edge formulation dominates the edge-assignment formulation.

Assume that the extended edge formulation satisfies the following properties: (i) $L = |V|$, denote by $\Lambda = \{1, \ldots, L\}$; (ii) the non eliminated variables are x^l_{ij} such that $i \geqslant l$ or $j \geqslant l$, and (iii) the model contains constraints (8). The following results remain true for the reduced extended edge formulation given in the second half of Section 4.2.

Theorem 2. *The cycle formulation dominates the extended edge formulation.*

Proof 1. For each l, let X^l be the set of vectors $x^l = \left(x^l_{ij}, (i,j) \in A\right)$ defined by constraints (7b), (7d), (7e), (8) and $\sum_{j:(i,j)\in A} x^l_{ij} \leqslant 1, \forall i \in V$. Each element of X^l corresponds either to a cycle of cardinality at most k, or to a set of disjoint cycles with a total number of edges not exceeding k. The extended edge formulation can be rewritten as:

$$\text{maximize} \sum_l \sum_{(i,j)\in A} w_{ij} x^l_{ij} \tag{16a}$$

$$\text{subject to} \sum_l \sum_{j:(i,j)\in A} x^l_{ij} \leqslant 1 \quad \forall i \in V \tag{16b}$$

$$x^l \in X^l \tag{16c}$$

Now let U^l be the set of vectors of X^l that induce at most one cycle. U^l can replace X^l in the above formulation, although to obtain an explicit formulation in variables x^l_{ij} constraints preventing multiple cycles would have to be added. Consider the following relaxation of the above model:

$$\text{maximize} \quad \sum_l \sum_{(i,j)\in A} w_{ij} x^l_{ij} \tag{17a}$$

$$\text{subject to} \quad \sum_l \sum_{j:(i,j)\in A} x^l_{ij} \leqslant 1 \quad \forall i \in V \tag{17b}$$

$$x^l \in conv(U^l) \quad \forall l \in \Lambda \tag{17c}$$

where $conv(U^l)$ denotes the convex hull of U^l. The optimal value of the above model is less than or equal to the optimal value of the LP relaxation of the extended edge model. Indeed, the optimal value of problem (17a), (17b) and (17c) is less than or equal to the optimal value of the linear relaxation of the problem (16a) and (16b) with $x^l \in U^l$. But this value is obviously less than or equal to the optimal value of the linear relaxation of problem (16a), (16b) and (16c) because $U^l \subseteq X^l$.

One way to write a linear program equivalent to model (17a), (17b) and (17c) is to replace $conv(U^l)$ by its extreme point representation. An extreme point of U^l is either the null vector or the inducing vector p^c of a cycle c in G of cardinality at most k, containing node l and not containing nodes $i < l$. Let $\mathcal{C}(k)$ be the set of all cycles of cardinality at most k and $\mathcal{C}^l$ be the set of cycles defined by U^l, that is $\mathcal{C}^l = \{c \in \mathcal{C}(k) : l \in V(c) \subseteq \{l, \ldots, n\}\}$ (recall that $V(c)$ denotes the set of vertices of cycle c). Now observe that $\{\mathcal{C}^l, l \in \Lambda\}$ is a partition of $\mathcal{C}(k)$, hence $x^l \in conv(U^l)$ if and only if there exist nonnegative scalars u_c, with $c \in \mathcal{C}^l$, such that $x^l = \sum_{c\in\mathcal{C}^l} u_c p^c$ and $\sum_{c\in\mathcal{C}^l} u_c \leqslant 1$. So $x^l \in conv(U^l)$, $l \in \Lambda$ can be rewritten as:

$$x^l_{ij} = \sum_{c\in\mathcal{C}^l:(i,j)\in c} u_c \quad \forall (i,j) \in A, \forall l \in \Lambda, \tag{18a}$$

$$\sum_{c\in\mathcal{C}^l} u_c \leqslant 1 \quad \forall l \in \Lambda, \tag{18b}$$

$$u_c \geqslant 0 \quad \forall c \in \mathcal{C}^l, \forall l \in \Lambda. \tag{18c}$$

Let $w_c = \sum_{(i,j)\in c} w_{ij}$ for $c \in \mathcal{C}(k)$. Using (18a), (18b) and (18c), the model (17a), (17b), (17c) can be written as follows.

$$\text{maximize} \sum_{c\in\mathcal{C}(k)} w_c u_c \tag{19a}$$

$$\text{subject to} \sum_{c\in\mathcal{C}(k):i\in V(c)} u_c \leqslant 1 \quad \forall i \in V, \tag{19b}$$

$$\sum_{c\in\mathcal{C}^l} u_c \leqslant 1 \quad \forall l \in \Lambda, \tag{19c}$$

$$u_c \geqslant 0 \quad \forall c \in \mathcal{C}^l, l \in \Lambda. \tag{19d}$$

It is straightforward to see that constraints (19c) are always satisfied with respect to constraints (19b). Thus problem (19a), (19b), (19c), (19d) is the cycle formulation and from the previous

discussion its optimal value is less than or equal to the value of the LP relaxation of the extended edge model. □

Theorem 3. *The extended edge formulation dominates the edge-assignment formulation.*

Proof 2. Let $\bar{x}^l_{ij}, (i,j) \in A, l \in \Lambda$ be an optimal solution of the linear relaxation of the extended edge model 7a, (7b)–(7e). We build a feasible solution for the LP relaxation of the edge-assignment model with same objective value.

Define the variables $\bar{x}_{ij}$ and $\bar{y}^l_i$ as:

$$\bar{x}_{ij} = \sum_{l \in \Lambda} \bar{x}^l_{ij} \quad \forall (i,j) \in A \tag{20}$$

$$\bar{y}^l_i = \sum_{j:(i,j) \in A} \bar{x}^l_{ij} \quad \forall i \in V, i \geqslant l \tag{21}$$

The verification that the objective value is the same for both solutions uses (20) and it is straightforward. Also, constraints (1b) and (1c) follow directly from (7b) and (7c) respectively, and (20). Similarly, (3a) follows from (7d) and (21), and (4a) is obtained from summing both sides of (21) over l and both sides of (20) over j. We show next that (4b) is satisfied.

By (20) and (21) $\bar{y}^l_i + \bar{x}_{ij} = \sum_{p:(i,p) \in A} \bar{x}^l_{ip} + \sum_{l \in \Lambda} \bar{x}^l_{ij}$. Now observe that the only common variable in the two previous sums is $\bar{x}^l_{ij}$. Hence $\bar{y}^l_i + \bar{x}_{ij} \leqslant \sum_{l \in \Lambda} \sum_{p:(i,p) \in A} \bar{x}^l_{ip} + \bar{x}^l_{ij} \leqslant 1 + \bar{y}^l_i$, the last inequality following from (7c) and from (21) and the nonnegativity of the variables.

Finally, $0 \leqslant \bar{y}^l_i \leqslant 1$ and $0 \leqslant \bar{x}_{ij} \leqslant 1$ are a consequence of the nonnegativity of $\bar{x}^l_{ij}$ and constraints (1c) and (4a), already verified. Thus there always exists a feasible point for the edge-assignment formulation which provides the same objective value as an optimal solution for the extended edge formulation. Hence we conclude that the value of the LP relaxation of the edge-assignment formulation is greater than or equal to the value of the LP relaxation of the extended edge formulation. □

Next we show that the extended edge formulation and the edge formulation do not dominate each other.

Remark 1. The edge formulation does not dominate the extended edge formulation.

Proof 3. We present an example in which the value of the LP relaxation of the edge formulation is larger than the one of the extended edge formulation. Let $k = 3$, the set of nodes (incompatible pairs) be $V = \{1,2,3,4,5\}$ and the set of arcs be $A = \{(1,2),(1,3),(1,4),(2,3),(3,1),(3,4),(4,1),(4,5),(5,1)\}$, $w_{ij} = 1$ $\forall (i,j) \in A$ – Fig. 6. The optimal value for this instance is 3, given for e.g. by cycle 1–4–5–1.

Observe that all cycles contain vertex 1. Hence in the extended edge formulation $x^l_{ij} = 0$ for all (i,j) and $l \geqslant 2$, and the optimal LP value is 3. The optimal LP solution for the edge formulation is $x_{12} = x_{13} = x_{23} = x_{41} = x_{45} = x_{51} = 0.5$, $x_{34} = 1$, and $x_{31} = x_{14} = 0$, with value 4. □

Remark 2. The extended edge formulation does not dominate the edge formulation.

Proof 4. Consider the example illustrated by Fig. 7, where $k = 3$, the set of nodes is $V = \{1,2,3,4\}$ and the set of arcs is $A = \{(1,2),(1,3),(2,3),(3,1),(3,4),(4,1),(4,2)\}$, $w_{ij} = 1$ $\forall (i,j) \in A$. The optimal value for this instance is 3.

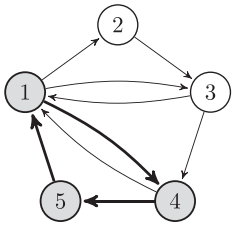

Fig. 6. An optimal solution for $k = 3$ is, for e.g. the cycle 1–4–5–1.

The optimal LP solution for the edge formulation is $x_{12} = x_{23} = x_{34} = x_{41} = 0.66(7)$, $x_{13} = x_{31} = 0.33(3)$, and $x_{42} = 0$, with value 3.33(3). An optimal LP solution for the extended edge formulation is $x^1_{12} = x^1_{23} = x^1_{34} = x^1_{41} = 0.75$, $x^2_{23} = x^2_{34} = x^2_{42} = 0.25$ and $x^l_{ij} = 0$ for all other variables, with value 3.75. □

Theorem 3 and Remark 2 imply that the edge-assignment formulation does not dominate the edge formulation. It is an open question whether the edge formulation dominates the edge-assignment formulation. However, if the reduction procedure is applied (Section 4.1), the edge formulation does not dominate edge-assignment. The example presented by Fig. 6 can be considered. The optimal value of LP solution for the edge formulation is 4, and the reduced edge-assignment formulation has value 3.

Despite the results presented in this section, compact formulations can turn out to be effective computationally because of their polynomial size, as will be shown in the computational study presented in the next section.

7. Computational analysis

Computational experiments were carried out to compare the proposed and known formulations in terms of time needed to find an optimal solution and of the LP gaps with respect to upper bounds of linear relaxations of the models. CPU times and bounds were obtained with CPLEX 12.2 on a computer with a Quad-Core Intel Xeon processor at 2.66 gigahertz, 16 gigabytes of RAM and running Mac OS X 10.6.6. Only one core of the processor was assigned to these experiments.

Two generators where used to create the instances for the computational study:

(1) the instance generator described in [35], which creates random graphs based on probability of blood type and of donor–patient compatibility. These instances will be referred to as *blood-type test instances*;
(2) a random generator implemented by the authors of this paper used to generate graphs with three different densities: low, medium and high. To do that different values are set for the probability of having 1 (i.e. compatibility) on each position of the adjacency matrix of a graph. *Low* density graphs are generated with probability 0.2. This value leads to an average density similar to the one obtained for the blood-type test instances. The probability is set to 0.5 for *medium* density, and to 0.7 for *high* density graphs.

The computational analysis was performed as follows. First, for cases with up to 50 nodes and k ranging from 3 to 6, fifty instances of the same size were generated with both generators for the three density levels.[3] The performance of all formulations was tested for all instances. Besides that, for this set of problem instances we compared

[3] Currently implemented kidney exchange programs work in general with a maximum value of $k = 3$. However this value has already been exceeded by large: the maximum number of simultaneous transplantations performed up to now being 6 [6].

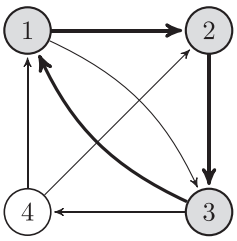

Fig. 7. An optimal solution for $k = 3$ is the cycle 1–2–3–1.

the average number of constraints and variables for different models and the percentage reduction in size for the edge-assignment and extended edge formulations when reduction procedures are implemented (see Sections 4.1, 4.2). Afterwards the formulations with generally better performance on "small" instances were selected for testing on larger problem instances ($n > 50$). Ten instances were generated for different large problem instance sizes.

In the remainder of this document the following notation will be used to refer to each formulation: *E* – edge formulation; *C* – cycle formulation; *EA* – reduced edge-assignment formulation; *EE* – reduced extended edge formulation. Computational results are provided in Tables 1 and 3 where:

- n is the number of nodes in the graph;
- k is the maximum length of the cycle;
- t_c and t_p are the average CPU times (in seconds) to find all cycles and paths, respectively. The time needed to carry out reduction procedures for the *EA* and the *EE* formulations was less than 1 s for all small instances and less than 3 seconds for all large instances, and therefore it is not shown in the tables;
- T is the average CPU time the solver CPLEX [16] needed to reach optimal solutions for the given set of instances (50 instances for small problems, 10 for large problems); maximum CPU time was set to 1800 seconds for all formulations;
- *#opt* is the number of instances from each set which were solved to optimality within the time limit (1800 seconds). Whenever this information is not provided in Table 1 it means that all instances of the set were solved;
- *gap* is the average LP gap associated to a formulation: $gap = \frac{UB-Opt}{Opt} * 100\%$, where *UB* is the upper bound provided by

Table 1
Results for small instances.

n	k	*C*		*E*			*EA*			*EE*		*C*			*E*			*EA*			*EE*	
		t_c/T	*gap*	t_p/T	*#opt*	*gap*	T	*#opt*	*gap*	T	*gap*	t_c/T	*#opt*	*gap*	t_p/T	*#opt*	*gap*	T	*#opt*	*gap*	T	*gap*
		Blood-type test instances										*Medium density test instances*										
10	3	0/0	0.0	0/0		0.5	0		1.0	0	0.0	0/0		0.6	0/0		0.9	0		1.0	0	0.7
	4	0/0	0.0	0/0		0.0	0		0.0	0	0.0	0/0		0.0	0/0		0.2	0		0.2	0	0.0
	5	0/0	0.0	0/0		0.0	0		0.0	0	0.0	0/0		0.0	0/0		0.0	0		0.0	0	0.0
	6	0/0	0.0	0/0		0.0	0		0.0	0	0.0	0/0		0.0	0/0		0.0	0		0.0	0	0.0
20	3	0/0	0.0	0/1		5.6	0		6.5	0	2.8	0/0		0.0	0/1		0.0	2		0.0	0	0.0
	4	0/0	0.0	0/0		1.5	0		1.9	0	1.1	0/0		0.0	0/5		0.0	1		0.0	0	0.0
	5	0/0	0.1	1/0		0.7	0		0.9	0	0.8	0/0		0.0	2/55		0.0	1		0.0	0	0.0
	6	0/0	0.0	9/1	47(47)	0.2	0		0.2	0	0.2	0/3		0.0	–			0		0.0	0	0.0
30	3	0/0	0.0	0/0		4.0	6		4.9	0	1.2	0/0		0.0	0/15		0.0	162	47	0.0	0	0.0
	4	0/0	0.0	1/1		0.6	1		0.7	0	0.5	0/0		0.0	2/138	49	0.0	30		0.0	1	0.0
	5	0/0	0.0	10/2	42(42)	0.0	0		0.0	0	0.0	0/5		0.0	–		–	4		0.0	0	0.0
	6	2/0	0.0	185/0	12(12)	0.0	0		0.0	0	0.0	4/267	49	0.0	–		–	2		0.0	0	0.0
40	3	0/0	0.1	0/48		5.6	85	45	6.1	0	2.7	0/0		0.0	1/171		0.0	564	24	0.0	1	0.0
	4	0/0	0.0	3/28		0.9	49		1.0	0	0.9	0/1		0.0	6/624	39	0.0	203	40	0.0	2	0.0
	5	1/0	0.0	54/4	18(18)	1.5	2		0.6	0	0.6	1/67		0.0	–		–	106	49	0.0	2	0.0
	6	11/1	0.0	1727/1	4(4)	2.4	3		0.2	0	0.2	–		–	–		–	16		0.0	1	0.0
50	3	0/0	0.0	0/134	49	3.6	112	30	3.9	0	2.0	0/0		0.0	1/440	47	0.0	745	3	0.0	4	0.0
	4	0/0	0.0	12/109	47	0.6	42	44	0.7	1	0.3	0/4		0.0	–		–	645	18	0.0	5	0.0
	5	2/0	0.1	213/7	3(3)	0.0	14	48	0.4	1	0.2	4/385	48	0.0	–		–	116	34	0.0	6	0.0
	6	45/6	0.0	–		–	8		0.1	1	0.0	-		-	-		-	62	46	0.0	3	0.0
		Low density test instances										*High density test instances*										
10	3	0/0	0.2	0/0		21.8	0		33.2	0	0.3	0/0		0.0	0/0		0.0	0		0.0	0	0.0
	4	0/0	0.5	0/0		11.6	0		15.2	0	0.9	0/0		0.0	0/0		0.0	0		0.0	0	0.0
	5	0/0	0.0	0/0		7.0	0		8.2	0	0.2	0/0		0.0	0/0		0.0	0		0.0	0	0.0
	6	0/0	0.2	0/0		1.1	0		1.7	0	1.1	0/0		0.0	0/1		0.0	0		0.0	0	0.0
20	3	0/0	1.4	0/0		21.5	1		23.7	0	2.6	0/0		0.0	0/1		0.0	2		0.0	0	0.0
	4	0/0	0.4	0/0		5.7	1		6.2	0	1.5	0/0		0.0	1/16		0.0	1		0.0	0	0.0
	5	0/0	0.6	1/1		1.9	1		1.9	0	1.2	0/2		0.0	–		–	0		0.0	0	0.0
	6	0/0	0.4	7/1		0.8	0		0.8	0	0.7	1/52		0.0	–		–	0		0.0	0	0.0
30	3	0/0	0.5	0/44		4.9	714	33	4.9	0	1.1	0/0		0.0	0/9		0.0	32		0.0	1	0.0
	4	0/0	0.1	0/33		0.2	348	42	0.2	1	0.1	0/1		0.0	–		–	5		0.0	1	0.0
	5	0/0	0.0	8/65		0.0	86		0.0	1	0.0	1/66		0.0	–		–	2		0.0	0	0.0
	6	2/0	0.0	188/226		0.0	17		0.0	1	0.0	–		–	–		–	1		0.0	0	0.0
40	3	0/0	0.0	0/512	20	0.1	–		0.5	0	0.1	0/0		0.0	1/64		0.0	238	36	0.0	2	0.0
	4	0/0	0.0	2/571	30	0.0	851	9	0.0	4	0.0	0/5		0.0	–		–	38	49	0.0	2	0.0
	5	1/1	0.0	–		–	453	30	0.0	5	0.0	4/66	34(41)	0.0	–		–	12		0.0	1	0.0
	6	10/2	0.0	–		–	246	45	0.0	3	0.0	–		–	–		–	4		0.0	1	0.0
50	3	0/0	0.0	0/506	3	0.0	–		0.0	0	0.0	0/0		0.0	3/238	48	0.0	250	15	0.0	10	0.0
	4	0/0	0.0	–		–	–		0.0	20	0.0	–		–	–		–	130	39	0.0	7	0.0
	5	2/1	0.0	–		–	–		0.0	15	0.0	–		–	–		–	25		0.0	5	0.0
	6	42/11	0.0	–		–	618	20	0.0	7	0.0	–		–	–		–	47		0.0	2	0.0

Table 2
Sizes of formulations and reduction for EA and EE. "Low" and "high" density test instances.

n	k	C		E		EA				EE			
		#var	#con	#var	#con	#var	#con	rv%	rc%	#var	#con	rv%	rc%
Low density test instances													
10	3	3	2	17	55	22	53	70.6	76.4	7	16	96.4	86.1
	4	5	4	17	62	25	65	66.3	71.2	12	24	93.9	80.0
	5	7	4	17	63	27	73	63.3	67.7	16	28	91.8	75.9
	6	8	5	17	55	28	75	62.0	66.7	18	30	90.5	74.2
30	3	82	28	174	5215	280	1318	56.2	68.5	169	242	96.8	74.8
	4	336	30	174	26,634	401	2354	37.4	43.6	498	484	90.5	49.6
	5	1377	30	174	131,770	496	2978	22.5	28.5	1008	674	80.8	29.8
	6	5681	30	174	626,695	539	3225	15.6	22.4	1412	761	73.0	20.7
50	3	363	50	491	44,579	851	6782	51.8	62.5	684	770	97.2	70.4
	4	2596	50	–	$>3\times10^6$	1342	13,059	24.0	27.8	2672	1752	89.1	32.6
	5	19,010	50	–	$>3\times10^6$	1601	15,628	9.3	13.6	6140	2271	75.0	12.7
	6	142,190	50	–	$>3\times10^6$	1665	16,332	5.7	9.6	7780	2399	68.3	7.7
High density test instances													
10	3	100	10	62	1667	110	488	6.4	10.6	139	105	77.7	12.3
	4	384	10	62	6856	114	506	2.8	7.1	211	114	66.1	5.3
	5	1331	10	62	23,732	114	507	2.7	7.0	228	114	63.1	5.1
	6	4075	10	62	65,993	114	507	2.7	7.0	229	114	63.0	5.1
30	3	2949	30	606	221,941	1026	12,290	4.2	6.0	3341	870	81.6	9.3
	4	41,556	30	–	$>3\times10^6$	1066	12,887	0.4	1.4	5731	950	68.5	1.0
	5	600,569	30	–	$>3\times10^6$	1066	12,896	0.4	1.4	6251	951	65.6	0.9
	6	$>3\times10^6$	–	–	$>3\times10^6$	1066	12,896	0.4	1.4	6252	951	65.6	0.9
50	3	14,157	50	1719	1,911,261	2883	56,974	3.7	5.0	15,272	2377	82.2	8.6
	4	$>3\times10^6$	–	–	$>3\times10^6$	2992	59,750	0.1	0.3	26,817	2595	68.8	0.2
	5	$>3\times10^6$	–	–	$>3\times10^6$	2992	59,762	0.1	0.3	29,272	2595	66.0	0.2
	6	$>3\times10^6$	–	–	$>3\times10^6$	2992	59,764	0.1	0.3	29,273	2595	65.9	0.2

Notations: – #*var* and #*con* are the average number of variables and constraints for a given n for different values of k;
– *rv%* and *rc%* are the average relative reductions on the number of variables and constraints in the *EA* and *EE* formulations after implementing reduction procedures.
Dashes for the formulations *C* and *E* mean that no test instance out of 50 was considered due to the bound on number of cycles or paths.

the linear relaxation of the formulation and *Opt* is the optimal value of the problem.

Values 0 in tables related to CPU time mean that the solver took less than 1 s to solve the instances.

Since the number of paths associated with the edge formulation increases sharply for larger values of k ($k = 5$ and $k = 6$), a bound of 3 million was set on the number of paths to be generated. The formulation was not studied for instances where the number of paths exceeded that value. The same bound was used for the number of cycles in the cycle formulation. With respect to this limitation in column #*opt* we show in parenthesis the number of instances out of 50 (or 10 for larger instances) that were studied, if necessary.

7.1. Small test instances

Test instances of 10, 20, 30, 40 and 50 nodes were created with the two generators, for the second one for the three graph densities. Fifty instances of each size were considered. Computational results are presented in Table 1.

The results for *blood-type test instances* clearly show the dominance of the *C* and *EE* formulations, both in terms of effectiveness and efficiency: all instances are solved to optimality in general with less CPU time than the *E* and *EA* formulations. The considerable increase in CPU time for the cycle formulation when $k = 6$ and $n = 40, 50$ is caused by the time needed to enumerate all the cycles. As shown, the smaller average *gap* is associated to the *C* formulation.

Although the density of the graphs of *low density test instances* is of the same order of magnitude of the previously reported ones, the *E* and *EA* formulations performed very poorly for larger instances: in most cases the *EA* formulation exceeded the maximum CPU time for all test instances of a given size (represented by (–) in Table 1); in several cases the *E* formulation was not also considered, either because the CPU time or the maximum number of paths were exceeded. The additional difficulty raised up by these instances may be partially explained by the larger LP gaps obtained for smaller problems. Again, the *C* and *EE* formulations dominate the others, both formulations having solved to optimality all instances.

The importance of developing compact formulations is reflected in the results obtained for *medium density test instances*, and corroborated and strengthened by *high density test instances*. In this cases the *EE* formulation proved to be extremely efficient at solving larger problem instances. It was capable of solving to optimality the both complete sets. The *EA*, although performing worse than the *EE*, still solved to optimality almost 95% of the high density instances considered.

Furthermore, none of the compact formulations suffered the "curse of dimensionality" that affects both the *E* and the *C* formulations, which exceeded either the maximum number of cycles/paths or CPU times for larger instances. In fact, the problems raised up by non-compact formulations become more evident for high density test instances: the number of cycles for the cycle formulation exceeds the allowed limit for all test instances of size 50, for values of $k = 4, 5, 6$; the edge formulation was not run even for instances with 20 nodes and $k = 5, 6$ because number of paths exceed the limit of 3 million previously set.

It is also worth mentioning that although for the blood-type and low density instances most of the CPU time associated to the *C* formulation was spent at generating cycles, in this case it was spent in the optimization phase.

Table 2 shows the average number of variables and constraints for problems of different size for the low and high density test instances, as well as the percentage of reduction of the number of variables and constraints for the *EA* and *EE* formulations after implementing reduction routines presented in Sections 4.1 and 4.2. Evidently the impact of the reduction is high, in particular

Table 3
Results for large instances.

n	k	C				EE		
		t_c	T	#opt	gap	T	#opt	gap
Blood-type test instances								
70	3	0	0	10	0.0	2	10	1.8
100		0	0	10	0.1	3	10	0.8
200		1	0	10	0.0	1221	4	0.4
300		3	2	10	0.0	–	0	–
500		23	11	10	0.0	–	0	–
800		141	148	10	0.0	–	0	–
900		223	398	9(9)	0.0	–	0	–
1000		343	479	7(7)	0.0	–	0	–
70	4	0	0	10	0.0	11	10	0.5
100		2	0	10	0.0	53	10	0.0
200		42	25	10	0.0	–	0	0.0
70	5	10	2	10	0.0	7	10	0.0
100		67	13	10	0.0	72	10	0.0
70	6	370	25	9(9)	0.0	4	10	0.0
100		3255	19	1(1)	0.0	102	10	0.0
Low density test instances								
70	3	0	0	10	0.0	5	10	0.0
100		0	0	10	0.0	52	10	0.0
300		3	11	10	0.0	–	0	–
500		26	605	7(7)	0.0	–	0	–
70	4	0	1	10	0.0	385	10	0.0
100		2	5	10	0.0	–	0	–
200		44	341	8(8)	0.0	–	0	–
70	5	10	10	10	0.0	371	10	0.0
100		64	226	10	0.0	–	0	0.0
70	6	368	211	10	0.0	90	10	0.0
100		–	–	0	–	240	4	0.0
Medium density test instances								
70	3	0	0	10	0.0	127	10	0.0
100		0	4	10	0.0	270	5	0.0
300		5	279	10	0.0	–	0	–
400		13	204	4(4)	0.0	–	0	–
70	4	1	25	10	0.0	138	10	0.0
100		3	516	9(9)	0.0	460	7	0.0
70	5	–	–	0	–	45	10	0.0
100		–	–	0	–	360	7	0.5[a]
70	6	–	–	0	–	33	10	0.0
100		–	–	0	–	177	8	0.5[a]
High density test instances								
70	3	0	2	10	0.0	161	10	0.0
100		0	5	10	0.0	505	7	0.6
200		2	167	10	0.0	–	0	–
70	4	2	301	9	0.0	90	10	0.0
100		–	–	0	–	419	7	0.3[a]
70	5	–	–	0	–	37	10	0.0
100		–	–	0	–	402	8	0.2
70	6	–	–	0	–	39	10	0.0
100		–	–	0	–	247	10	0.0

[a] For the test instances which were not solved to optimality within the time limit with any formulation the gap value for the best found lower bound is given by $gap = \frac{UB-LB}{LB} * 100\%$, where LB is the best found lower bound and UB is the LP upper bound for the optimal value.

for the low density test instances. No results are provided for these formulations before reduction as they were clearly worse.

The results of the computational study for small instances show that in general CPU times increase with increasing k and graph density for the cycle formulation; however they decrease with increasing density for the other models and decrease with increasing k for compact formulations. Increasing times for the cycle and edge formulations can be justified by the increasing number of cycles/variables and paths/constraints; whereas decreasing times for other models could be explained by smaller gaps. Indeed gaps decrease for *EE* and *EA* with k and density, while remaining approximately constant for *C*. This decrease in gaps may be explained as follows: *EE* and *EA* are exact formulations if the problems are uncapacitated, that is, for k sufficiently large there is always an LP solution that is optimal for the integer program. When k increases the problems become closer to being uncapacitated so the LP solutions are closer to being integral and the LP values are closer to the integer optima. When the density increases the explanation is not so clear; probably since more feasible cycles are available, it may be easier for LP solutions to have some entire feasible cycles in their composition, i.e. more variables equal to one.

7.2. Large scale test instances

This section reports the results obtained for the *C* and *EE* formulations, for problems ranging from 70 to 1000 pairs (see Table 3); 10 instances of each size were generated. These formulations were selected because they were the dominant for at least one set of the previous computational simulations: blood-type, low, medium or high density graphs.

Again for blood-type and low density graphs the *C* formulation dominates over the *EE* for lower values of k, being able to solve some problems with 1000 pairs for $k = 3$. However, as in the previous analysis results for medium and high density graphs confirm the effectiveness of the compact formulation on dense graphs with large values of k. With the cycle formulation it was possible to solve instances for $k = 3$ and some instances for $k = 4$. This formulation was not capable of solving any instance with $k > 4$ within the limit on number of cycles considered. For these values of k the *EE* was more efficient being able to solve to optimality some instances.

To conclude, these results clearly indicate that appropriate methods have to be implemented if one wishes to use any of the formulations discussed in larger pools and for bigger size cycles.

8. Conclusions

This paper presents two new formulations for the Kidney Exchange Problem – edge-assignment and extended edge formulations – that have the advantage over other formulations proposed in the literature of having polynomially bounded number of constraints and variables. A proof of dominance of some formulations over others is also given and a discussion on the adaptability of each formulation to different problem variants is provided. Finally, computational results that compare the previous and proposed formulations in terms of time needed to find an optimal solution and of the gaps of linear relaxations upper bounds of the models are provided.

Computational results show that the edge formulation has a bad performance and that it is not effective at solving instances larger than 50 pairs. The non-compact cycle formulation is very efficient for low density graphs with small values of k. However for larger values of k and especially if graphs are denser this formulation becomes inefficient. In such cases compact formulations provide better results – in particular the extended edge formulation – and are able to solve larger problems. Therefore, although we prove in this paper that linear relaxations of the compact formulations do not provide better upper bounds for optimal solutions than the cycle formulation, computational results reinforce the idea that compact formulations are of practical relevance.

As future work an interesting direction is to use decomposition methods on the extended edge formulation, in order to solve larger

problems. The adaptation of these models to dynamic environments will also be the subject of additional research.

Acknowledgments

We would like to thank Prof. João Pedro Pedroso from the University of Porto, Portugal, Prof. Filipe Alvelos from University of Minho, Braga, Portugal, and Prof. David Manlove from the University of Glasgow, Scotland, for their useful comments. Dorien de Regt participated in the early stage of this work as part of her master's project at the Delft University of Technology.

References

[1] Abraham, D., 2009. Matching Markets: Design and Analysis. Ph.D. Thesis. Carnegie-Mellon University, School of Computer Science.
[2] Abraham, D., Blum, A., Sandholm, T., 2007. Clearing algorithms for Barter exchange markets: Enabling nationwide kidney exchanges. In; Proceedings of the 8th ACM Conference on Electronic Commerce, June 13–16, pp. 295–304.
[3] Anderson, R., Ashlagi, I., Gamarnik, D., 2012. A dynamic model of kidney exchange programs. In: The 2012 Stochastic Networks Conference, June 18–22.
[4] I. Ashlagi, D. Gilchrist, A. Roth, M. Rees, Nonsimultaneous chains and dominos in kidney paired donation – revisited, American Journal of Transplantation 11 (5) (2011) 984–994.
[5] Awasthi, P., Sandholm, T., 2009. Online stochastic optimization in the large: application to kidney exchange. In: Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09), pp. 405–411.
[6] BBC, 2008. BBC news website. six-way kidney transplant first (9/04/2008) <http://news.bbc.co.uk/1/health/7338437.stm> (last accessed December, 2012).
[7] P. Biro, D. Manlove, R. Rizzi, Maximum weight cycle packing in directed graphs, wiht application to kidney exchange programs, Discrete Mathematics, Algorithms and Applications 1 (4) (2009) 499–517.
[8] M. de Klerk, K. Keizer, F. Claas, B. Haase-Kromwijk, W. Weimar, The Dutch national living donor kidney exchange program, American Journal of Transplantation 5 (2005) 2302–2305.
[9] M. de Klerk, M. Witvliet, B. Haase-Kromwijk, F. Claas, W. Weimar, Highly efficient living donor kidney exchange program for both blood type and crossmatch incompatible donor–recipient combinations, Transplantation 82 (12) (2006) 1616–1620.
[10] Dickerson, J., Procaccia, A., Sandholm, T., 2011. Optimizing kidney exchange with transplant chains: theory and reality. In: AAMAS-12: Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems, June.
[11] J. Edmonds, Paths, trees, and flowers, Canadian Journal of Mathematics 17 (1965) 449–467.
[12] S. Gentry, R. Montgomery, D. Segev, Kidney paired donation: fundamentals, limitations, and expansions, American Journal of Kidney Disease 57 (1) (2010) 144–151.
[13] S. Gentry, R. Montgomery, B. Swihart, D. Segev, The roles of dominos and nonsimultaneous chains in kidney paired donation, American Journal of Transplantation 9 (2009) 1330–1336.
[14] S. Gentry, D. Segev, M. Simmerling, R. Montgomery, Expanding kidney paired donation through participation by compatible pairs, American Journal of Transplantation 7 (10) (2007) 2361–2370.
[15] A. Gurkan, S. Kacar, C. Varilsuha, S. Tilif, I. Coker, C. Karaca, M. Karaoglan, Exchange kidney transplantation: a good solution in living kidney transplantation, Transplantation Proceedings 36 (2004) 2952–2953.
[16] IBM, 2012. ILOG CPLEX optimizer <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/> (last accessed in December, 2012).
[17] R. Johnson, J. Allen, S. Fuggle, J. Bradley, C. Rudge, Early experience of paired living kidney donation in the United Kingdom, Transplantation 86 (12) (2008) 1672–1677.
[18] I. Kaplan, J. Houp, M. Leffell, J. Hart, A. Zachary, A computer match program for paired and unconventional kidney exchanges, American Journal of Transplantation 5 (2005) 2306–2308.
[19] L. Kranenburg, W. Zuidema, W. Weimar, M. Hilhorst, J. IJzermans, J. Passchier, J. Busschbach, Strategies to advance living kidney donation, Progress in Transplantation 19 (2009) 71–75.
[20] J. Kwak, O. Kwon, K.L. KS, C. Kang, H. Park, J. Kim, Exchange-donor program in renal transplantation: a single-center experience, Transplant Proc 31 (1999) 344–345.
[21] M. Lucan, Five years of single-center experience with paired kidney exchange transplantation, Transplantation Proceedings 39 (2007) 1371–1375.
[22] A. Mahendran, P. Veitch, Paired exchange programs can expand the live kidney donor pool, British Journal of Surgery 94 (2007) 657–664.
[23] D. Manlove, G. O'Malley, Paired and altruistic kidney donation in the UK: algorithms and experimentation, Lecture Notes in Computer Science 7276 (2012) 271–282.
[24] F. Margot, Symmetry in integer linear programming, in: M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, L. Wolsey (Eds.), 50 Years of Integer Programming 1958–2008, Springer, 2009, pp. 647–686.
[25] R. Montgomery, S. Gentry, W. Marks, D. Warren, J. Hiller, J. Houp, A. Zachary, J. Melancon, W. Maley, H.R.C. Simpkins, D. Segev, Domino paired kidney donation: a strategy to make best use of live non-directed donation, The Lancet 368 (9533) (2006) 419–421.
[26] G. Nemhauser, L. Wolsey, Integer and Combinatiorial Optimization, A Wiley-Interscience Publication, 1999.
[27] F. Rapaport, The case for a living emotionally related international kidney donor exchange registry, Transplant Proc 18 (1986) 5–9.
[28] M. Rees, J. Kopke, R. Pelletier, D. Segev, M. Rutter, A. Fabrega, J. Rogers, O. Pankewycz, J. Hiller, A. Roth, T. Sandholm, M. Ünver, R. Montgomery, A nonsimultaneous, extended, altruistic-donor chain, The New England Journal of Medicine 360 (2009) 1096–1101.
[29] L. Ross, The ethical limits in expanding living donor transplantation, Kennedy Institute of Ethics Journal 16 (2) (2006) 151–172.
[30] L. Ross, E. Woodle, Ethical issues in increasing living kidney donations by expanding kidney paired exchange programs, Transplantation 69 (2000) 1539–1543.
[31] L. Ross, S. Zenios, Practical and ethical challenges to paired exchange programs, American Journal of Transplantation 4 (2004) 1553–1554.
[32] A. Roth, T. Sönmez, M. Ünver, Kidney exchange, Quarterly Journal of Economics 119 (2) (2004) 457–488.
[33] A. Roth, T. Sönmez, M. Ünver, A kidney exchange clearinghouse in New England, Practical market design 95 (2) (2005) 376–380.
[34] A. Roth, T. Sönmez, M. Ünver, Efficient kidney exchange: coincidence of wants in markets with compatibility-based preferences, The American Economic Review 97 (3) (2007) 828–851.
[35] S. Saidman, A. Roth, T. Sönmez, M. Ünver, F. Delmonico, Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges, Transplantation 81 (2006) 773–782.
[36] D. Segev, S. Gentry, D. Warren, B. Reeb, R. Montgomery, Kidney paired donation and optimizing the use of live donor organs, The Journal of the American Medical Association 293 (15) (2005) 1883–1890.
[37] G. Thiel, P.V.L. Gurke, T. Gasser, K. Lehmann, T. Voegele, A. Kiss, G. Kirste, Crossover renal transplantation: hurdles to be cleared!, Transplant Proc 33 (2001) 811–816
[38] M. Ünver, Dynamic kidney exchange, Review of Economic Studies 77 (2010) 372–414.
[39] J. Veale, G. Hil, National kidney registry: 213 transplants in three years, Clinical Transplants (2010) 333–344.
[40] C. Wallis, K. Samy, A. Roth, M. Rees, Kidney paired donation, Nephrol Dial Transplant 26 (2011) 2091–2099.
[41] S. Zenios, Optimal control of a paired-kidney exchange program, Management Science 48 (3) (2002) 328–342.
[42] S. Zenios, G. Chertow, L. Wein, Dynamic allocation of kidneys to candidates on the transplant waiting list, Operations Research 48 (4) (2000) 549–569.