Centro de Investigação Operacional

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

Miguel Constantino, Xenia Klimentova, Ana Viana and Abdur Rais

CIO – Working Paper 4/2012

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] *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*

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, and the underlying optimization problems can be formulated as integer programming models.

Previously proposed models for kidney exchange programs have exponential number 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 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.

Keywords: Kidney exchange program, integer programming, combinatorial optimization, healthcare.

1. Introduction

Kidney transplants are essential for survival of many patients suffering from kidney failures, 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. Figure 1 illustrates the simplest case with only two pairs, (P_1, D_1) and (P_2, D_2),

Email addresses: mfconstantino@fc.ul.pt (Miguel Constantino), xenia.klimentova@inescporto.pt (Xenia Klimentova), aviana@inescporto.pt (Ana Viana), abdur.rais@dps.uminho.pt (Abdur 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

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 (arrowed lines represent the exchange in the figure).

Kidney exchange programs have already been introduced in many countries, including South Korea [1], Switzerland [2], Turkey [3], Romania [4], The Netherlands [5, 6, 7], UK [8, 9] and the US [10, 11, 12, 13]. 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.

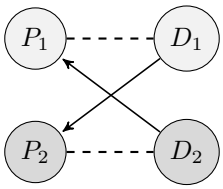


Figure 1: A 2-way exchange.

The objective for optimization in a kidney exchange program is generally to maximize the collective benefit for a given pool of incompatible pairs, usually measured by the number of possible kidney exchanges [11, 5]. – in the entirety of the paper we will refer to this optimization problem as the *Kidney Exchange Problem* (KEP). Although such 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 [14] and/or the quality-adjusted life expectancy of transplant candidates [15].

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 Figure 1 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 donor from one incompatible pair gives one kidney to the recipient in the other pair and vice versa. One can note that size of the exchange cycle can be increased. For example, the 3-*way* exchange presented in Figure 2 allows 3 patients to get transplants instead of 2; solid lines here represent compatibilities and arrowed lines represent the actual exchanges that derive maximum collective benefit.

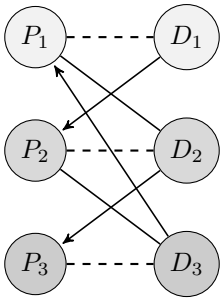


Figure 2: A 3-way exchange.

Generally k-*cycle* exchanges with $k \geq 3$ can be better for optimization as it has 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, i.e., k is not fixed ($k = n$), the problem turns into an assignment problem and can be solved in polynomial time [12]. But 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 [12]. Second, because the 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 cancelled, 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 [16] 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 [17] and difficult to solve efficiently when a problem instance is large.

Current work on solving the KEP focuses mostly on Integer Programming (IP) formulations. Two IP models addressed in this paper as "edge formulation" and "cycle formulation" were proposed independently in [12] and [18]. Despite the very good results reported for the cycle formulation in [12], the question of finding a compact formulation that has the number of variables and constraints bounded by a polynomial on the size of the problem (i.e., on the total number of pairs in a donor-recipient pool), is still open: the cycle formulation presents an exponential number of variables, while the edge formulation has exponential number of constraints.

This paper focuses on mathematical modelling aspects of the KEP: we propose two new compact formulations for the problem. Moreover we investigate the relationships of 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. Finally Section 7 reports the computational analysis and conclusions on the effectiveness of each formulation.

2. Literature review

The concept of kidney exchange program for incompatible patient-donor pairs was first promoted in 1986 in [19] 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 [20, 21]; an overview of contemporary ideas and challenges can be found in [22, 23]. 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 [24, 25, 11] (see Figure 1). The notion can be generalized to a k-*exchange* ($k \geq 3$) in which up to k pairs can be involved in the exchange [5, 12, 8].

Using graph theory the underlying optimization problem can be embedded on a directed graph in which vertices represent non-compatible donor-recipient pairs and arcs between vertices represent compatibilities. The KEP shown in Figure 2 is presented in a directed graph in figure 3.

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

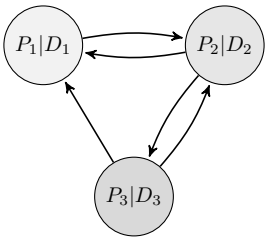


Figure 3: A 3-way exchange on a directed graph.

incompatible pair to the chain [26, 27]. The maximum size of a chain is mandated by national or regional programs. Figure 4 presents a chain of size 2: an altruistic donor donates a kidney to patient of pair 1, P_1, the patient's related donor D_1 donates to P_2, and D_2 donates to the first compatible recipient in the deceased donors waiting list.

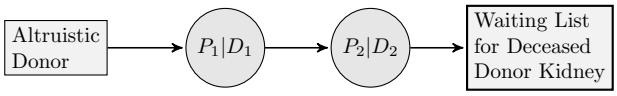


Figure 4: Non-directed donation.

Contrary to the above mentioned problems where simultaneity of exchanges is considered, *Never-Ending Altruistic Donor* (NEAD) chains allow non-simultaneity of exchanges [28, 29, 30, 23]. 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 [31]. In this variant the compatible pair can be involved into an exchange only in case the patient of this pair benefits from being in the pool (e.g. receive a "better" kidney).

Another variant is the *multiple donors* case, when one or more patients have multiple donors associated. In this case if the patient is selected for kidney transplantation the donor that would allow a cycle to be created will be the one selected [12].

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 [32, 33, 15].

2.2. Solution methods

The complexity of k-exchange problem was investigated in [17, 12, 8]. In [17, 12] it is shown that for a given graph G and $k \geq 3$ the problem of deciding if G admits a perfect cycle cover containing cycles of length at most k is NP-complete. The proof uses reduction to 3D–matching problem which is NP-complete. In [8] 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 allows polynomial-time approximation algorithm with approximation ratio bounded by a constant.

As mentioned earlier, the basic variant of the KEP — 2-exchange problem — can be solved in polynomial time using Edmond's algorithm [16] for maximum cardinality matching. This solution approach was followed by [11] and [34]. For unrestricted k, the problem is also solved in polynomial time using a reduction to the maximum-weight perfect matching problem in a bipartite graph [14].

When $k \geq 3$ a natural and perspective way for attacking the KEP is through IP models. Two different integer programs were proposed in [12] and [18]: the *edge* and the *cycle* formulations. In [12] it is presented a sketch of a proof that the cycle formulation provides 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 column generation method with branch-and-bound was designed for the cycle formulation. The cycle formulation is used in [14] 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 of NEAD strategy were presented in [29]. In [27] the authors created a pool of incompatible pairs based on the statistic 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 [30] presented some simulations similar to [27], 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 [28].

The possibility of opening the pool of kidney exchange programs to compatible pairs has been addressed by e.g. [35] and [31]. A mixed pool of compatible and incompatible pairs was simulated in [31], the results showing the benefit of such a policy in terms of increase of 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 [35].

Some results on the dynamic variant of KEP are reported in [32, 33, 15]. The authors in [32] 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 [33], 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 [15] 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 modelled by a set of linear differential equations.

3. Problem definition and formulation

As indicated in Figure 3, 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 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. If the objective is other than maximizing total number of transplants (e.g., maximize weighted exchange) to each arc can be associated a weight w_{ij}, otherwise $w_{ij} = 1$ $ij \in A$.

Figure 5 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 Figure 5 for $k = 3$ the possible cycles are 1-2-3-1 and 3-4-3. However these cycles are not vertex-disjoint for having 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 3 pairs can be involved in an exchange 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: The *Kidney Exchange Problem* can be defined as follows:

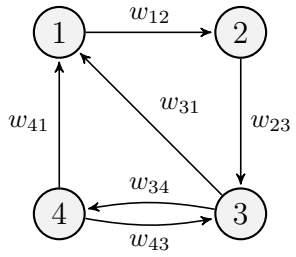


Figure 5: KEP graph.

Find a maximum weight packing 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.

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:

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

The following IP formulation for the KEP can be found in [12] and [18]:

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

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

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

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

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

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 recieve 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 formulation 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 and define a variable z_c for each cycle $c \in \mathcal{C}(k)$:

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

The model can be written as follows (where $w_c = \sum_{(i,j)\in c} w_{ij}$):

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

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

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

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 in this formulation is induced by the exponential number of variables. Indeed, the number of cycles can grow exponentially with k.

4. New compact formulations

The number of constraints or variables in previously proposed formulations 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 [36] 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 on 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

The edge-assignment formulation differs from the edge formulation in section 3.1 by replacing the "path constraints" in equations (1d) by a set of alternative constraints.

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 represented by l, $1 \leq l \leq L$ and 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:

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

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:

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

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

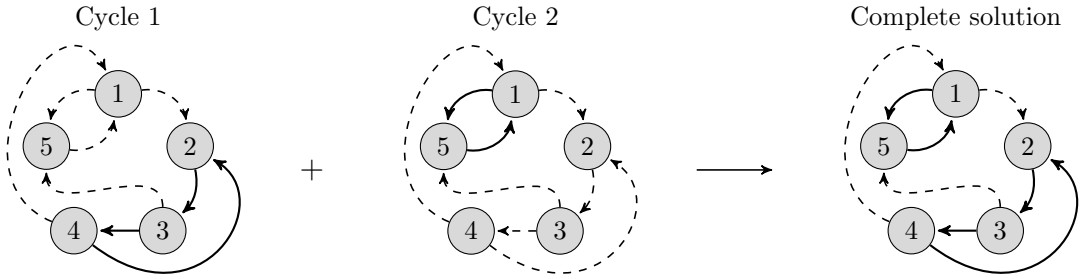
$$y_i^l \in \{0,1\} \qquad \forall i \in V, \quad \forall l \in 1,\ldots,L. \tag{4c}$$

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

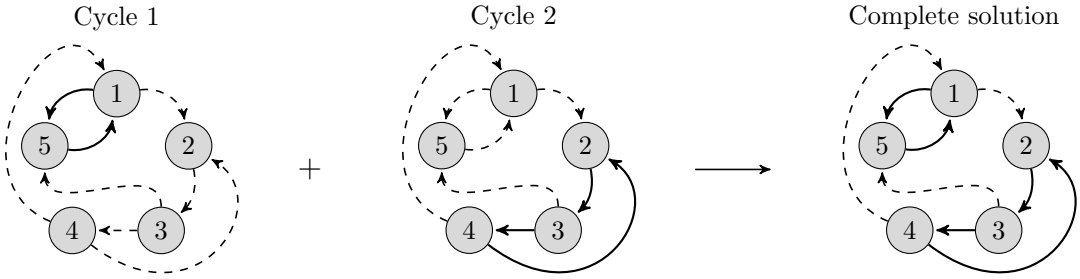
The *edge-assignment* formulation is composed of constraints from the edge formulation — (1a), (1b), (1c) and (1e) — together with the constraints (3a), (4a), (4b) and (4c).

For the formulation given above, in some cycle l, the set of nodes $i \in l$ giving $y_i^l = 1$ can belong to more than one cycle. However this is not a problem because the total number of arcs (or nodes) in a cycle cannot exceed k and cycles are guaranteed to have no more than k arcs.

Furthermore, the edge assignment formulation allows for multiple equivalent solutions. 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 Figure 6. A solution consisting of cycles between pairs 2, 3 and 4 indexed by $l_1 = 1$ and pairs 1 and 5 indexed by $l_2 = 2$ is equivalent to having pairs 1 and 5 indexed by $l_1 = 1$ and pairs 2, 3 and 4 indexed by $l_2 = 2$.



(a) Solution one



(b) Solution two

Figure 6: Multiplicity of equivalent solutions

One way to avoid this multiplicity of solutions 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 $l = \min\{i_1, \ldots, i_r\}$. In the example of Figure 6 only the second solution satisfies this condition, hence the first solution would not be considered. In this case only variables y_i^l with $i \geq l$ are necessary, and this can be enforced by restricting the variables y_i^l to indices $1 \leq l \leq i \leq L$ and by adding constraints:

$$y_i^l \leqslant y_l^l \qquad \forall i, l \in V, i > l \tag{5a}$$

Reduced edge assignment formulation

In some situations the variable y_i^l can be eliminated for tightening the model further if l and i cannot be in the same cycle. Let $\tilde{V}^l = \{i \in V \ : \ i \geq l\}$ and d_{ij}^l denote the shortest path distance in terms of arcs in graph G from i to j for $i, j \in \tilde{V}^l$ such that the path passes only through vertices of set $\tilde{V}^l$. 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 to be considered in the model. More precisely, for each vertex $l \in V$, let us build the set of vertices $V^l = \{i \in V | i \geq l \ \text{ and } \ d_{li}^l + d_{il}^l \leq k\}$. Denote by $\mathcal{L}$ the set of indices l such that $V^l \neq \emptyset$. The *reduced edge assignment* formulation can now be represented by equations (6a) – (6i).

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

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

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

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

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

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

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

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

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

4.2. Extended edge formulation

For 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 edges 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 defined as follows:

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

The extended edge formulation being written as:

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

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

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

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

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

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 less than k.

As with the edge-assignment formulation the extended edge formulation can also generate many equivalent solutions. 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:

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

Reduced extended edge formulation

On the same line of 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 containing both node l and an arc (i, j) with $l \leq i, j$, 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 $A^l = \{(i, j) \in A \mid i, j \in V^l \text{ and } d_{li}^l + 1 + d_{jl}^l \leq k\}$. With this notation one can write the reduced extended edge formulation as follows.

$$\begin{aligned}
\text{maximize} \quad & \sum_{l \in \mathcal{L}} \sum_{(i,j) \in A^l} w_{ij} x_{ij}^l, && && (9a)\\
\text{subject to} \quad & \sum_{j:(j,i) \in A^l} x_{ji}^l = \sum_{j:(i,j) \in A^l} x_{ij}^l && \forall i \in V^l, \forall l \in \mathcal{L}, && (9b)\\
& \sum_{l \in \mathcal{L}} \sum_{j:(i,j) \in A^l} x_{ij}^l \leqslant 1 && \forall i \in \bigcup_{l \in \mathcal{L}} V^l, && (9c)\\
& \sum_{(i,j) \in A^l} x_{ij}^l \leqslant k && \forall l \in \mathcal{L} && (9d)\\
& \sum_{j:(i,j) \in A^l} x_{ij}^l \leqslant \sum_{j:(l,j) \in A^l} x_{lj}^l && \forall i \in V^l, \forall l \in \mathcal{L} && (9e)\\
& x_{ij}^l \in \{0, 1\}. && \forall (i, j) \in A, \forall l \in \mathcal{L}
\end{aligned}$$

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 participation, the problem where compatible pairs are included into the pool, and the problem with one or more patients having multiple donors associated.

Inclusion of altruistic donors

The inclusion of altruistic donors in KEP IP formulations is discussed in [14]. Exchanges involving altruistic donors are labeled as *domino paired chains* (DPC) and the authors do only consider chains of lengths 1 and 2 (when one or two incompatible pairs are involved in the exchange). The idea can however

be generalized to 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 us assume that nodes $\{1, \ldots, m\}$ represent the altruistic donors, nodes $\{m+1, \ldots, m+n\}$ represent the incompatible pairs, and that a dummy patient that is compatible with all donors $j \in \{m+1, \ldots, m+n\}$ is associated to each altruistic donor. An example is given in Figure 7.

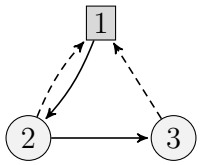


Figure 7: Inclusion of altruistic donor. For this example $m = 1$ and $n = 2$, node 1 is the altruistic donor, dashed arcs $(2, 1)$ and $(3, 1)$ are the arcs to altruistic donor's dummy patient.

Within this description 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. Assume also that $k' \leq k$. 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 the following additional set of constraints must be added:

$$\sum_{1 \leqslant j \leqslant k'} x_{i_j i_{j+1}} \leq k' - 1 \quad \forall \text{paths\_a } (i_1, i_2, \cdots, i_{k'}, i_{k'+1}) \text{ with } i_1 \neq i_{k'+1} \tag{10}$$

where *paths_a* is the set of paths of length k' containing one altruistic donor.

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 equation (3a) limiting the size of cycles is divided in the two following equations:

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

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

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

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

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

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} $\forall i \in V_C$.

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 corresponding patient appears to be created will be selected.

For this problem variant the IP models discussed in this paper 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 for 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 transplant from i to j is assigned to the arc between nodes i and j. The associated IP models will be the same 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 integer program and A and B be two different linear formulations of $\mathcal{IP}$, which define LP upper bounds (for the maximization problem) $UB(A)$ and $UB(B)$. We say that formulation A *dominates* formulation B if $UB(A) \leq UB(B)$. 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 [12].

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) $L = |V|$; ii) the non eliminated variables are x^l_{ij} such that $i \geq l$ or $j \geq l$, and iii) the model contains constraints (8). The following results remain true if the elimination procedures described in 4.2 are applied.

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

Proof. For each l, let X^l be the set of vectors $x^l = (x^l_{ij}, (i, j) \in A)$ defined by constraints (7b), (7d), (7e), (8) and $\sum_{j:(i,j)\in A} x^l_{ij} \leq 1, \forall i \in V$. 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:

$$\text{maximize} \quad \sum_l \sum_{(i,j)\in A} w_{ij} x^l_{ij} \tag{13a}$$

$$\text{subject to} \quad \sum_l \sum_{j:(i,j)\in A} x^l_{ij} \leqslant 1 \qquad \forall i \in V \tag{13b}$$

$$x^l \in X^l \tag{13c}$$

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:

$$\text{maximize} \quad \sum_{l} \sum_{(i,j)\in A} w_{ij} x_{ij}^{l} \tag{14a}$$

$$\text{subject to} \quad \sum_{l} \sum_{j:(i,j)\in A} x_{ij}^{l} \leqslant 1 \qquad \forall i \in V \tag{14b}$$

$$x^{l} \in conv(U^{l}) \qquad \forall l \in \{1, ...L\} \tag{14c}$$

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 value of the LP relaxation of the extended edge model. Indeed, the optimal value of problem (14a)-(14c) is less than or equal to the optimal value of the linear relaxation of the problem (13a)-(13b) with $x^l \in U^l$. But this value is obviously less than or equal to the value of the linear relaxation of problem (13a)-(13c) because $U^l \subseteq X^l$.

One way to write a linear program equivalent to model (14a)-(14c) 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 $\mathcal{C}^l = \{c \in \mathcal{C}(k) : l \in c \subseteq \{l, \ldots, n\}\}$. Now observe that $\{\mathcal{C}^l, l \in 1, \ldots, L\}$ 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 $\sum_{c\in\mathcal{C}^l} u_c \leq 1$. So $x^l \in conv(U^l)$ for each $l \in \{1, \ldots, L\}$ can be rewritten as:

$$x_{ij}^{l} = \sum_{c\in\mathcal{C}^l:(i,j)\in c} u_c \qquad \forall (i,j) \in A \tag{15a}$$

$$\sum_{c\in\mathcal{C}^l} u_c \leq 1 \tag{15b}$$

$$u_c \geq 0 \qquad \forall c \in \mathcal{C}^l \tag{15c}$$

Let $w_c = \sum_{(i,j)\in c} w_{ij}$ for $c \in \mathcal{C}(k)$. Using (15a), (15b) and (15c), the model (14a)-(14c) can be written as follows.

$$\text{maximize} \quad \sum_{c\in\mathcal{C}(k)} w_c u_c \tag{16a}$$

$$\text{subject to} \quad \sum_{c\in\mathcal{C}(k)\,:\,i\in c} u_c \leqslant 1 \qquad \forall i \in V \tag{16b}$$

$$\sum_{c\in\mathcal{C}^l} u_c \leq 1 \qquad \forall l \in \{1, ...L\} \tag{16c}$$

$$u_c \geq 0 \qquad \forall c \in \mathcal{C}^l, l \in \{1, ...L\} \tag{16d}$$

It is straightforward to see that constraints (16c) are always satisfied with respect to constraints (16b). Thus problem (16a)-(16d) 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: Let $\bar{x}_{ij}^l, (i,j) \in A, l \in \Lambda = \{1, \ldots L\}$ be an optimal solution of the linear relaxation of the extended edge model (7a) – (7e). We build a feasible solution for the LP relaxation of the edge-assignment model with same objective value.

Define variables $\bar{x}_{ij}$ and $\bar{y}_i^l$ as:

$$\bar{x}_{ij} = \sum_{l \in \Lambda} \bar{x}^l_{ij} \quad \forall (i,j) \in A \tag{17}$$

$$\bar{y}^l_i = \sum_{j:(i,j)\in A} \bar{x}^l_{ij} \quad \forall i \in V, i \geq l \tag{18}$$

The verification that the objective value is the same for both solutions uses (17) and it is straightforward. Also, constraints (1b) and (1c) follow directly from (7b) and (7c) respectively, and (17). Similarly, (3a) follows from (7d) and (18), and (4a) is obtained from summing both sides of (18) over l and both sides of (17) over j. We show next that (4b) is satisfied.

By (17) and (18) $\bar{y}^l_i + \bar{x}_{ij} = \sum_{p:(i,p)\in A} \bar{x}^l_{ip} + \sum_{l\in\Lambda} \bar{x}^l_{ij}$. Now observe that the only common variable in the two previous sums is $\bar{x}^l_{ij}$. Hence $\bar{y}^l_i + \bar{x}_{ij} \leq \sum_{l\in\Lambda} \sum_{p:(i,p)\in A} \bar{x}^l_{ip} + \bar{x}^l_{ij} \leq 1 + \bar{y}^l_i$, the last inequality following from (7c) and from (18) and the nonnegativity of the variables.

Finally, $0 \leq \bar{y}^l_i \leq 1$ and $0 \leq \bar{x}_{ij} \leq 1$ are a consequence of nonegativity 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 optimal solution for extended edge formulation. Hence we conclude that the value of the LP relaxation of the edge-assignment formulation is greater than or equal to 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: 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 $A = \{(1,2),(1,3),(1,4),(2,3),(3,4),(4,5),(5,1)\}$ – Figure 8. The optimal value for this instance is three, given by cycle 1-4-5-1.

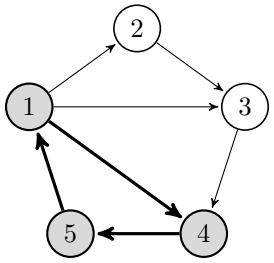


Figure 8: Example 1: an optimal solution for $k = 3$ is the cycle $1 - 4 - 5 - 1$.

Observe that all cycles contain vertex 1. Hence in the extended edge formulation $x^l_{ij} = 0$ for all (i,j) and $l \geq 2$, and the optimal LP value is 3 (although there are several optimal non integer solutions). The optimal LP solution for the edge formulation is $x_{12} = x_{23} = x_{34} = x_{45} = x_{56} = 0.66(7)$ and $x_{13} = x_{14} = 0$, with value 3.33(3). □

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

Proof: Consider the example illustrated by Figure 9, where $k = 3$, the set of nodes is$V = \{1,2,3,4\}$ and the set of arcs is $A = \{(1,2),(1,3),(2,3),(3,1),(3,4),(4,1),(4,2)\}$. The optimal value for this instance is 3.

The optimal LP solution for the edge formulation is $x_{12} = x_{23} = x_{34} = x_{41} = 0.66(7)$ and $x_{13} = x_{31} = 0.33(3)$, with value 3.33(3). An optimal LP solution for the extended edge formulation is $x^1_{12} = x^1_{23} = x^1_{34} =$

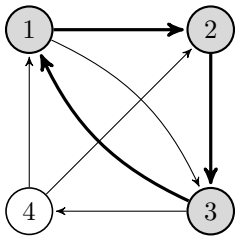


Figure 9: Example 2: an optimal solution for $k = 3$ is the cycle $1 - 2 - 3 - 1$.

$x^1_{41} = 0.75$, $x^2_{23} = x^2_{34} = x^2_{42} = 0.25$ 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 but we strongly believe in this conjecture.

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 where 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 GHz, 16 Gb of RAM and running Mac OS X 10.6.6. Only one thread was assigned to these experiments.

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

1) the instance generator described in [10], 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. The performance of all formulations was tested for all instances. Besides that for this set of problems we compared the average number of constraints and variables for different models and the percent of 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 general better performance on "small" instances were selected for testing on larger problems ($n > 50$). Ten instances were generated for different large problem 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 2-6 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. Time needed to carry out reduction procedures for the EA and the EE formulations was less than 1 second for all small instances and less than 3 seconds for all large problems, and therefore it is not shown in the tables;

- T is the average CPU time the solver CPLEX [37] 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 1800.0 seconds for all formulations;

- $[min, max]$ are the minimum and maximum CPU times taken to reach optimal solutions for a given set (including preprocessing times t_c and t_e for the cycle and edge formulations);

- $\#opt$ is the number of instances from each set which were solved to optimality within the time limit (1800 seconds);

- gap is the average LP gap associated to a formulation: $gap = \frac{UB-Opt}{Opt} * 100\%$, where UB is the upper bound provided by the linear relaxation of the formulation and Opt is the optimal value of the problem.

Values 0.00 in tables related to CPU time mean that the solver took less than 0.1 second 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 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 and for the second one for the three graph densities. Fifty instances of each size were considered. Table 1 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. Other computational results are presented in tables 2–5.

Blood-type test instances

The results for these instances, presented in Table 2, 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. However a shorter time that is required by the EE formulation to reach optimality for larger test instances with larger values of k together with small gap values is promising for handling larger scale instances.

In terms of size reduction, the introduction of ordering constraints and elimination procedures for EA and EE resulted in significantly fewer variables and constraints. The number of constraints of the EE formulation was reduced on average 65%. For $k = 3$ reduction was on average 80% and 56% for $k = 6$. The number of variables for all instances was reduced by 91% on average. No results are provided for these formulations before reduction as they were clearly worse.

Low density test instances

Results for randomly generated graphs with "low" density are presented in Table 3. Although the density of these graphs 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 3); in several cases the E formulation

also was not 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 over the others, both formulations having solved to optimality all instances.

The introduction of ordering constraints and elimination procedures for EA and EE resulted in a reduction of the number of constraints on average 48% for the EE formulation(from 76% for $k = 3$ to 30% for $k = 6$). The number of variables was decreased on average by 86% for all instances of this group (Table 1).

Medium density test instances

The importance of developing compact formulations is reflected in the results obtained for "medium" density test instances (see Table 4). In this case the EE formulation proved to be extremely efficient at solving larger problems. 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.

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.

The reduction on the number of variables for the reduced EE was of 74% on average, while the reduction on the number of constraints was now less significant: 11% on average.

High density test instances

Results for "high" density test instances are displayed in Table 5, corroborating and strengthening the conclusions drawn with the previous results on the advantages of compact formulations. For this set of instances only the EE was capable of solving to optimality the complete set. The EA, although performing worse than the EE, still solved to optimality almost 95% of the instances considered.

The problems raised up by non-compact formulations, related to the exponential numbers of variables associated to the C formulation and of constraints associated to the E formulation, become more evident with these results: 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 $k = 5, 6$.

LP gaps for all the formulations are equal to 0% for all tests and all values of k.

Here the reductions on the EE was only of 4% for the number of constraints and of 70% for the number of variables (Table 1).

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 of 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 which is optimal for the integer program. When k increases the problems become closer to being uncapacitated so the LP solutions are closer to be integer and the LP values 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 6); 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 wills 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 nodes. 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, and Prof. Filipe Alvelos from University of Minho, Braga, Protugal, for their useful comments.

References

[1] J. Kwak, O. Kwon, K. L. KS, C. Kang, H. Park, J. Kim, Transplant Proc 31 (1999) 344–345.
[2] G. Thiel, P. V. L. Gurke, T. Gasser, K. Lehmann, T. Voegele, A. Kiss, G. Kirste, Transplant Proc 33 (2001) 811–816.
[3] A. Gurkan, S. Kacar, C. Varilsuha, S. Tilif, I. Coker, C. Karaca, M. Karaoglan, Transplantation Proceedings 36 (2004) 2952–2953.
[4] M. Lucan, Transplantation Proceedings 39 (2007) 1371–1375.
[5] M. de Klerk, K. Keizer, F. Claas, B. Haase-Kromwijk, W. Weimar, American Journal of Transplantation 5 (2005) 2302–2305.
[6] M. de Klerk, M. Witvliet, B. Haase-Kromwijk, F. Claas, W. Weimar, Transplantation 82 (2006) 1616–1620.
[7] L. Kranenburg, W. Zuidema, W. Weimar, M. Hilhorst, J. IJzermans, J. Passchier, J. Busschbach, Progress in Transplantation 19 (2009) 71–75.
[8] P. Biro, D. Manlove, R. Rizzi, Technical report TR-2009-298. Department of computing science, University of Glasgow (2009).
[9] R. Johnson, J. Allen, S. Fuggle, J. Bradley, C. Rudge, Transplantation 86 (2008) 1672–1677.
[10] S. Saidman, A. Roth, T. Sönmez, M. Ünver, F. Delmonico, Transplantation 81 (2006) 773–782.

[11] D. Segev, S. Gentry, D. Warren, B. Reeb, R. Montgomery, The Journal of the American Medical Association 293 (2005) 1883–1890.
[12] D. Abraham, A. Blum, T. Sandholm, Proceedings of the 8th ACM conference on Electronic commerce, June 13-16,h 2007 (2007) 295–304.
[13] J. Veale, G. Hil, Clinical Transplants (2010) 333–344.
[14] D. Manlove, G. O'Malley, Paired and altruistic kidney donation in the UK: Algorithms and experimentation, To appear in Proceedings of SEA 2012: the 11th International Symposium on Experimental Algorithms. Springer: Lecture Notes in Computer Science, 2012.
[15] S. Zenios, G. Chertow, L. Wein, Operations Research 48 (2000) 549–569.
[16] J. Edmonds, Canadian Journal of Mathematics 17 (1965) 449–467.
[17] D. Abraham, Matching markets: Design and analysis, PhD Thesis. Carnegie-Mellon University, School of Computer Science, 2009.
[18] A. Roth, T. Sönmez, M. Ünver, The American Economic Review 97 (2007) 828–851.
[19] F. Rapaport, Transplant Proc 18 (1986) 5–9.
[20] L. Ross, E. Woodle, Transplantation 69 (2000) 1539–1543.
[21] L. Ross, S. Zenios, American Journal of Transplantation 4 (2004) 1553–1554.
[22] C. Wallis, K. Samy, A. Roth, M. Rees, Nephrol Dial Transplant 26 (2011) 2091–2099.
[23] S. Gentry, R. Montgomery, D. Segev, American journal of kidney disease 57 (2010) 144–151.
[24] A. Mahendran, P. Veitch, British Journal of Surgery 94 (2007) 657–664.
[25] I. Kaplan, J. Houp, M. Leffell, J. Hart, A. Zachary, American Journal of Transplantation 5 (2005) 2306–2308.
[26] R. Montgomery, S. Gentry, W. Marks, D. Warren, J. Hiller, J. Houp, A. Zachary, J. Melancon, W. Maley, H. R. C. Simpkins, D. Segev, The Lancet 368 (2006) 419–421.
[27] S. Gentry, R. Montgomery, B. Swihart, D. Segev, American Journal of Transplantation 9 (2009) 1330–1336.
[28] J. Dickerson, A. Procaccia, T. Sandholm, In AAMAS-12: Proc. 11th Intl. Joint Conference on Autonomous Agents and Multiagent Systems, Jun 2011 (2011).
[29] 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, The New England Journal of Medicine 360 (2009) 1096–1101.
[30] I. Ashlagi, D. Gilchrist, A. Roth, M. Rees, American Journal of Transplantation 11 (2011) 984–994.
[31] S. Gentry, D. Segev, M. Simmerling, R. Montgomery, American Journal of Transplantation 7 (2007) 2361–2370.
[32] M. Ünver, Review of Economic Studies 77 (2010) 372–414.
[33] P. Awasthi, T. Sandholm, Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09) (2009) 405–411.
[34] A. Roth, T. Sönmez, M. Ünver, Practical market design 95 (2005) 376–380.
[35] L. Ross, Kennedy Institute of Ethics Journal 16 (2006) 151–172.
[36] G. Nemhauser, L. Wolsey, Integer and Combinatiorial Optimization, A Wiley-Interscience Publication, 1999.
[37] Ibm ilog cplex optimizer, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, 2012.

		C		E		EA				EE			
n	k	$\#var$	$\#con$	$\#var$	$\#con$	$\#var$	$\#con$	$rv\%$	$rc\%$	$\#var$	$\#con$	$rv\%$	$rc\%$
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
20	3	28	15	77	1032	120	409	58.5	70.1	60	103	96.1	76.6
	4	80	19	77	3297	157	646	45.6	52.7	142	179	90.9	59.4
	5	217	19	77	10011	188	802	34.6	41.2	252	242	83.9	44.9
	6	563	19	77	28381	207	880	28.0	35.4	347	280	77.8	36.4
30	3	82	28	174	5215	280	1318	56.2	68.5	169	242	96.8	74.8
	4	336	30	174	26634	401	2354	37.4	43.6	498	484	90.5	49.6
	5	1377	30	174	131770	496	2978	22.5	28.5	1008	674	80.8	29.8
	6	5681	30	174	626695	539	3225	15.6	22.4	1412	761	73.0	20.7
40	3	187	40	311	17499	522	3264	53.9	65.5	369	461	97.0	72.6
	4	1052	40	311	125158	790	6144	30.2	35.1	1276	998	89.8	40.6
	5	6047	40	-	$> 3 * 10^6$	966	7585	14.6	19.8	2779	1350	77.7	19.6
	6	35052	40	-	$> 3 * 10^6$	1029	8109	9.1	14.2	3719	1475	70.2	12.2
50	3	363	50	491	44579	851	6782	51.8	62.5	684	770	97.2	70.4
	4	2596	50	-	$> 3 * 10^6$	1342	13059	24.0	27.8	2672	1752	89.1	32.6
	5	19010	50	-	$> 3 * 10^6$	1601	15628	9.3	13.6	6140	2271	75.0	12.7
	6	142190	50	-	$> 3 * 10^6$	1665	16332	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	23732	114	507	2.7	7.0	228	114	63.1	5.1
	6	4075	10	62	65993	114	507	2.7	7.0	229	114	63.0	5.1
20	3	875	20	266	39949	456	3771	4.2	6.1	1047	400	80.4	9.0
	4	7851	20	266	446931	473	3926	0.7	2.2	1724	433	67.6	1.5
	5	70343	20	-	$> 3 * 10^6$	473	3927	0.7	2.2	1871	434	64.8	1.5
	6	616924	20	-	$> 3 * 10^6$	473	3928	0.7	2.2	1872	434	64.8	1.5
30	3	2949	30	606	221941	1026	12290	4.2	6.0	3341	870	81.6	9.3
	4	41556	30	-	$> 3 * 10^6$	1066	12887	0.4	1.4	5731	950	68.5	1.0
	5	600569	30	-	$> 3 * 10^6$	1066	12896	0.4	1.4	6251	951	65.6	0.9
	6	$> 3 * 10^6$	-	-	$> 3 * 10^6$	1066	12896	0.4	1.4	6252	951	65.6	0.9
40	3	7164	40	1093	753123	1839	29130	3.9	5.4	7872	1532	82.0	8.8
	4	138934	40	-	$> 3 * 10^6$	1908	30556	0.2	0.8	13683	1671	68.7	0.5
	5	2239636	33	-	$> 3 * 10^6$	1908	30561	0.2	0.7	14923	1672	65.9	0.5
	6	$> 3 * 10^6$	-	-	$> 3 * 10^6$	1908	30561	0.2	0.7	14924	1672	65.9	0.5
50	3	14157	50	1719	1911261	2883	56974	3.7	5.0	15272	2377	82.2	8.6
	4	$> 3 * 10^6$	-	-	$> 3 * 10^6$	2992	59750	0.1	0.3	26817	2595	68.8	0.2
	5	$> 3 * 10^6$	-	-	$> 3 * 10^6$	2992	59762	0.1	0.3	29272	2595	66.0	0.2
	6	$> 3 * 10^6$	-	-	$> 3 * 10^6$	2992	59764	0.1	0.3	29273	2595	65.9	0.2

Table 1: Sizes of formulations and reduction for EA and EE. "Low" and "high" density test instances. 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.

		C					E					EA				EE			
n	k	t_c	T	$[min, max]$	$\#opt$	gap	t_p	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap
10	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.0]	50	0.5	0.0	[0.0, 0.0]	50	1.0	0.0	[0.0, 0.0]	50	0.0
	4	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	5	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	6	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
20	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.6	[0.0, 26.9]	50	5.6	0.3	[0.0, 11.6]	50	6.5	0.0	[0.0, 0.3]	50	2.8
	4	0.0	0.0	[0.0, 0.0]	50	0.0	0.1	0.1	[0.0, 1.2]	50	1.5	0.1	[0.0, 0.7]	50	1.9	0.0	[0.0, 0.1]	50	1.1
	5	0.0	0.0	[0.0, 0.2]	50	0.1	0.8	0.4	[0.5, 8.9]	50	0.7	0.0	[0.0, 0.2]	50	0.9	0.0	[0.0, 0.1]	50	0.8
	6	0.1	0.0	[0.1, 0.4]	50	0.0	8.5	0.5	[6.9, 23.7]	47(47)	0.2	0.0	[0.0, 0.2]	50	0.2	0.0	[0.0, 0.1]	50	0.2
30	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.1	0.3	[0.0, 4.4]	50	4.0	5.8	[0.0, 160.0]	50	4.9	0.0	[0.0, 0.2]	50	1.2
	4	0.0	0.0	[0.0, 0.1]	50	0.0	0.7	1.0	[0.4, 7.5]	50	0.6	0.5	[0.0, 7.0]	50	0.7	0.1	[0.0, 0.3]	50	0.5
	5	0.1	0.0	[0.1, 0.2]	50	0.0	9.8	2.2	[7.8, 30.6]	42(42)	0.0	0.3	[0.0, 3.0]	50	0.0	0.1	[0.0, 0.2]	50	0.0
	6	1.7	0.1	[1.5, 3.4]	50	0.0	184.5	0.3	[180.2, 189.8]	12(12)	0.0	0.2	[0.0, 2.6]	50	0.0	0.1	[0.0, 0.2]	50	0.0
40	3	0.0	0.0	[0.0, 0.0]	50	0.1	0.1	48.0	[0.1, 870.8]	50	5.6	85.1	[0.1, 1228.5]	45	6.1	0.1	[0.0, 0.7]	50	2.7
	4	0.0	0.1	[0.0, 0.2]	50	0.0	2.9	27.9	[1.7, 303.2]	50	0.9	48.8	[0.1, 1129.8]	50	1.0	0.3	[0.0, 1.6]	50	0.9
	5	0.5	0.1	[0.5, 1.0]	50	0.0	54.0	3.5	[51.5, 68.5]	18(18)	1.5	2.4	[0.1, 28.9]	50	0.6	0.3	[0.0, 1.0]	50	0.6
	6	10.7	0.5	[9.7, 15.4]	50	0.0	1726.8	0.8	[1664.2, 1897.9]	4(4)	2.4	3.0	[0.1, 71.0]	50	0.2	0.3	[0.0, 1.1]	50	0.2
50	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.4	133.8	[0.2, 1697.6]	49(50)	3.6	111.5	[0.3, 1689.0]	30	3.9	0.4	[0.0, 3.0]	50	2.0
	4	0.1	0.0	[0.1, 0.3]	50	0.0	12.3	108.5	[5.4, 1369.4]	47(50)	0.6	42.3	[0.2, 331.3]	44	0.7	1.2	[0.0, 9.5]	50	0.3
	5	1.6	0.4	[1.5, 7.5]	50	0.1	213.1	6.7	[210.6, 225.2]	3(3)	0.0	13.9	[0.1, 273.4]	48	0.4	0.9	[0.0, 3.1]	50	0.2
	6	44.8	6.0	[40.9, 130.1]	50	0.0	-	-	-	0	-	7.8	[0.2, 224.3]	50	0.1	0.9	[0.0, 3.2]	50	0.0

Table 2: "Blood-type" test instances.

		C					E					EA				EE			
n	k	t_c	T	$[min, max]$	$\#opt$	gap	t_p	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap
10	3	0.0	0.0	[0.0, 0.0]	50	0.2	0.0	0.0	[0.0, 0.0]	50	21.8	0.0	[0.0, 0.0]	50	33.2	0.0	[0.0, 0.0]	50	0.3
	4	0.0	0.0	[0.0, 0.0]	50	0.5	0.0	0.0	[0.0, 0.0]	50	11.6	0.0	[0.0, 0.0]	50	15.2	0.0	[0.0, 0.0]	50	0.9
	5	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.0]	50	7.0	0.0	[0.0, 0.0]	50	8.2	0.0	[0.0, 0.0]	50	0.2
	6	0.0	0.0	[0.0, 0.0]	50	0.2	0.0	0.0	[0.0, 0.0]	50	1.1	0.0	[0.0, 0.0]	50	1.7	0.0	[0.0, 0.0]	50	1.1
20	3	0.0	0.0	[0.0, 0.0]	50	1.4	0.0	0.2	[0.0, 1.5]	50	21.5	1.4	[0.0, 19.8]	50	23.7	0.0	[0.0, 0.0]	50	2.6
	4	0.0	0.0	[0.0, 0.0]	50	0.4	0.0	0.4	[0.1, 2.0]	50	5.7	1.2	[0.0, 6.4]	50	6.2	0.0	[0.0, 0.1]	50	1.5
	5	0.0	0.0	[0.0, 0.0]	50	0.6	0.5	0.8	[0.5, 4.8]	50	1.9	0.6	[0.0, 3.3]	50	1.9	0.1	[0.0, 0.5]	50	1.2
	6	0.1	0.0	[0.1, 0.2]	50	0.4	7.2	1.1	[6.8, 11.3]	50	0.8	0.2	[0.0, 1.0]	50	0.8	0.1	[0.0, 0.3]	50	0.7
30	3	0.0	0.0	[0.0, 0.0]	50	0.5	0.0	43.5	[3.5, 269.2]	50	4.9	713.5	[18.0, 1764.5]	33	4.9	0.0	[0.0, 0.1]	50	1.1
	4	0.0	0.0	[0.0, 0.0]	50	0.1	0.4	33.2	[2.1, 162.2]	50	0.2	348.3	[24.0, 1539.8]	42	0.2	0.5	[0.0, 2.6]	50	0.1
	5	0.1	0.1	[0.1, 0.2]	50	0.0	8.1	64.7	[9.8, 255.5]	50	0.0	85.7	[0.8, 897.6]	50	0.0	1.0	[0.2, 3.2]	50	0.0
	6	1.6	0.3	[1.7, 2.5]	50	0.0	188.0	225.6	[193.9, 1132.5]	50	0.0	17.0	[0.8, 171.9]	50	0.0	0.6	[0.2, 3.0]	50	0.0
40	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.1	511.9	[39.5, 1370.1]	20(50)	0.1	-	-	0	0.5	0.1	[0.0, 0.2]	50	0.1
	4	0.0	0.1	[0.0, 0.2]	50	0.0	1.8	571.3	[61.0, 1763.0]	30(50)	0.0	850.8	[164.2, 1665.3]	9	0.0	3.7	[0.4, 14.4]	50	0.0
	5	0.5	0.5	[0.7, 1.6]	50	0.0	-	-	-	0	-	453.1	[5.2, 1674.1]	30	0.0	5.0	[0.6, 38.4]	50	0.0
	6	10.4	1.9	[10.0, 16.1]	50	0.0	-	-	-	0	-	246.4	[8.8, 1773.5]	45	0.0	3.2	[0.5, 15.3]	50	0.0
50	3	0.0	0.0	[0.0, 0.1]	50	0.0	0.2	505.5	[253.7, 988.2]	3(50)	0.0	-	-	0	0.0	0.4	[0.1, 2.3]	50	0.0
	4	0.1	0.1	[0.1, 0.4]	50	0.0	-	-	-	0	-	-	-	0	0.0	20.2	[1.9, 321.2]	50	0.0
	5	1.6	0.9	[1.8, 4.8]	50	0.0	-	-	-	0	-	-	-	0	0.0	14.8	[2.1, 43.1]	50	0.0
	6	41.8	10.8	[43.8, 92.2]	50	0.0	-	-	-	0	-	618.1	[65.6, 1586.3]	20	0.0	7.0	[1.1, 28.5]	50	0.0

Table 3: "Low" density instances.

		C					E					EA				EE				
n	k	t_c	T	$[min, max]$	$\#opt$	gap	t_p	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap
10	3	0.0	0.0	[0.0, 0.0]	50	0.6	0.0	0.0	[0.0, 0.1]	50	0.9	0.0	[0.0, 0.2]	50	1.0	0.0	[0.0, 0.0]	50	0.7
	4	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.1]	50	0.2	0.0	[0.0, 0.1]	50	0.2	0.0	[0.0, 0.0]	50	0.0
	5	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.2]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	6	0.0	0.0	[0.0, 0.1]	50	0.0	0.0	0.1	[0.0, 0.3]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
20	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.9	[0.1, 2.7]	50	0.0	2.4	[0.2, 15.8]	50	0.0	0.1	[0.0, 0.3]	50	0.0
	4	0.0	0.0	[0.0, 0.1]	50	0.0	0.2	5.4	[0.8, 19.0]	50	0.0	1.2	[0.2, 4.9]	50	0.0	0.1	[0.1, 0.5]	50	0.0
	5	0.0	0.2	[0.1, 0.9]	50	0.0	2.3	54.8	[7.5, 288.2]	50	0.0	0.5	[0.1, 2.7]	50	0.0	0.1	[0.1, 0.4]	50	0.0
	6	0.3	2.9	[0.5, 11.2]	50	0.0	-	-	-	50	-	0.3	[0.1, 1.9]	50	0.0	0.1	[0.0, 0.5]	50	0.0
30	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.1	15.3	[1.3, 78.8]	50	0.0	162.0	[1.8, 1016.0]	47	0.0	0.4	[0.2, 2.1]	50	0.0
	4	0.0	0.3	[0.1, 0.6]	50	0.0	2.3	138.4	[28.9, 614.3]	49(50)	0.0	30.0	[1.0, 256.5]	50	0.0	0.7	[0.1, 3.8]	50	0.0
	5	0.3	5.0	[0.7, 21.5]	50	0.0	-	-	-	0	-	3.8	[0.6, 23.2]	50	0.0	0.4	[0.2, 2.0]	50	0.0
	6	3.8	267.0	[10.1, 1609.0]	49(50)	0.0	-	-	-	0	-	1.7	[0.2, 16.3]	50	0.0	0.3	[0.1, 0.9]	50	0.0
40	3	0.0	0.0	[0.0, 0.1]	50	0.0	0.5	171.1	[3.0, 1370.9]	50	0.0	563.8	[9.2, 1773.0]	24	0.0	1.3	[0.1, 8.0]	50	0.0
	4	0.1	0.8	[0.3, 3.4]	50	0.0	6.1	624.4	[127.7, 1705.9]	39(50)	0.0	203.2	[2.5, 1152.7]	40	0.0	2.0	[0.1, 15.8]	50	0.0
	5	1.2	66.8	[5.5, 517.8]	50	0.0	-	-	-	0	-	105.8	[2.5, 1470.9]	49	0.0	1.5	[0.1, 8.7]	50	0.0
	6	-	-	-	50	-	-	-	-	0	-	16.3	[1.5, 183.0]	50	0.0	1.2	[0.5, 7.7]	50	0.0
50	3	0.0	0.1	[0.0, 0.2]	50	0.0	1.1	440.3	[80.6, 1597.5]	47(50)	0.0	744.9	[31.2, 1749.7]	3	0.0	4.3	[1.0, 25.5]	50	0.0
	4	0.2	4.3	[0.6, 18.5]	50	0.0	-	-	-	0	-	644.9	[16.7, 1721.1]	18	0.0	5.4	[2.7, 13.1]	50	0.0
	5	3.8	385.4	[15.8, 1657.6]	48(50)	0.0	-	-	-	0	-	116.1	[4.7, 1019.4]	34	0.0	6.0	[1.3, 47.7]	50	0.0
	6	-	-	-	0	-	-	-	-	0	-	61.7	[4.0, 605.3]	46	0.0	2.7	[1.1, 6.8]	50	0.0

Table 4: "Medium" density instances.

		C					E					EA				EE			
n	k	t_c	T	$[min, max]$	$\#opt$	gap	t_p	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap	T	$[min, max]$	$\#opt$	gap
10	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.0	[0.0, 0.1]	50	0.0	0.0	[0.0, 0.1]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	4	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	0.1	[0.0, 0.3]	50	0.0	0.0	[0.0, 0.1]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	5	0.0	0.0	[0.0, 0.1]	50	0.0	0.0	0.2	[0.0, 0.8]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
	6	0.0	0.0	[0.0, 0.1]	50	0.0	0.1	0.5	[0.1, 2.1]	50	0.0	0.0	[0.0, 0.0]	50	0.0	0.0	[0.0, 0.0]	50	0.0
20	3	0.0	0.0	[0.0, 0.0]	50	0.0	0.1	1.0	[0.2, 5.5]	50	0.0	2.2	[0.2, 33.3]	50	0.0	0.1	[0.0, 0.2]	50	0.0
	4	0.0	0.2	[0.0, 0.4]	50	0.0	0.7	15.7	[2.7, 83.3]	50	0.0	0.5	[0.0, 4.4]	50	0.0	0.1	[0.0, 0.2]	50	0.0
	5	0.1	1.8	[0.3, 6.3]	50	0.0	-	-	-	0	-	0.3	[0.0, 1.2]	50	0.0	0.1	[0.0, 0.2]	50	0.0
	6	0.9	52.0	[3.3, 398.6]	50	0.0	-	-	-	0	-	0.2	[0.0, 1.1]	50	0.0	0.1	[0.0, 0.2]	50	0.0
30	3	0.0	0.0	[0.0, 0.1]	50	0.0	0.3	8.5	[1.4, 21.5]	50	0.0	32.3	[1.0, 274.3]	50	0.0	0.5	[0.2, 2.9]	50	0.0
	4	0.0	0.5	[0.2, 2.5]	50	0.0	-	-	-	0	-	4.5	[0.7, 25.0]	50	0.0	0.5	[0.0, 3.0]	50	0.0
	5	0.8	65.8	[3.3, 592.8]	50	0.0	-	-	-	0	-	1.9	[0.5, 14.4]	50	0.0	0.4	[0.1, 0.7]	50	0.0
	6	-	-	-	0	-	-	-	-	0	-	1.0	[0.2, 2.4]	50	0.0	0.4	[0.0, 1.5]	50	0.0
40	3	0.0	0.1	[0.0, 0.2]	50	0.0	1.1	64.3	[5.2, 339.5]	50	0.0	237.8	[5.8, 1395.8]	36	0.0	2.2	[0.1, 8.8]	50	0.0
	4	0.2	4.9	[0.7, 31.0]	50	0.0	-	-	-	0	-	37.8	[2.4, 1310.8]	49	0.0	2.3	[0.1, 14.6]	50	0.0
	5	3.8	65.6	[20.9, 143.2]	34(41)	0.0	-	-	-	0	-	11.6	[2.0, 324.0]	50	0.0	1.1	[0.1, 1.9]	50	0.0
	6	-	-	-	0	-	-	-	-	0	-	3.9	[1.7, 10.0]	50	0.0	1.4	[0.2, 9.9]	50	0.0
50	3	0.0	0.4	[0.1, 0.6]	50	0.0	2.5	237.8	[13.7, 816.7]	48(50)	0.0	250.4	[15.8, 1746.7]	15	0.0	9.8	[3.3, 33.7]	50	0.0
	4	-	-	-	0	0.0	-	-	-	0	-	130.2	[6.8, 1042.5]	39	0.0	6.7	[0.6, 25.3]	50	0.0
	5	-	-	-	0	-	-	-	-	0	-	25.3	[5.8, 581.9]	50	0.0	4.7	[0.5, 21.3]	50	0.0
	6	-	-	-	0	-	-	-	-	0	-	46.8	[5.4, 1062.7]	50	0.0	2.3	[0.5, 6.2]	50	0.0

Table 5: "High" density instances.

		C					EE			
n	k	t_c	T	$(min-max)$	$\#opt$	gap	T	$(min-max)$	$\#opt$	gap
Blood-type test instances										
70	3	0.0	0.0	[0.0, 0.0]	10	0.0	1.6	[0.2, 11.0]	10	1.8
100		0.1	0.0	[0.1, 0.1]	10	0.1	2.9	[0.8, 6.0]	10	0.8
200		0.8	0.3	[0.8, 1.6]	10	0.0	1220.5	[45.3, 1800.4]	4	0.4
300		3.3	1.5	[3.8, 5.6]	10	0.0	-	-	0	-
500		23.1	10.5	[30.5, 43.8]	10	0.0	-	-	0	-
800		141.4	147.9	[172.8, 697.8]	10	0.0	-	-	0	-
900		223.4	397.8	[297.6, 999.4]	9(9)	0.0	-	-	0	-
1000		343.1	478.6	[525.2, 1002.5]	7(7)	0.0	-	-	0	-
70	4	0.4	0.1	[0.4, 0.7]	10	0.0	10.8	[1.0, 48.8]	10	0.5
100		1.9	0.4	[1.9, 3.0]	10	0.0	52.7	[5.8, 231.7]	10	0.0
200		41.7	25.0	[41.9, 108.5]	10	0.0	-	-	0	0.0
70	5	10.2	2.1	[9.9, 14.6]	10	0.0	7	[1.6, 28.9]	10	0.0
100		66.8	13.2	[71.3, 102.7]	10	0.0	71.6	[10.9, 322.1]	10	0.0
70	6	369.5	25.0	[359.8, 445.6]	9(9)	0.0	3.9	[1.5, 11.3]	10	0.0
100		3254.5	19.4	[3273.9, 3273.9]	1(1)	0.0	101.9	[10.5, 727.6]	10	0.0
Low density test instances										
70	3	0.0	0.1	[0.0, 0.1]	10	0.0	5.1	[0.6, 18.7]	10	0.0
100		0.1	0.2	[0.1, 1.0]	10	0.0	51.7	[10.6, 153.4]	10	0.0
300		3.4	11.0	[6.1, 87.2]	10	0.0	-	-	0	-
500		25.6	604.5	[98.6, 1745.5]	7(7)	0.0	-	-	0	-
70	4	0.4	0.8	[0.6, 3.2]	10	0.0	384.9	[36.9, 1492.9]	10	0.0
100		2.0	4.6	[2.8, 15.7]	10	0.0	-	-	0	-
200		44.1	341.0	[133.4, 838.0]	8(8)	0.0	-	-	0	-
70	5	9.6	9.9	[12.6, 46.4]	10	0.0	371.4	[19.6, 1616.7]	10	0.0
100		64.0	225.5	[110.0, 562.4]	10	0.0	-	-	0	0.0
70	6	367.8	210.9	[428.6, 726.5]	10	0.0	89.5	[17.5, 298.0]	10	0.0
100		-	-	-	0	-	240.4	[166.0, 348.3]	4	0.0
Medium density test instances										
70	3	0.0	0.4	[0.3, 0.4]	10	0.0	126.8	[18.1, 293.5]	10	0.0
100		0.1	3.8	[1.4, 14.0]	10	0.0	270.2	[242.2, 313.6]	5	0.0
300		4.6	279.1	[47.2, 835.1]	10	0.0	-	-	0	-
400		13.2	204.3	[61.8, 406.0]	4(4)	0.0	-	-	0	-
70	4	0.7	24.5	[2.7, 135.3]	10	0.0	138.1	[35.1, 418.8]	10	0.0
100		3.2	516.0	[16.5, 1620.0]	9(9)	0.0	459.8	[10.5, 697.3]	7	0.0
70	5	-	-	-	0	-	44.5	[15.6, 85.0]	10	0.0
100		-	-	-	0	-	359.5	[145.0, 543.8]	7	0.5*
70	6	-	-	-	0	-	33.2	[17.7, 59.7]	10	0.0
100		-	-	-	0	-	176.9	[82.2, 317.9]	8	0.5*
High density test instances										
70	3	0.0	1.7	[0.7, 2.3]	10	0.0	161.1	[34.4, 563.8]	10	0.0
100		0.2	4.7	[0.8, 16.6]	10	0.0	505.3	[272.4, 908.1]	7	0.6
200		1.5	166.6	[9.6, 800.7]	10	0.0	-	-	0	-
70	4	1.5	300.6	[11.1, 1587.8]	9	0.0	90.4	[29.3, 474.4]	10	0.0
100		-	-	-	0	-	418.6	[264.0, 602.5]	7	0.3*
70	5	-	-	-	0	-	36.7	[2.2, 117.9]	10	0.0
100		-	-	-	0	-	401.8	[201.5, 840.2]	8	0.2
70	6	-	-	-	0	-	38.8	[14.7, 165.7]	10	0.0
100		-	-	-	0	-	247.2	[72.1, 922.8]	10	0.0

Table 6: Large instances.

* 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.