A compact formulation for maximizing the expected number of transplants in kidney exchange programs

Filipe Alvelos¹, Xenia Klimentova², Abdur Rais⁴ and Ana Viana²,³
1 Centro Algoritmi / Departamento de Produção e Sistemas, Universidade do Minho, 4710-057 Braga, Portugal
2 INESC TEC, Campus da FEUP, 4200-465 Porto, Portugal
3 ISEP - School of Engineering, Polytechnic of Porto, 4200-072 Porto, Portugal
4 Centro Algoritmi Universidade do Minho, 4710-057 Braga, Portugal
E-mail: xenia.klimentova@inescporto.pt

Abstract. Kidney exchange programs (KEPs) allow the exchange of kidneys between incompatible donor-recipient pairs. Optimization approaches can help KEPs in defining which transplants should be made among all incompatible pairs according to some objective. The most common objective is to maximize the number of transplants. In this paper, we propose an integer programming model which addresses the objective of maximizing the expected number of transplants, given that there are equal probabilities of failure associated with vertices and arcs. The model is compact, i.e. has a polynomial number of decision variables and constraints, and therefore can be solved directly by a general purpose integer programming solver (e.g. Cplex).

1. Introduction
Kidney transplantation is essential for survival of many patients suffering from chronic kidney disease for which there is no known cure. Kidneys from cadavers used to be the most acceptable source of kidneys for transplantation but it only met a tiny fraction of the demand. Alternative possibilities of transplant existed when there were people willing to donate kidneys to help their loved ones. However, in many cases, kidney recipients and the living donors might be incompatible for success of transplantation because of blood or tissue (physiological) incompatibility.

To bypass this incompatibility deadlock of a recipient-donor pair, the concept of paired kidney exchange between two recipient-donor pairs is utilized increasingly more in practice. In this context, when physiological compatibility is fulfilled, the donor in one pair provides a kidney to the recipient in a second pair in return for the donor in the second pair to provide a kidney to the recipient of the first pair. Two kidney patients may thus benefit from one such “pairwise” exchange of donor kidneys. This concept of kidney exchange can be extended to include more than two pairs and the program facilitating such exchange is generally known as the kidney exchange program.

These programs are typically modelled as a single-objective optimization problem and most past research focused on maximizing the total number of planned transplants [1–3]. Moreover, deterministic mathematical models had been used for optimization which overlooks the likelihood of changes in the pool of recipient-donor pairs (data sets) [4–7]. Changes in the data sets are indeed realistic and should be taken into account as donors can drop out of the pool, recipients...
may die prior to actual implementation of a scheduled program application, or more rigorous compatibility tests performed after “algorithmic” matching determine that pairs selected for the exchange are incompatible.

In this paper we consider the possibility of such unexpected or last-minute changes in modelling by associating probabilities with the recipient-donor pairs. We describe an integer programming model having polynomial number of variables and constraints what allows being solved directly by a general purpose integer programming solver (e.g. Cplex [8]). For the objective of optimization, we maximize the expected number of successful kidney transplants.

2. Problem statement
Given a directed graph $G(V, A)$, the set of vertices $V$ is the set of incompatible donor-patient pairs. Two vertices $i$ and $j$ are connected by arc $(i, j) \in A$ if a donor from pair $i$ is compatible with a patient of pair $j$. A cycle in $G$ represents a feasible kidney exchange between two or more pairs. Usually, a limit, denoted by $K$, is imposed on the length of a cycle to exclude long cycles, which are less desirable due to logistics considerations (since all transplants in a cycle must be done, ideally, at the same time) and also because if a failure occurs, in general, more transplants will be affected in a longer cycle than in a shorter one.

The kidney exchange problem (KEP) for maximizing the number of planned transplants can be defined as follows: find a packing of vertex-disjoint cycles with length at most $K$ with maximum number of arcs.

One of the main problems in the implementation of the solution of a KEP instance is its unreliability. Last-minute testing of donor and patient can elicit new incompatibilities that were not detected before, causing some donations in a cycle to be cancelled (so called, positive cross-matching), patients or donors may become unavailable, e.g., due to illness or to backing out, etc. To take into account this fact we consider a problem where a probability of failure is associated to arcs and vertices. We assume that the probability of arc failure is equal, for all arcs. The same holds for vertices:

\[ p_i = p^v, \forall i \in V, \quad p_{ij} = p^a, \forall (i, j) \in A. \]

It is common practice to allow that vertices in a cycle with failure can be re-matched among them, if a shorter cycle involving only vertices from that cycle can be obtained [9]. Consider the example presented in figure 1 and assume only vertices failure. The presented cycle leads to 3 transplants if none of the three vertices fails, and to 2 transplants (cycle $(1, 2)$) in case of withdrawal of vertex 3 only.

![Figure 1. Cycle with 3 vertices.](image)

The expected number of transplants for this cycle is (see [4]):

\[ E(c) = 3(1 - p_1)(1 - p_2)(1 - p_3) + 2(1 - p_1)(1 - p_2)p_3 \]

and in case of equal probabilities of vertices failure resumes to:

\[ E(c) = 3(1 - p^v)^3 + 2p^v(1 - p^v)^2 \]
For cycles having similar configurations, which may be defined as all non-isomorphic graphs of given size $k$ having at least a cycle of size $k$, the expected value is computed in the same way. The number of different configurations of graphs grows very rapidly with the number of vertices: for $k = 4$ there are 61 different configurations and for $k = 5$ there are 3725 different non-isomorphic graphs.

Denote by $T$ the set of all different configurations of sizes $k$, with $k = 2, \ldots, K$; we index the configurations by $t$. For the KEP with $K = 4$ there exist 66 configurations of cycles:

- 1 configuration of cycles of length 2: $t = 0$;
- 4 non-isomorphic configurations of cycles of length 3: $t = 1, \ldots, 4$;
- 61 non-isomorphic configurations of length 4 with $t = 5, \ldots, 65$.

Each configuration $t \in T$ can be represented by arcs of a basic cycle and its set of additional arcs. In figure 2, left, we represent in solid lines a cycle of size 3, composed by arcs $(1,2), (2,3), (3,1)$. Configurations of this cycle are those involving one, two, or three extra arcs (represented by dashed lines). Similarly, figure 2, right, represents all the possible arcs for a cycle of size 4.

![Figure 2. Cycle of length 3 and 4. The basic cycle is presented by solid lines; additional arcs are dashed.](image)

Figure 3 illustrates all configurations of size 3 and three examples of configurations of size 4.

![Figure 3. Configurations of 3- and 4-cycles.](image)

Denote by $C_t$ the set of arcs in the basic cycle of configuration $t$. Denote by $Q_t$ the set of additional arcs, and let $R_t$ be the set of arcs that define a configuration. For example, for configurations of size 3 presented in figure 3: $C_1 = \{(1,2), (2,3), (3,1)\}$, $Q_1 = \{(2,1), (3,2), (1,3)\}$, $R_1 = \{(1,2), (2,3), (3,1)\}$, $R_2 = \{(1,2), (2,3), (3,1), (2,1)\}$, $R_3 = \{(1,2), (2,3), (3,1), (2,1), (1,3)\}$, $R_4 = \{(1,2), (2,3), (3,1), (2,1), (1,3), (3,2)\}$.

Similarly, for configurations of size 4 presented in figure 3, we have: $R_5 = \{(1,2), (2,3), (3,4), (4,1), (1,3), (3,1), (3,2)\}$, $R_6 = \{(1,2), (2,3), (3,4), (4,1), (2,1), (4,2)\}$, $R_7 = \{(1,2), (2,3), (3,4), (4,1), (1,4)\}$.
If we have equal probabilities of arc and vertex failures, the expected values for all cycles can be calculated in advance without enumerating them, since they are associated with a tractable number of configurations. The same does not happen if we have different probabilities. In this case, calculating the expected values for all cycles implies enumerating all the cycles, which are in exponential number with respect to the size of the graph, and therefore, they may not be tractable for reasonable sized instances. We say that each configuration defines a type of cycle \( t \in T \), and the expected value of that type gives the weight for cycle of a given type \(- v_t, t \in T \). Given these weights on cycles the KEP of maximizing the expected number of transplants may be solved by the so-called cycle formulation [3, 10, 11]. However this formulation has exponential number of variables associated to all cycles in the graph of size up to \( K \), that need to be enumerated a priori. On the other hand, the known compact formulations for the KEP [3] are not suitable for solving it, as the presented problem has the weights associated to the types of cycles which may not be transformed into the weights of arcs and vertices. In the next section we present a new compact integer programming formulation that overcomes the problem.

3. Integer programming formulation

In this section we present a new compact, i.e. with a polynomial number of decision variables and constraints, integer programming formulation for maximizing the expected number of transplants in a KEP when equal probabilities of failure are associated to arcs and vertices. Differently from the usual approaches for KEP optimization problems, this new formulation operates with variables on vertices (not arcs) of the graph.

Let \( N \) be an upper bound on the number of cycles in any solution (e.g. the number of vertices \( |V| \)). The cycles in the solution can be represented by an index \( l \), with \( l \in L = \{1, \ldots, N\} \). In this case we say that a cycle is initialized by vertex \( l \). Given that only cycles with at most \( K \) arcs are feasible, preprocessing allows reducing the number of vertices and arcs of the graphs associated with the cycles initialized by each \( l \), which we represent by subgraphs \( G^l(V^l, A^l) \) (see [3] for more details). The set of vertices \( V^l, l \in L \) is formed by all vertices with an index higher than \( l \), and the vertices that are more distant than the given limit \( K \) are eliminated. The set of arcs is given as \( A^l = \{(i, j) \in A : i, j \in V^l\} \).

Note that as the non-isomorphic configurations of the graph are considered, the label of each vertex in a configuration can be referred as a position of the vertex of the graph in the configuration. For example, in the graph presented in figure 4 the cycle \((1, 5, 4)\) is a 3-cycle of type 2 (see figure 3) and its vertices have positions 3, 1 and 2, respectively. The cycle \((1, 2, 3)\) is a 3-cycle of type 3 having the vertices in positions 2, 3 and 1, respectively.

![Figure 4. Example of graph with three cycles with different configurations.](image)

The following decision variables are used in the model: \( z_{tl} = 1 \) if vertex \( l \) initialize the cycle of type \( t \), \( z_{tl} = 0 \) otherwise, where \( l \in L, t \in T \); \( y_{il}^{qt} = 1 \) if vertex \( i \) is selected for position \( q \) in the cycle of type \( t \) initialized by vertex \( l \), \( y_{il}^{qt} = 0 \), otherwise, \( \forall i \in V^l, q \in P_t = \{1, \ldots, k\}, l \in L, t \in T \). Furthermore, we define \( R_t = (C_t \cup Q_t)/R_t \).
The compact model for the problem under discussion is written as follows:

\[
\begin{align*}
\text{maximize } & \sum_{t \in T} \sum_{l \in L} v^t z_{tl} \\
\text{subject to } & \sum_{l \in L} \sum_{q \in P_t} \sum_{i \in L} y_{iql}^{tl} \leq 1, \forall i \in \bigcup_{l \in L} V^l, \quad (2) \\
& \sum_{q \in P_t} y_{iql}^{tl} \leq \sum_{q \in P_t} y_{iql}^{tl}, \forall l \in L, \forall i \in V^l, t \in T, \quad (3) \\
& z_{tl} = \sum_{q \in P_t} y_{iql}^{tl}, \forall l \in L, t \in T, \quad (4) \\
& z_{tl} = \sum_{i \in V^l} y_{iql}^{tl}, \forall q \in P_t, l \in L, t \in T, \quad (5) \\
& y_{iql}^{ttl} + y_{jql}^{ttl} \leq 1, \forall (i, j) \in \tilde{A}^l, \forall (u, v) \in \tilde{R}_t, l \in L, t \in T, \quad (6) \\
& y_{iql}^{ttl} + y_{jql}^{ttl} \leq 1, \forall (i, j) \in A^l, \forall (u, v) \in R_t, l \in L, t \in T, \quad (7) \\
& z_{tl}, y_{iql}^{tl} \in \{0, 1\}, \forall i \in V^l, q \in P_t, l \in L, t \in T
\end{align*}
\]

Constraints (2) guarantee that each vertex can be assigned only to one cycle of one type in one position. Constraints (3) force vertex \(l\) to be included into the cycle initialized by \(l\). Constraints (4) put vertex \(l\) equal to 1 for some position \(p\) if cycle of type \(t\) is initialized by \(l\) \((z_{tl} = 1)\). Constraints (5) can be seen as assignment constraints for the vertex positions.

Constraints (6) state that two vertices, \(i\) and \(j\) with no arc connecting them \(((i, j) \in \tilde{A}^l)\), cannot be chosen together for positions \(u\) and \(v\) if an arc between positions \(u\) and \(v\) is desired \(((u, v) \in \tilde{R}_t)\). Constraints (7) state that two nodes, \(i\) and \(j\), having an arc connecting them \(((i, j) \in A^l)\), cannot be chosen together for positions \(u\) and \(v\) if an arc between positions \(u\) and \(v\) is not desired \(((u, v) \in R_t)\).

Given that in this problem the expected number of transplants is to be maximized, the probability of failure of a cycle is explicitly taken into account in the model.

For the graph presented in figure 4 the optimal solution includes cycles \((1, 2, 3)\) and \((4, 6)\), having \(z^{31} = y_1^{31} = y_2^{31} = y_1^{31} = 1\) for 3-cycle and \(z^{41} = y_4^{104} = y_4^{204}\) for 2-cycle.

4. Conclusions

The kidney exchange problem is usually treated as a deterministic problem, where all data is considered to be certain, and we want to maximize the number of possible transplants in a pool of incompatible patient-donor pairs. However, in practice, several uncertain events can happen and strongly affect the decisions based on the solutions from deterministic optimization models, e.g. pairs may drop-out of the pool in the interval between algorithm matching and actual transplants, or last minute incompatibility between pairs may be detected.

This paper handles event uncertainty in kidney exchange programs through association of probabilities of arc and vertex failures. The main contribution is the proposal of a compact Integer Programming model, where the maximum expected number of transplants is to be maximized. By assuming that all arcs have the same probability of failure (the same assumption holding for vertices), we can compute \emph{a priori} the expected probability of survival of all types of cycles. This value is in the objective function, weighting cycle selection.
Acknowledgments
This work is financed by the ERDF — European Regional Development Fund through the COMPETE Programme (operational programm for competitiveness), 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”, by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds through FCT within project ”NORTE-07-0124-FEDER-000057”. This work has also been supported by FCT — Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project scope: PEst-OE/EEI/UI0319/2014.

References
[1] de Klerk M, Keizer K, Claas F, Haase-Kromwijk B and Weimar W 2005 American Journal of Transplantation 5 2302–2305
[2] Segev D, Gentry S, Warren D, Reeb B and Montgomery R 2005 The Journal of the American Medical Association 293 1883–1890
[3] Constantino M, Klimentova X, Viana A and Rais A 2013 European Journal of Operational Research 231 57–68
[4] Pedroso J P 2014 Computational Science and Its Applications ICCSA 2014 (Lecture Notes in Computer Science vol 8580) ed Murgante B, Misra S, Rocha A M A, Torre C, Rocha J G, Falcão M I, Taniar D, Apduhan B O and Gervasi O (Springer International Publishing) pp 32–46
[5] Dickerson J, Procaccia A and Sandholm T 2013 In EC-13: Proc. 14th ACM Conference on Electronic Commerce, June
[6] Li Y, Kalbfleisch J, Song P, Zhou Y, Leichtman A and Rees M 2011 Optimization and simulation of an evolving kidney paired donation (KPD) program Working Paper Series 90 Department of Biostatistics, University of Michigan http://www.bepress.com/umichbiostat/paper90
[7] Chen Y, Li Y, Kalbfleisch J, Zhou Y, Leichtman A and Song P K 2012 IEEE Trans. Biomed. Engineering 59 1985–1991
[8] IBM 2014 ILOG CPLEX optimizer http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/ (last accessed in September, 2014)
[9] Manlove D and O’Malley G 2012 Lecture Notes in Computer Science 7276 271–282
[10] Roth A, Sümme T and Ünver M 2007 The American Economic Review 97 828–851
[11] Abraham D, Blum A and Sandholm T 2007 In: Proceedings of the 8th ACM conference on Electronic commerce, June 13-16 295–304