A Branch-Price-and-Cut algorithm for the Kidney Exchange Problem

Matteo Petris[1,2][0000−0002−8391−0995], Claudia Archetti[2][0000−0002−3524−1600], Diego Cattaruzza[3][0000−0002−1814−2547], Maxime Ogier[3][0000−0002−8503−7632], and Frédéric Semet[3][0000−0002−1334−5417]

[1] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[2] Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School, Cergy-Pontoise, France
{matteo.petris,claudia.archetti}@essec.edu
[3] Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
{diego.cattaruzza,maxime.ogier,frederic.semet}@centralelille.fr

Abstract. We study a Kidney Exchange Problem (KEP) with altruistic donors and incompatible patient-donor pairs. Kidney exchanges can be modelled in a directed graph as circuits or as paths with limited length. The exchanges are associated with a medical benefit to perform the transplants. The aim of the KEP is to determine a set of disjoint kidney exchanges of maximal medical benefit or maximal cardinality. We consider a set packing formulation where the variables are associated with the circuits and paths and we solve it via a Branch-Price-and-Cut (BPC) algorithm. The pricing problem is formulated as an Elementary Longest Path Problem with Length Constraints (ELPPLC). We ensure the correctness of the algorithm by solving it by a label correcting dynamic programming algorithm. We strengthen the linear relaxation via the inclusion of the subset-row inequalities which are separated via two novel heuristics. The results reveal that the BPC algorithm is the only exact approach from the literature able to effectively solve various and difficult benchmark instances.

Keywords: Kidney exchange · Altruistic donors · Elementary paths · Elementary circuits · Branch-Price-and-Cut.

1 Problem Formulation

The KEP can be defined on a directed weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ referred to as *compatibility graph*. The vertex set is $\mathcal{V} = \mathcal{I} \cup \mathcal{D}$ where $\mathcal{I}$ and $\mathcal{D}$ are the sets of incompatible patient-donor pairs and altruistic donors, respectively. The arc set $\mathcal{A}$ contains an arc (i, j) from vertex $i \in \mathcal{V}$ to patient-donor pair $j \in \mathcal{I}$ if the kidney of the donor associated with i is compatible with the patient of pair j. We assign a weight W_{ij} to each arc $(i, j) \in \mathcal{A}$ representing the medical benefit of the transplant. Kidney exchanges are modelled as circuits of length at most $L^C > 1$ and as paths of length at most $L^P > 1$ in graph $\mathcal{G}$. The KEP aims to determine a union of pairwise vertex-disjoint exchanges circuits and paths of maximum weight. If the weighs on the arcs are set to one, the aim of the KEP is to maximise the number of transplants.

Let $\mathcal{E} = \mathcal{E}^C \cup \mathcal{E}^P$ be the set of the exchanges in graph $\mathcal{G}$, where $\mathcal{E}^C$ ($\mathcal{E}^P$) is the set of the exchange circuits (paths) in graph $\mathcal{G}$. The weight of $e \in \mathcal{E}$ is $W_e = \sum_{(i,j)\in e} W_{ij}$. Let a_i^e be a binary parameter equal to one if vertex $i \in \mathcal{V}$ is involved in $e \in \mathcal{E}$ and zero otherwise. For each $e \in \mathcal{E}$, we define a binary variable λ_e taking value one if e is part of the solution and zero otherwise.

The Set Packing formulation [SP] for the KEP reads as follows:

$$[\text{SP}] \quad \max \sum_{e\in\mathcal{E}} W_e \lambda_e \tag{1}$$

$$\text{s.t.} \sum_{e\in\mathcal{E}} a_i^e \lambda_e \leq 1 \qquad \forall i \in \mathcal{V} \tag{2}$$

$$\lambda_e \in \{0,1\} \qquad \forall e \in \mathcal{E}. \tag{3}$$

Objective function (1) maximises the weights of the exchanges. Set Packing Constraints (2) ensure that each vertex is involved in at most one exchange circuit or path. Constraints (3) define variables λ_e as binary.

2 A Branch-Price-and-Cut Algorithm

Formulation [SP] is defined over exponentially-many variables λ_e, $e \in \mathcal{E}$. We solve [SP] by means of a Branch-Price-and-Cut (BPC) algorithm [2].

Let $\pi_i \geq 0$, $i \in \mathcal{V}$ be the dual prices associated with Constraints (2). The reduced cost of a λ_e variable is $\bar{W}_e = W_e - \sum_{i\in\mathcal{V}} a_i^e \pi_i$. The pricing problem is [PP] $\max\{\bar{W}_e : e \in \mathcal{E}\}$ and looks for the most positive reduced cost variables λ_e. It can be decomposed in $|\mathcal{I}| + 1$ independent subproblems: [PP-C](i) $\max\{\bar{W}_e : e \in \mathcal{E}_i^C\}$, $i \in \mathcal{I}$, to price out the variables associated with circuits and [PP-P] $\max\{\bar{W}_e : e \in \mathcal{E}^P\}$ to price out variables associated with paths. These problems can be formulated as an *Elementary Longest Path Problem with Length Constraint* (ELPPLC) which is NP-hard in the strong sense. We exploit the following results from the literature: (i) thanks to the length constraints, problems [PP-C](i) are solvable in polynomial-time by a variant of the Bellman-Ford algorithm (see, e.g., [5]); (ii) the authors in [1] identify conditions when also problem [PP-P] can be solved in polynomial-time.

The BPC algorithm incorporates the following elements of novelty: (i) when the conditions to solve [PP-P] do not occur, the ELPPLC needs to be solved to ensure the correctness of the algorithm. In [1], the ELPPLC is solved via an integer program with exponentially-many constraints. In the BPC algorithm, we solve the ELPPLC via a more efficient label correcting dynamic programming algorithm [4]. (ii) In the BPC algorithm, we strengthen the linear relaxation with the inclusion of non-robust valid inequalities, namely, the subset row inequalities. Usually, in the literature, they are separated by enumeration. However, the size of the KEP instances prevents to do so in reasonable time. Hence, we develop two fast heuristics to separate them which are based on the separation of the *clique* and *odd-hole* inequalities, respectively. In addition, we prove that each violated odd-hole inequality leads to a subset-row inequality with the same violation.

3 Computational Experiments

We test the BPC algorithm on three sets of instances from the literature: PrefLib dataset, set of instances in [5] and [3].

Table 1. Results on the three sets of instances.

Instances						BPC results		
set	L^C	L^P	obj.	$\|\mathcal{I}\|$	#	#opt.	avg.t[s]	avg.gap[%]
PrefLib	3, 4	3, 4, 5, 6	#TR	16 to 512	1360	1360	1.4	-
				1024, 2048	480	480	174.7	-
[5]	3	3, 6, 12	MB	$50, 100, 250$	135	135	7.1	-
				$500, 750, 1000$	135	56	381.2	0.4
[3]	3, 4, 5, 6, 7, 8	0	#TR	50 to 1000	840	840	2.5	-
			MB	50 to 400	480	384	89.6	0.4

The results show that the instances where the objective is the maximisation of the number of transplants (#TR) are easy, no matter the size of the compatibility graphs: all are solved to optimality. Conversely, not all the instances where the objective is the maximisation of the medical benefit (MB) are solved, however, the ones not solved are left with an optimality gap of 0.4%, on average. Against the literature, we provide comparable results with [1] on the PrefLib dataset. We outperform those in [5] and [3] on their respective instance sets.

References

1. Arslan, A.N., Omer, J., Yan, F.: Kidneyexchange. jl: A julia package for solving the kidney exchange problem with branch-and-price. Mathematical Programming Computation pp. 1–34 (2024)
2. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: Column generation for solving huge integer programs. Operations Research **46**(3), 316–329 (jun 1998)
3. Delorme, M., Manlove, D., Smeets, T.: Half-cycle: A new formulation for modelling kidney exchange problems. Operations Research Letters **51**(3), 234–241 (2023)
4. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks **44**(3), 216–229 (2004)
5. Pansart, L., Cambazard, H., Catusse, N.: Dealing with elementary paths in the kidney exchange problem. arXiv preprint arXiv:2201.08446 (2022)