Kidney Exchange Problem: models and algorithms

Lucie Pansart, Hadrien Cambazard, Nicolas Catusse, Gautier Stauffer

Kidney Exchange Problem: models and algorithms

Lucie Pansart[1], Hadrien Cambazard[1], Nicolas Catusse[1], Gautier Stauffer[2]

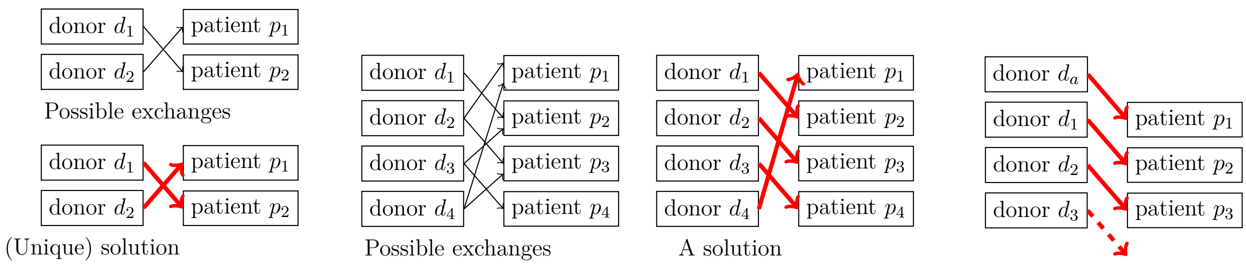
[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France
{lucie.pansart,hadrien.cambazard,nicolas.catusse}@grenoble-inp.fr

[2] Kedge Business School, 33000 Bordeaux, France gautier.stauffer@kedgebs.com

Mots-clés : *kidney exchange problem, integer programming, column generation*

1 Introduction

Kidney transplant is often the only effective treatment to cure end stage renal disease, affecting one out of thousand European citizens. Waiting for a compatible deceased donor can be really long and uncertain, so living donor transplant is a good alternative: a close relation can give one of its kidneys to a patient. As approximately 40% of living donors are incompatible with their specified recipient, several countries have independently developed kidney exchange programs to overcome this issue. In such a program, a patient with an incompatible donor can "swap" its donor with another patient in a similar position (see Figure 1a). More generally, an exchange can involve several incompatible pairs donor-patient by permutation of donors, creating cycles of donation (see Figure 1b). It can also include altruistic donors who give their kidney without waiting another one in return for a relative. Such a donor creates a chain of donation (see Figure 1c). A Kidney Exchange Program seeks for an optimal exchange. The objective of the KEP can be defined by different criteria (number of transplants, weight of exchanges, probability of success...).



(a) A pairwise exchange

(b) $\pi(i)$ is the new donor of patient i: $(\pi(p_1), \pi(p_2), \pi(p_3), \pi(p_4)) = (d_4, d_1, d_2, d_3)$

(c) A chain of donation

FIG. 1: Examples of kidney exchanges. Black edge = compatibility; red and thick edge = chosen transplant. Donor d_i is a relative of patient p_i.

2 Problem

We define the kidney exchange problem (KEP) as the problem of finding an optimal (here of maximal weight) kidney exchanges in a pool of pairs donor-patient.

Let's define a directed graph, called **compatibility graph**, $G = (V = P \cup N, A)$ with a weight function on arcs $w : A \to \mathbb{R}^+$. The vertices P are the incompatible pairs donors-patient, the vertices N are the altruistic donors. We add a weighted arc between vertices u and v if the donor u can give its kidney to patient v. $w(uv)$ = medical benefit of the transplant.

Thus, a solution to KEP is a set of disjoint **walks** (cycles and chains) of maximum weight in graph G (walks must be disjoint because a donor can only give once and a patient must receive only one kidney). Moreover, in practice a cycle of donation must have a limited size k since it implies $2k$ simultaneous surgical operations. A chain of donation can have a limited size l, but no consensus exists on this parameter. So solving the KEP is equivalent to solving a Maximum Weighted $\leq k$-Cycle and $\leq l$-Chain Packing Problem, which is the problem of finding disjoint cycles of size at most k and chains of size at most l of maximal weight.

Complexity Solving the KEP is NP-hard for $k \geq 3$ [1, 3].

3 Solving with integer programming

Many integer programs (IP) exist for this problem [2]. We consider the one called **Cycle& Chain formulation**: $\mathcal{C}$ is the set of all cycles and chains of size at most k and l in G, we define the binary decision variables $x_c = 1$ if walk c is chosen in the solution, 0 otherwise. The objective is to maximize the total weight of selected walks and the constraints are that each vertex must be selected in at most one walk. The Cycle&Chain Formulation is then:

$$\text{(CF)} \quad max \sum_{c \in \mathcal{C}} w_c x_c \tag{1}$$

$$st \sum_{c \in \mathcal{C}: v \in c} x_c \leq 1 \qquad \forall v \in V \tag{2}$$

$$x_c \in \{0, 1\} \qquad \forall c \in \mathcal{C} \tag{3}$$

Unfortunately, the number of variables grows exponentially with l and k. To tackle this problem, we can use a **Branch-and-Price** (B&P) algorithm, which mixes Branch-and-Bound – the standard algorithm solving IP – with Column Generation (CG) [1]. The column generation solves the linear relaxation on a subset of walks, which grows when a new column (=a new variable) is generated.

When we add a variable to the set of current variables, we want to choose a variable which may improve the current solution. Thus, we search a variable (= a feasible walk) with a positive reduced cost. The problem of finding such a walk is called the pricing problem.

Finding a cycle of size at most k and of positive reduced cost can be done in polynomial time with a modified Bellman-Ford algorithm [4].

Finding a chain of size at most l beginning with an altruistic donor of positive reduced cost is $\mathcal{NP}$-complete [5].

No implementation of a KEP solver using the Cycle&Chain formulation has been done in practice to handle large instances to the best of our knowledge. Indeed, it cannot scale when it is implemented without column generation and the use of CG is complex due to the hardness of the pricing problem. The purpose of our work is to provide this solver, thanks to numerous improvements such as heuristics, dual bound, filtering and cutting planes, as well as reasoning with other formulations, including an independent set formulation.

References

[1] David J Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 295–304. ACM, 2007.

[2] Miguel Constantino, Xenia Klimentova, Ana Viana, and Abdur Rais. New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research*, 231(1):57–68, 2013.

[3] John P Dickerson, David F Manlove, Benjamin Plaut, Tuomas Sandholm, and James Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 25–42. ACM, 2016.

[4] K Glorie, A Wagelmans, and J van de Klundert. Iterative branch-and-price for large multi-criteria kidney exchange. *Econometric institute report*, 11, 2012.

[5] Benjamin Plaut, John P Dickerson, and Tuomas Sandholm. Hardness of the pricing problem for chains in barter exchanges. *arXiv preprint arXiv:1606.00117*, 2016.