Mathematical models for the kidney exchange problem with reserve arcs

Maxence Delorme[(1)], Wendy Liu[(1)], David Manlove[(2)]

(1) *Department of Econometrics and Operations Research, Tilburg University, The Netherlands*
(2) *School of Computing Science, University of Glasgow, United Kingdom*

Corresponding author `m.delorme@tilburguniversity.edu`

Abstract

The kidney exchange problem with reserve arcs (KEP-RA) is an extension of the classical kidney exchange problem in which one is allowed to select in the solution a limited number of arcs that do not belong to the compatibility graph. This problem is motivated by recent breakthroughs in the field of kidney transplantation involving immunosuppressants that have allowed certain donors to give their kidney to an incompatible recipient. After showing that existing integer linear programming formulations for the kidney exchange problem can easily be extended to KEP-RA, we demonstrate that there always exists an optimal KEP-RA solution in which every cycle contains at most one reserve arc, and we use that property to develop effective variable reduction procedures and new ad hoc modelling structures. Empirical experiments show that trivial model extensions are not able to cope with medium size instances, whereas our enhanced models are able to solve instances with up to a thousand recipient-donor pairs. We also extend our approaches to include non-directed donors. Finally, we evaluate the number of transplants enabled by each reserve arc in various settings and demonstrate that, even though reserve arcs tend to have a diminishing return, there are instances for which this is not the case.

Keywords: Kidney exchange, Reserve arcs, Exact algorithms, Integer programming, Preprocessing.

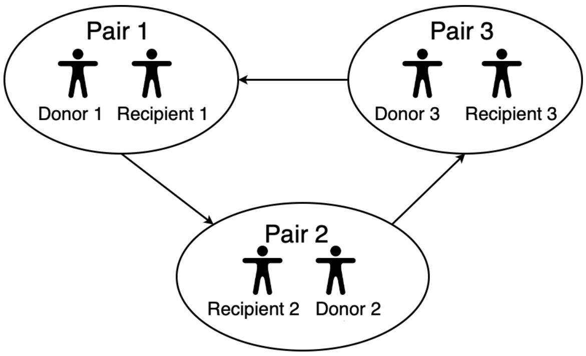
1 Introduction

Kidneys are vital organs located in the lower back of the human body, on either side of the spine. They maintain a healthy balance of water, salts, and minerals, while also filtering blood and removing waste and excessive fluids from the body. If one does not have at least one fully functioning kidney, they are said to have impaired kidney function. It is estimated that roughly 10% of the world's population suffers from such a condition (Bikbov et al. 2020), which often worsens with time, eventually leading to the need for specialist medical care such as dialysis or transplantation. Whereas the latter option offers both a longer life expectancy and a better quality of life, even considering the risks involved in the necessary surgery, the supply of donor kidneys (coming from both living and deceased individuals) is hardly enough to cover the demand.

Besides the fact that the pool of donor kidneys is smaller than the pool of recipients in need of a kidney, there are also compatibility issues that need to be considered: a donor and a recipient do not always form a match from a medical perspective. The National Kidney Foundation (2024) describes two kinds of incompatibilities that may arise between a donor and a recipient: *blood-type incompatibility* and *tissue-type incompatibility*. Disregarding the Rhesus factor, which does not influence blood-type (in)compatibility (National Kidney Foundation 2024), there exist four blood types, namely `O`, `A`, `B`, and `AB`. Whereas an `O` donor is blood-type compatible with all recipients, an `AB` donor is only blood-type compatible with other `AB` recipients. Regarding `A` and `B` donors, they are blood-type compatible with recipients having the same blood type and with `AB` recipients. Fully understanding tissue-type incompatibility requires more advanced medical knowledge. In simple terms, each individual possesses a certain number of Human Leukocyte Antigens (HLA) inherited from their biological parents, which, put together, form an HLA type. Whereas there are only four blood types to consider when determining blood-type (in)compatibility, there are many more HLA types to consider when determining tissue-type (in)compatibility.

If a recipient has found a living donor who is willing and legally allowed to donate them a kidney, the two individuals go through blood typing and tissue typing procedures to determine whether they form a match. If they do not, the recipient-donor pair may consider joining a kidney exchange programme if such a possibility exists in the country in which they reside. A kidney exchange programme allows a set of recipient-donor pairs registered in the system to exchange their donor kidneys if the resulting swaps satisfy a set of requirements. An example of possible swaps between three incompatible recipient-donor pairs is shown in Figure 1. In this figure, a three-way exchange – also referred to as a *cycle* in the literature – is illustrated. Donor 1 donates a kidney to Recipient 2. In return, Donor 2 donates a kidney to Recipient 3. Finally, Recipient 1 receives a kidney from Donor 3. Hence, each pair donates and receives exactly one kidney.

Figure 1: Kidney exchange between three pairs



Non-directed donors (sometimes referred to as altruistic donors) are willing to give one of their kidneys without being paired with any recipient. They can trigger a sequence of transplants that forms a chain instead of a cycle (Roth et al. 2006). Such donors are important in kidney exchange programmes as they significantly increase the number of transplants: indeed, chains are not only less constrained than cycles from a mathematical point of view (as no compatibility requirement is imposed on the donor of the last pair of a chain), they are also less constrained from a logistical point of view (as not all kidney transplants need to take place simultaneously, which is usually the case with cycles, see Biró et al. 2020).

Compatible pairs may also join a kidney exchange programme with the aim of the recipient obtaining a better matched kidney than from their paired donor (e.g., from a younger donor); the pair may also help to create other cycles and chains that would not otherwise exist.

Kidney exchange programmes usually determine a set of feasible cycles and chains at regular intervals using a matching algorithm. For a cycle or chain to be considered as feasible, typically *compatibility* and *cardinality* constraints are imposed. The former ensures that swapped donors are always compatible with their new recipient. Compatibility relationships can be captured by a directed graph, where vertices represent non-directed donors and recipient-donor pairs, and an arc is drawn from one vertex to another if the donor of the first vertex is compatible with the recipient of the second vertex. Cardinality constraints limit the number of recipient-donor pairs involved in the same cycle or chain due to practical constraints. Note that due to chains being less constrained than cycles as described above, typically the maximum chain length L is larger than the maximum cycle length K. When it comes to determining which set of feasible cycles and chains is the best, even though each programme has its own rules (see Biró et al. 2019, Biró et al. 2020 for an overview of these rules in European kidney exchange programmes), one typical goal is to select a set that maximises the total number of transplants. In what follows, we will restrict attention to cycles initially, and later in the paper we show how to model chains. We therefore define the Kidney Exchange Problem (KEP) as the problem of finding a vertex-disjoint set of cycles with the maximum number of transplants, given a set of recipient-donor pairs, a compatibility graph, and a limit on the number of pairs involved in any cycle (Delorme et al. 2023).

Even though the introduction of kidney exchange programmes has substantially increased the number of recipient-donor pairs who were able to proceed with a kidney donation (Biró et al. 2020), it can happen that a pair stays in the programme for many runs without ever being included in a swap. Whereas it is possible for a pair that has spent a long time in the programme to be given a higher priority (see, e.g., the rules in the UK described by Manlove and O'Malley 2015), some highly sensitised recipients can be very difficult to match. In recent years, major breakthroughs involving immunosuppressants and normothermic perfusion machines (Andersson and Kratz 2020, MacMillan et al. 2023) have allowed a donor to give their kidney to a recipient even though their blood types were deemed to be incompatible. Immunosuppressants have also been used to allow a donor to give their kidney to a recipient that was not a perfect HLA match (Aziz et al. 2021). One could think that such breakthroughs may have made kidney exchange programmes obsolete since every donor could, in theory, be made compatible with their recipient. However, this is not yet the case: there is both an additional risk and an additional cost associated with these procedures. These considerations have led the research community to study variants of KEP in which incompatible transplants can be made compatible at a certain cost (Andersson and Kratz 2020, Aziz et al. 2021, Heo et al. 2021). In this work, we study one such variant, the Kidney Exchange Problem with Reserve Arcs (KEP-RA), an extension of KEP in which one is allowed to select a certain number of arcs that do not appear in the compatibility graph (called reserve arcs, or RA).

1.1 Our contribution

Inspired by the recent KEP literature with immunosuppressants, we introduce a four-level problem typology consisting of KEP, KEP-RA, KEP with Half-Compatible Arcs (KEP-HCA), and KEP with Costs (KEP-C). After classifying the existing literature according to the proposed typology, we show that there is a polynomial-time reduction from a problem at a certain level to the problem at the next level, and we report known complexity results for special cases of each problem, namely the case in which only cycles of size two are allowed, the case in which the cycle size is bounded by an integer greater than or equal to three, and the case in which the cycle size is unbounded. We then show that existing Integer Linear Programming (ILP) formulations for KEP can easily be extended to model KEP-RA. We continue by demonstrating that there always exists an optimal KEP-RA solution in which every cycle contains at most one RA, and we use that property to develop an effective variable reduction procedure for each model. We then utilize that same property to create a new effective modelling structure to represent cycles with exactly one RA and we extend our findings to the case where non-directed donors are also considered in the instance. We then empirically evaluate each of the proposed strategies and show that we are able to solve KEP-RA instances with the same order of magnitude as solvable KEP instances, whether reduced-cost variable fixing is used or not. Finally, we provide new managerial insights about KEP-RA, especially regarding how the total number of transplants evolves when the number of permitted RAs increases.

1.2 Layout

The remainder of this paper is structured as follows. In Section 2, we introduce our four-level problem typology, review the existing literature associated with each level, and report known complexity results. In Section 3, we briefly discuss existing ILP models for KEP and show how they can be extended to KEP-RA. In Section 4, we discuss preprocessing techniques and we introduce our new modelling structure for cycles with exactly one RA in Section 5. In Section 6, we show how our findings can be extended to also include non-directed donors. In Section 7, we report the outcomes of extensive computational experiments aimed at evaluating each of the proposed approaches together with some managerial insights. Finally, conclusions are drawn in Section 8.

2 A four-level typology and previous work

In all problems considered in the proposed typology for KEP with immunosuppressents, we are given a set of n recipient-donor pairs together with a limit K on the maximum number of pairs that can be included in any cycle and a compatibility graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where vertex set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ contains one node per recipient-donor pair. The objective is to compute a subset of arcs $\mathcal{A}' \subseteq \mathcal{A}$, called *the transplants*, such that $|\mathcal{A}'|$ is maximised while being solely composed of vertex-disjoint cycles with cardinality at most K.

- In the standard KEP, for every (not-necessarily distinct) recipient-donor pairs $p_1, p_2 \in \mathcal{V}$, arc (p_1, p_2) belongs to $\mathcal{A}$ if the donor of pair p_1 is compatible with the recipient of pair p_2 (the case $p_1 = p_2$ gives a self-loop and corresponds to a compatible pair).
- In KEP-RA, our problem of interest, we are also given a budget $B \in \mathbb{N}_0$. In addition, arc set $\mathcal{A}$ is split in two arc subsets: the standard arcs $\mathcal{A}_s$, and the reserve arcs $\mathcal{A}_r$. Given any (not-necessarily distinct) recipient-donor pairs $p_1, p_2 \in \mathcal{V}$, arc (p_1, p_2) belongs to $\mathcal{A}_s$ if the donor of pair p_1 is compatible with the recipient of pair p_2. If that is not the case, then (p_1, p_2) belongs to $\mathcal{A}_r$. At most B arcs in subset $\mathcal{A}_r$ can be selected in the solution.

- In KEP-HCA, arc set $\mathcal{A}$ is now split into three arc subsets: the standard arcs $\mathcal{A}_s$, the half-compatible arcs $\mathcal{A}_h$, and the incompatible arcs $\mathcal{A}_i$. Given any (not-necessarily distinct) recipient-donor pairs $p_1, p_2 \in \mathcal{V}$, arc (p_1, p_2) belongs to $\mathcal{A}_s$ if the donor of pair p_1 is compatible with the recipient of pair p_2, it belongs to $\mathcal{A}_h$ if the donor of pair p_1 is not immediately compatible with the recipient of pair p_2, but that could be remedied with the use of immunosuppressents, and it belongs to $\mathcal{A}_i$ otherwise. At most B arcs in subset $\mathcal{A}_h$ can be selected in the solution whereas no arcs in $\mathcal{A}_i$ are allowed.
- In KEP-C, arc set $\mathcal{A}$ is not divided into subsets anymore and contains arc (p_1, p_2) for every (not-necessarily distinct) recipient-donor pairs $p_1, p_2 \in \mathcal{V}$. Each arc $a \in \mathcal{A}$ is also associated with a non-negative integer cost $c_a \leq B + 1$. The total cost of the arcs selected in the solution must be at most B.

KEP has been extensively studied in the literature, both from a practical (Biró et al. 2019) and theoretical (Mak-Hau 2017) point of view. Among the contributions to KEP that are directly relevant to this work, we mention Roth et al. (2005) who showed that KEP can be solved in polynomial time when the cycle size limit K is set to 2, and Abraham et al. (2007) who showed that the problem becomes $\mathcal{NP}$-hard when $K \geq 3$, but that it can be solved in polynomial time again when K is unbounded. Due to its applications (one unit of the objective function is often associated with the life of a human being), KEP usually requires an exact solution, which is why a large proportion of the literature has focused on developing exact approaches for the problem, including integer programming (Roth et al. 2007, Abraham et al. 2007, Constantino et al. 2013) and branch-and-price algorithms (Klimentova et al. 2014, Riascos-Álvarez et al. 2024), among others (Delorme et al. 2022, 2024).

Regarding KEP-RA, the problem was studied by Aziz et al. (2021) who referred to it as "the Silver Bullet model". The authors proved that KEP-RA can be solved in polynomial time when the cycle size limit K is unbounded. They also showed that the cycle formulation for KEP (Roth et al. 2007) can easily be extended to model KEP-RA. Heo et al. (2021) studied KEP-RA from a mechanism design point of view in the case where only pairwise exchanges were allowed.

As far as KEP-HCA is concerned, empirical studies were conducted by Sönmez et al. (2018) and Andersson and Kratz (2020) in which the budget was unlimited and where only pairwise exchanges were allowed. The former studied a version of the problem where `A` donors could be made half-compatible with `B` and `O` recipients, whereas the latter studied a version of the problem where all blood-type incompatibilities could be removed with immunosuppressants (but not tissue-type incompatibilities). Aziz et al. (2021) also proposed an ILP model for KEP-HCA and empirically showed that each unit increase in the budget B had a diminishing return in the optimal solution value.

To the best of our knowledge, KEP-C was only studied in the case where $K = 2$ under the name of "budgeted matching" (Berger et al. 2011, Janssen 2022), with a predominant focus on the weighted case (note that in KEP, each arc has unit weight).

Numerous other KEP extensions have been studied in the literature. We mention for example the work of Klimentova et al. (2023) who looked for stable KEP solutions given a preference list for each recipient-donor pair, the work of Blom et al. (2024) who looked for robust KEP solutions taking into account potential failures of donors and recipients, and the work of Carvalho and Lodi (2023) who studied the multi-agent KEP, which arises when several countries join their recipient-donor pools.

We continue this section by showing that, in the proposed four-level typology, there is a polynomial-time reduction from a problem at a certain level to the problem at the next level.

Theorem 1. *KEP $\propto$ KEP-RA $\propto$ KEP-HCA $\propto$ KEP-C*

Proof. Proof. Given in Appendix A. □ □

Next, we show that KEP-C can be solved in polynomial time when the cycle size K is at most 2. We point out that we could not find a formal proof for this theorem in the literature besides the work of Janssen (2022), and the latter does not handle the case in which both endpoints of an edge may be attached to the same vertex. For completeness, we formalise the proof in the following.

Theorem 2. *When $K = 2$, KEP-C can be solved in polynomial time*

Proof. Proof. Given in Appendix B. □ □

We conclude this section by displaying in Table 1 known complexity results for KEP, KEP-RA, KEP-HCA, and KEP-C when $K = 2$, $K \geq 3$, and $K = \infty$. For each case, we state whether the problem is in $\mathcal{P}$, is $\mathcal{NP}$-hard, or is of unknown complexity. We also report the research article in which the proof of complexity can be found.

Table 1: Complexity for KEP, KEP-RA, KEP-HCA, and KEP-C

Problem	$K = 2$	$K \geq 3$	$K = \infty$
KEP	$\mathcal{P}$ (Roth et al. 2005)	$\mathcal{NP}$-hard (Abraham et al. 2007)	$\mathcal{P}$ (Abraham et al. 2007)
KEP-RA	$\mathcal{P}$ (Theorems 1 & 2)	$\mathcal{NP}$-hard (Theorem 1)	$\mathcal{P}$ (Aziz et al. 2021)
KEP-HCA	$\mathcal{P}$ (Theorems 1 & 2)	$\mathcal{NP}$-hard (Theorem 1)	unknown
KEP-C	$\mathcal{P}$ (Theorem 2)	$\mathcal{NP}$-hard (Theorem 1)	unknown

As a last note, we mention that this four-level typology can be used with other KEP extensions such as KEP with non-directed donors or the weighted KEP. In the latter problem, each arc is associated with a weight and the goal is to maximise the total weight of the selected arcs. Note that Table 1 is different for weighted KEP variants as, for example, weighted KEP-C is clearly $\mathcal{NP}$-hard when $K = 2$ and $K = \infty$ by reduction from the Subset Sum problem (Berger et al. 2011).

3 Adapting ILP models for KEP to KEP-RA

In the following, we show how the four most effective (according to the experiments of Delorme et al. 2023) ILP formulations for KEP, namely the Cycle Formulation (CF), the Half-Cycle Formulation (HCF), the Extended Edge Formulation (EEF), and the Position-Indexed Edge Formulation (PIEF), can easily be extended to model KEP-RA.

3.1 Cycle formulation for KEP-RA

In CF, which was introduced by Roth et al. (2007), one enumerates every cycle c of $\mathcal{G}$ containing at most K vertices and gathers them in cycle set $\mathcal{C}$. A binary variable x_c is then associated with each cycle $c \in \mathcal{C}$, taking value 1 if cycle c is selected in the solution and value 0 otherwise. If we denote by $\mathcal{V}(c)$ the set of recipient-donor pairs included in cycle c and by r_c the number of RAs contained in cycle c, the adaptation of CF to KEP-RA is as follows:

$$\max \quad z = \sum_{c \in \mathcal{C}} |\mathcal{V}(c)| \, x_c \tag{1}$$

$$\text{s.t.} \quad \sum_{\substack{c \in \mathcal{C} \\ v \in \mathcal{V}(c)}} x_c \leq 1 \qquad v \in \mathcal{V}, \tag{2}$$

$$\sum_{c \in \mathcal{C}} r_c x_c \leq B, \tag{3}$$

$$x_c \in \{0, 1\} \qquad c \in \mathcal{C}. \tag{4}$$

The objective function (1) maximises the number of transplants, whereas constraints (2) ensure that each recipient-donor pair is included in at most one of the selected cycles. Finally, constraint (3) ensures that the number of RAs used in the solution is at most B. CF has $|\mathcal{C}|$ variables and $|\mathcal{V}| + 1$ constraints. As in the case of a complete compatibility graph, $|\mathcal{C}| \approx |\mathcal{V}|^K$, the number of variables in CF increases very quickly with K and $|\mathcal{V}|$ in practice.

3.2 Half-Cycle formulation for KEP-RA

To reduce the rate of the variable increase displayed by CF, Delorme et al. (2023) introduced HCF, a cycle-based formulation in which each cycle is represented by two compatible halves instead of a whole. From the modelling perspective, such a strategy reduces the number of necessary variables at the expense of supplementary constraints. In practice, one enumerates every sequence of up to $\lceil \frac{K}{2} \rceil + 1$ vertices (also called a half-cycle) and gathers them in set $\mathcal{H}$. A binary variable x_h is then associated which each half-cycle $h \in \mathcal{H}$, taking value 1 if half-cycle h is selected in the solution and value 0 otherwise. If we denote by $\mathcal{V}^s(h)$ the recipient-donor pair starting half-cycle h, by $\mathcal{V}^e(h)$ the recipient-donor pair ending half-cycle h, by $\mathcal{V}^m(h)$ the other recipient-donor pairs included in half-cycle h (i.e., those appearing in the middle of the half-cycle), and by r_h the number of RAs contained in half-cycle h the adaptation of HCF to KEP-RA for even K values is as follows:

$$\max \quad z = \sum_{h \in \mathcal{H}} (|\mathcal{V}^m(h)| + 1)\, x_h \tag{5}$$

$$\text{s.t.} \quad \sum_{\substack{h \in \mathcal{H} \\ v \in \mathcal{V}^s(h) \cup \mathcal{V}^e(h)}} 0.5 x_h + \sum_{\substack{h \in \mathcal{H} \\ v \in \mathcal{V}^m(h)}} x_h \le 1 \qquad v \in \mathcal{V}, \tag{6}$$

$$\sum_{\substack{h \in \mathcal{H} \\ v_1 \in \mathcal{V}^s(h), v_2 \in \mathcal{V}^e(h)}} x_h = \sum_{\substack{h \in \mathcal{H} \\ v_1 \in \mathcal{V}^e(h), v_2 \in \mathcal{V}^s(h)}} x_h \qquad v_1 \in \mathcal{V}, v_2 \in \mathcal{V} : v_2 > v_1, \tag{7}$$

$$\sum_{h \in \mathcal{H}} r_h x_h \le B, \tag{8}$$

$$x_h \in \{0, 1\} \qquad h \in \mathcal{H}. \tag{9}$$

The objective function (5) maximises the number of transplants (note that the starting and ending nodes of a selected half-cycle only count for halves as they both also appear in another selected half-cycle). Constraints (6) make sure that each recipient-donor pair appears at most once in the middle or at most twice at the start or end of the selected half-cycles. Constraints (7) ensure that every selected half-cycle can be combined with another selected half-cycle to form a complete cycle. Finally, constraint (8) makes sure that the number of RAs used in the solution is at most B. When K is odd, one needs to prevent two half-cycles with size $\lceil \frac{K}{2} \rceil + 1$ to be combined together to form a cycle with size $K + 1$. This is done by setting all variables associated with half-cycles $h \in \mathcal{H}$ such that $\mathcal{V}^s(h) > \mathcal{V}^e(h)$ and $|\mathcal{V}^m(h)| = \lceil \frac{K}{2} \rceil$ to 0. HCF has $|\mathcal{H}|$ variables and $O(|\mathcal{V}|^2)$ constraints. In the case of a complete compatibility graph, $|\mathcal{H}| \approx |\mathcal{V}|^{1+\lceil \frac{K}{2} \rceil}$, the number of variables in HCF is smaller than the number of variables in CF when $K \ge 4$.

3.3 Extended edge formulation for KEP-RA

In EEF, which was introduced by Constantino et al. (2013), one duplicates the compatibility graph $\mathcal{G}$ once for each vertex, resulting in n compatibility graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A}_1 = \{\mathcal{A}_{1_s} \cup \mathcal{A}_{1_r}\}), \ldots, \mathcal{G}_n = (\mathcal{V}_n, \mathcal{A}_n = \{\mathcal{A}_{n_s} \cup \mathcal{A}_{n_r}\})$, where $\mathcal{A}_{\rangle_\int}$ and $\mathcal{A}_{\rangle_\triangledown}$ denote the standard and reserve arcs in $\mathcal{G}_\rangle$, respectively ($1 \le i \le n$). For each graph copy corresponding to $v \in \mathcal{V}$, we gather in $\mathcal{A}_v(v')$ the arcs that emanate from the vertex corresponding to recipient-donor v'. We also associate a binary variable x_a^v to each arc $a \in \mathcal{A}_v$, taking value 1 if arc a is selected in graph copy v and value 0 otherwise. If we denote by $t(a)$ the tail of arc a and by $h(a)$ the head of arc a ($v \in \mathcal{V}, a \in \mathcal{A}_v$), the adaptation of EEF to KEP-RA is as follows:

$$\max \quad z = \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v} x_a^v \tag{10}$$

$$\text{s.t.} \quad \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v(v')} x_a^v \le 1 \qquad v' \in \mathcal{V}, \tag{11}$$

$$\sum_{\substack{a \in \mathcal{A}_v \\ h(a) = v'}} x_a^v = \sum_{\substack{a \in \mathcal{A}_v \\ t(a) = v'}} x_a^v \qquad v \in \mathcal{V}, v' \in \mathcal{V}_v, \tag{12}$$

$$\sum_{a \in \mathcal{A}_v} x_a^v \le K \qquad v \in \mathcal{V}, \tag{13}$$

$$\sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_{v_r}} x_a^v \le B, \tag{14}$$

$$x_a^v \in \{0, 1\} \qquad v \in \mathcal{V}, a \in \mathcal{A}_v. \tag{15}$$

The objective function (10) maximises the number of transplants whereas constraints (11) make sure that each recipient-donor pair is included in at most one transplant over the n graph copies. Flow conservation constraints

(12) ensure that the arcs selected in a given graph copy (if any) form (at least) one cycle. Constraints (13) make sure that, for each graph copy $v \in \mathcal{V}$, the cycles selected in the graph copy involve at most K recipient-donor pairs in total (this can be, for example, a single cycle with size K or one cycle with size $K-3$ and one cycle with size 3). Finally, budget constraint (14) limits the total number of RAs used in the solution over the n graph copies. EEF has $O(|\mathcal{V}||\mathcal{A}|)$ variables and $O(|\mathcal{V}|^2)$ constraints, which is polynomial in the input size (unlike CF and HCF), but still grows quickly with $|\mathcal{V}|$ in practice because of the fact that the compatibility graph is complete. Note, however, that several EEF model improvements initially proposed for KEP can be extended to KEP-RA: (i) given a certain node ordering, it is only necessary to consider recipient-donor pair v' in graph copy v if $v \leq v'$ $(v, v' \in \mathcal{V})$, (ii) an arc can only be selected in graph copy $v \in \mathcal{V}$ if v itself belongs to an arc selected in graph copy v, and (iii) an arc can only be selected in graph copy v if there exists at least one feasible cycle satisfying (i) and (ii) that includes that arc.

3.4 Position-indexed edge formulation for KEP-RA

The main drawback of EEF is its weaker Linear Programming (LP) relaxation compared to CF and HCF, which comes from the fact that cycles above the cycle size limit may be created when the decision variables are continuous (e.g., a cycle with size $2K$ where each arc is selected 0.5 times) whereas this was not the case in CF and HCF.

To remedy the situation, Dickerson et al. (2016) introduced PIEF, an extension of EEF in which the vertex set $\mathcal{V}_v$ $(v \in \mathcal{V})$ of each compatibility graph is duplicated once for each position a vertex is allowed to take in a cycle, resulting in K vertex sets $\mathcal{V}_v^1, \ldots, \mathcal{V}_v^K$. The associated arc set $\mathcal{A}_v$ is then updated accordingly to link nodes between vertex sets $\mathcal{V}_v^k$ and $\mathcal{V}_v^{k+1}$ $(k = 1, \ldots, K-1)$ to extend the cycle as well as nodes between vertex sets $\mathcal{V}_v^k$ and $\mathcal{V}_v^1$ $(k = 1, \ldots, K)$ to close the cycle (with a corresponding update made to the RA set A_{v_r}). We gather in $\mathcal{A}_v(v')$ the arcs in graph copy v that emanate from one of the K vertices corresponding to recipient-donor v'. As in EEF, one then associates in each graph copy $v \in \mathcal{V}$ a binary variable x_a^v to each arc $a \in \mathcal{A}_v$, taking value 1 if arc a is selected in graph copy v and value 0 otherwise. The adaptation of PIEF to KEP-RA is as follows:

$$\max \quad z = \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v} x_a^v \tag{16}$$

$$\text{s.t.} \quad \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v(v')} x_a^v \leq 1 \qquad v' \in \mathcal{V}, \tag{17}$$

$$\sum_{\substack{a \in \mathcal{A}_v \\ h(a)=v'}} x_a^v = \sum_{\substack{a \in \mathcal{A}_v \\ t(a)=v'}} x_a^v \quad v \in \mathcal{V}, k = 1, \ldots, K, v' \in \mathcal{V}_v^k, \tag{18}$$

$$\sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_{v_r}} x_a^v \leq B, \tag{19}$$

$$x_a^v \in \{0, 1\} \qquad v \in \mathcal{V}, a \in \mathcal{A}_v. \tag{20}$$

The objective function (16) maximises the number of transplants, capacity constraints (17) make sure that each recipient-donor pair is included in at most one transplant over the n graph copies and over the K cycle positions, flow conservation constraints (18) ensure that the arcs selected in a given graph copy (if any) form exactly one cycle, and budget constraint (19) limits the total number of RAs used over the n graph copies in the solution. PIEF has $O(K|\mathcal{V}||\mathcal{A}|)$ variables and $O(K|\mathcal{V}|^2)$ constraints, but several variable reduction procedures initially proposed for KEP can be extended to KEP-RA: (i) given a certain node ordering, it is only necessary to consider the K vertices associated with recipient-donor pair v' in graph copy v if $v \leq v'$ $(v, v' \in \mathcal{V})$, (ii) if a cycle is formed in graph copy $v \in \mathcal{V}$, then it must start by a vertex associated with recipient-donor pair v, or in other words, $\mathcal{V}_v^1 = \{v\}$ $(v \in \mathcal{V})$, and (iii) an arc can only be selected in graph copy v if there exists at least one feasible cycle satisfying (i) and (ii) that includes that arc.

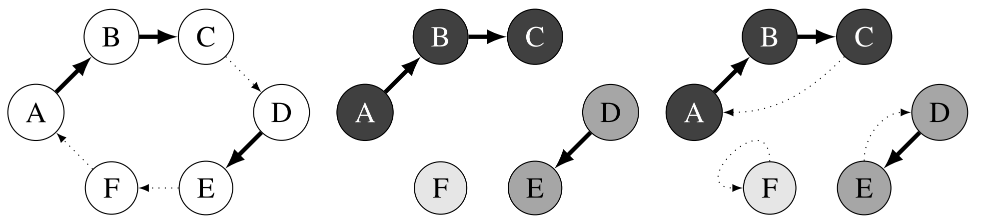
4 Preprocessing techniques for KEP-RA

As will be shown in our computational experiments, the size of KEP-RA instances that can be solved to optimality by the aforementioned models with a state-of-the-art ILP solver is much smaller than the size of solvable KEP instances. Indeed, the fact that the compatibility graph is complete not only increases the number of variables and constraints required by each model, it also decreases the impact of most existing preprocessing techniques tailored for KEP. In this section, we introduce a KEP-RA-specific symmetry-reduction criterion and show how it can be used to derive an effective variable reduction procedure for each of the studied models.

Theorem 3. *There always exists an optimal KEP-RA solution in which every cycle contains at most one RA.*

Proof. Proof. Consider an optimal KEP-RA solution s in which at least one cycle, say c, contains $k \geq 2$ RAs (see the left part of Figure 2, where RAs are represented as dotted arcs). After removing the k RAs from c, one obtains a set of k weakly connected components, each of those either being composed of $v \geq 2$ vertices linked by a set of $v - 1$ arcs forming a directed path, or consisting of a single vertex (see the middle part of Figure 2). One can then add an arc (which may or may not be an RA) to each of the k weakly connected components so as to obtain k cycles, either by linking the last and first vertices of the component's directed path if it has two vertices or more, or by adding a self-loop if the component has only one vertex. We obtain an equivalent KEP-RA solution s' in which every recipient-donor pair that was included in cycle c is now included in one of the k cycles with at most one RA per cycle (see the right part of Figure 2). The resulting solution s' is feasible because: (i) it satisfies the budget constraint as it contains no more RAs than solution s as the solution was constructed by removing k RAs and adding k arcs, which are either standard arcs or RAs, (ii) it satisfies the capacity and cardinality constraints as the new cycles generated are disjoint and their size is always below the size of (feasible) cycle c. Finally, observe that since solution s' includes the exact same recipient-donor pairs as solution s, it is therefore also optimal. □ □

Figure 2: Transforming a cycle with 3 RAs into 3 cycles with one RA



Based on Theorem 3, one does not need to account for the possibility of using more than one RA per cycle in any of the above mentioned models. In particular, only cycles (resp. half-cycles) with one RA need to be generated in set $\mathcal{C}$ (resp. set $\mathcal{H}$) for CF (resp. for HCF) and model improvement (iii) in EEF and PIEF becomes "an arc can only be selected in graph copy v if there exists at least one feasible cycle containing one RA or fewer satisfying (i) and (ii) that includes that arc".

5 A tailored cycle representation for KEP-RA

In this section we introduce a new structure aimed at efficiently modelling cycles with exactly one RA. Such a structure is inspired by the Position-Indexed Chain-Edge Formulation (PICEF), which was introduced by Dickerson et al. (2016) to model chains in KEP. Indeed, following the arguments presented in the previous section, a cycle of size k with exactly one RA can also be seen as a chain of size $k - 1$ with no RA.

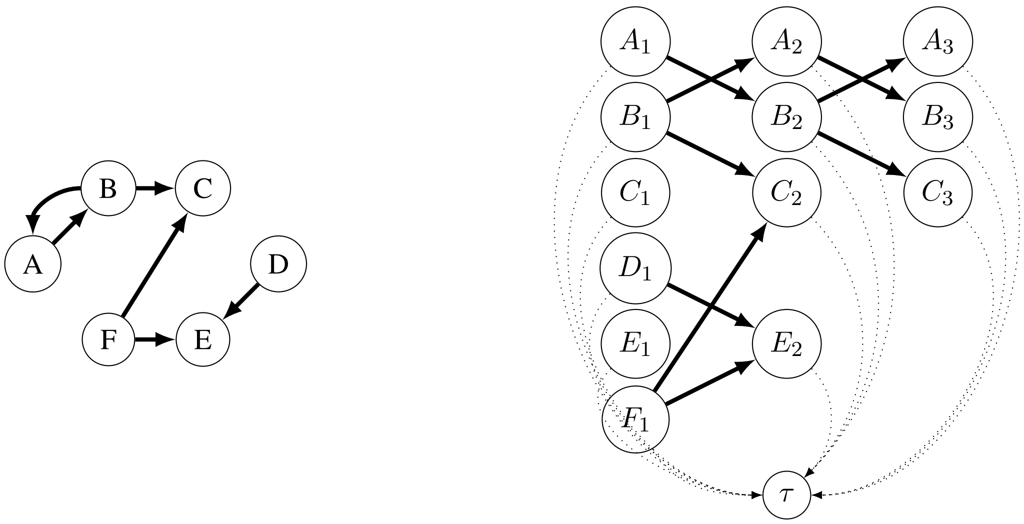
Our structure, which is referred to as Position-Indexed Cycle with One Reserve Arc (PICORA) in the following, models every cycle of size at most K containing exactly one RA as a path in a graph $\mathcal{G}'$ composed of up to K arcs. Given the compatibility graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ of a KEP-RA instance, graph $\mathcal{G}' = (\mathcal{V}', \mathcal{A}')$ is built such that:

- Vertex set $\mathcal{V}' = \{\mathcal{V}'_1 \cup \mathcal{V}'_2 \cup \cdots \cup \mathcal{V}'_K \cup \tau\}$ where $\mathcal{V}'_1$ contains a copy of every vertex that can start a path, $\mathcal{V}'_k$ contains a copy of every vertex that can be in position k ($k = 2, \ldots, K$) in a path, and τ is a dummy vertex used to represent the end of a path. More specifically, $\mathcal{V}'_1$ contains a copy of every vertex in $\mathcal{V}$, and $\mathcal{V}'_k$ contains a copy of every vertex in the set $\{v' \in \mathcal{V} \mid \exists\, v \in \mathcal{V}'_{k-1}$ such that $\mathcal{V}(v) \neq v'$ and $(\mathcal{V}(v), v') \in \mathcal{A}\}$, where $\mathcal{V}(v')$ is a function that maps a vertex copy $v' \in \mathcal{V}'_k$ ($k = 1, \ldots, K$) to its original vertex $v \in \mathcal{V}$.

- Arc set $\mathcal{A}'$ contains a non-terminal arc (v, v') for every distinct pair of vertices $v \in \mathcal{V}'_k, v' \in \mathcal{V}'_{k+1}$ of every position $k = 1, \ldots, K-1$ if both $(\mathcal{V}(v), \mathcal{V}(v')) \in \mathcal{A}$ and $\mathcal{V}(v) \neq \mathcal{V}(v')$, and a terminal arc (v, τ) for every vertex $v \in \mathcal{V}'_k$ of every position $k = 1, \ldots, K$.

Example 1. *Consider the KEP-RA instance with $|\mathcal{V}| = 6$ recipient-donor pairs whose compatibility graph is shown on the left part of Figure 3. The corresponding PICORA graph $\mathcal{G}'$ for $K = 3$ is depicted on the right part of the figure. Cycle $A \to B \to C \to A$ of size 3 with one RA in $\mathcal{G}$ is represented by path $A_1 \to B_2 \to C_3 \to \tau$ in $\mathcal{G}'$ whereas cycle $C \to C$ of size 1 with one RA in $\mathcal{G}$ is represented by path $C_1 \to \tau$ in $\mathcal{G}'$. (Note that the model will forbid a path such as $A_1 \to B_2 \to A_3 \to \tau$, containing two vertices v, v' such that $\mathcal{V}(v) = \mathcal{V}(v')$.)*

Figure 3: Example of the proposed modelling structure for cycles with exactly 1 RA



After constructing graph $\mathcal{G}' = (\mathcal{V}', \mathcal{A}')$, one associates a binary variable y_a to each arc $a \in \mathcal{A}'$ taking value 1 if arc a is selected in the solution and value 0 otherwise. Note that PICORA needs another structure to model cycles with no RA. Any of the structures proposed in the KEP literature (e.g., CF, HCF, EEF, or PIEF) may be used. For example, model CF+PICORA for KEP-RA is as follows:

$$\max \quad z = \sum_{c \in \mathcal{C}} |\mathcal{V}(c)| \, x_c + \sum_{a \in \mathcal{A}'} y_a \tag{21}$$

$$\text{s.t.} \quad \sum_{\substack{c \in \mathcal{C} \\ v \in \mathcal{V}(c)}} x_c + \sum_{\substack{a \in \mathcal{A}' \\ \mathcal{V}(t(a)) = v}} y_a \leq 1 \qquad v \in \mathcal{V}, \tag{22}$$

$$\sum_{\substack{a \in \mathcal{A}' \\ h(a) = v}} y_a = \sum_{\substack{a \in \mathcal{A}' \\ t(a) = v}} y_a \qquad v \in \mathcal{V}' \setminus \{\mathcal{V}'_1 \cup \tau\}, \tag{23}$$

$$\sum_{\substack{a \in \mathcal{A}' \\ h(a) = \tau}} y_a \leq B, \tag{24}$$

$$x_c \in \{0, 1\} \qquad c \in \mathcal{C}, \tag{25}$$

$$y_a \in \{0, 1\} \qquad a \in \mathcal{A}'. \tag{26}$$

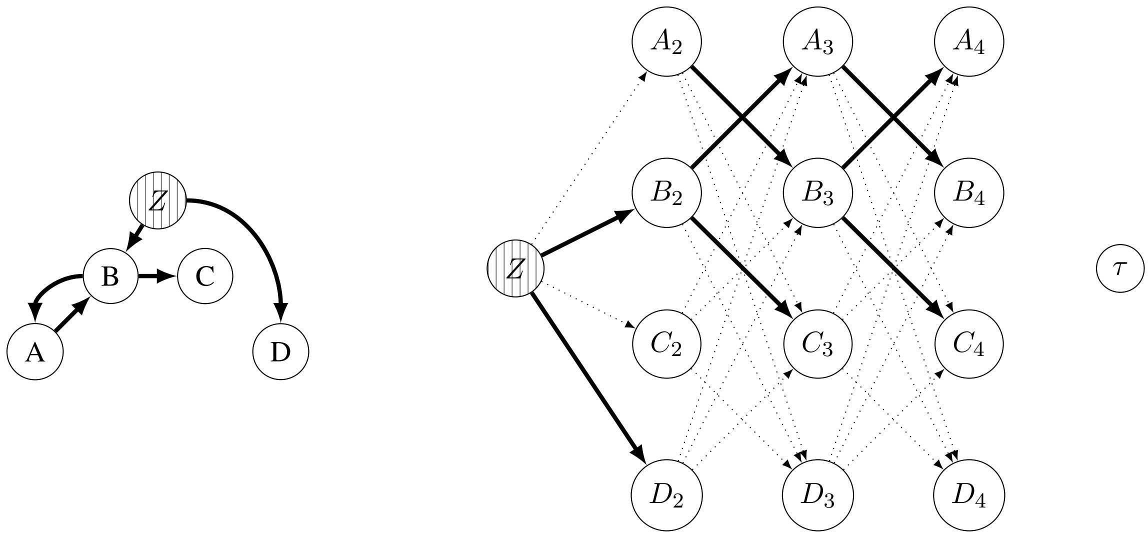
The objective function (21) maximises the number of transplants, whether those occur in the CF structure or in the PICORA structure. Constraints (22) make sure that each recipient-donor pair is included in at most one transplant, independently of the structure in which it is used. Constraints (23) ensure flow conservation at every node in the PICORA structure, except those that start or end a path. Finally, constraint (24) limits the number of paths used in the PICORA structure, which is equivalent to the number of RAs used in the solution, to at most B. On its own, the PICORA structure has $O(K|\mathcal{A}|)$ variables and $O(K|\mathcal{V}|)$ constraints.

6 Including non-directed donors in KEP-RA

Thus far we have included only cycles in our models, but we now show how to incorporate chains in KEP-RA. The most intuitive (but not necessarily the most efficient) way to model chains in KEP-RA is to treat them as cycles, which can be done by pairing each non-directed donor with a dummy recipient that is compatible with the donor of every recipient-donor pair. One can also extend the PICEF structure proposed by Dickerson et al. (2016) and represent every chain (both the ones that include RAs and the ones that do not) as a path in a graph $\mathcal{G}''$. As in PICORA, given the compatibility graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ of a KEP-RA instance, graph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{A}'')$ is built such that vertex set $\mathcal{V}'' = \{\mathcal{V}''_1 \cup \mathcal{V}''_2 \cup \cdots \cup \mathcal{V}''_L \cup \tau\}$ where $\mathcal{V}''_1$ contains a copy of every non-directed donor, $\mathcal{V}''_l$ $(l = 2, \dots, L)$ contains a copy of every vertex $v \in \mathcal{V}$, and τ is a dummy vertex used to represent the end of every chain. Arc set $\mathcal{A}''$ contains an arc (v, v') for every distinct pair of vertices $v \in \mathcal{V}''_l, v' \in \mathcal{V}''_{l+1}, \mathcal{V}(v) \neq \mathcal{V}(v')$ of every position $l = 1, \dots, L-1$, and an arc (v, τ) for every vertex $v \in \mathcal{V}''_l$ of every position $l = 1, \dots, L$. Each selected arc $(v, v') \in \mathcal{A}''$ then contributes 1 to the RA budget if $(\mathcal{V}(v), \mathcal{V}(v'))$ is an RA in $\mathcal{G}$, and 0 otherwise.

Example 2. *Let us consider the KEP-RA instance with four recipient-donor pairs A, B, C, D and one non-directed donor Z for which the compatibility graph is shown on the left part of Figure 4. The corresponding PICEF graph $\mathcal{G}''$ for $L = 4$ is depicted on the right part of the figure (arcs corresponding to RAs are dotted whereas terminal arcs from each vertex to τ are omitted for visibility). Chain $Z \rightarrow D \rightarrow A \rightarrow B$ of length 4 with one RA in $\mathcal{G}$ is represented by path $Z \rightarrow D_2 \rightarrow A_3 \rightarrow B_4 \rightarrow \tau$ in $\mathcal{G}''$, whereas chain $Z \rightarrow B \rightarrow C$ of length 3 with no RA in $\mathcal{G}$ is represented by path $Z \rightarrow B_2 \rightarrow C_3 \rightarrow \tau$ in $\mathcal{G}''$.*

Figure 4: Example of the direct extension of PICEF to model chains with and without RAs

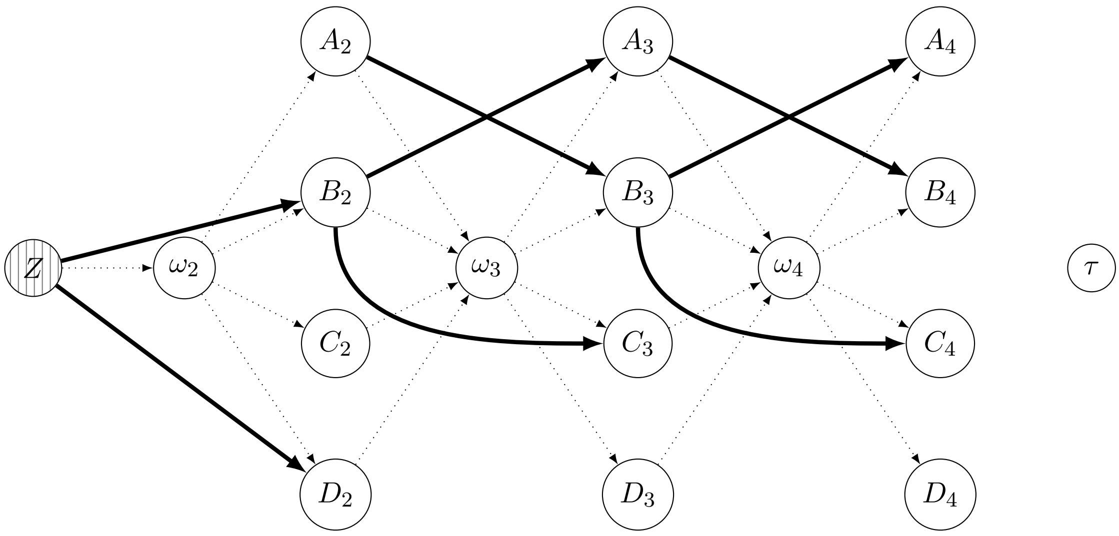


After constructing graph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{A}'')$, one associates a binary variable z_a to each arc $a \in \mathcal{A}''$, taking value 1 if arc a is selected in the solution and value 0 otherwise. Note that the extension of PICEF needs another structure to model cycles with and without RAs, such as those presented in Sections 3, 4, and 5. From a modelling perspective, a set of constraints similar to (23) should be added to enforce flow conservation in the PICEF structure, whereas the objective function, capacity and budget constraints should be updated to include the z_a variables (extending 21, 22, and 24) respectively. The adapted PICEF structure has $O(|\mathcal{V}|^2 L)$ variables and $O(|\mathcal{V}|L)$ constraints.

To reduce the number of arcs that are necessary to model the RAs in $\mathcal{G}''$, one may consider using the original PICEF structure for the standard arcs only, and model the RAs using one dummy vertex ω_l per subset $\mathcal{V}''_l$ $(l = 2, \dots, L)$ attached to each vertex $v \in \mathcal{V}''_{l-1}$ by an arc (v, ω_l), and to each vertex $v \in \mathcal{V}''_l$ by an arc (ω_l, v). (Each selected arc of the form (ω_l, v) contributes 1 to the RA budget.) This reduced PICEF structure has $O((|\mathcal{A}| + |\mathcal{V}|)L)$ variables and $O(|\mathcal{V}|L)$ constraints and results in a significant model size decrease in practice.

Example 2. *(resumed) Let us consider the same KEP-RA instance with four recipient-donor pairs A, B, C, D and one non-directed donor Z for which the compatibility graph is shown on the left part of Figure 4. The* *corresponding reduced PICEF graph $\mathcal{G}''$ for $L = 4$ is depicted in Figure 5 (arcs corresponding to RAs are dotted whereas terminal arcs from each non-dummy vertex to τ are omitted for visibility). Chain $Z \rightarrow D \rightarrow A \rightarrow B$ of length 4 with one RA in $\mathcal{G}$ is represented by path $Z \rightarrow D_2 \rightarrow \omega_3 \rightarrow A_3 \rightarrow B_4 \rightarrow \tau$ in $\mathcal{G}''$ whereas chain $Z \rightarrow B \rightarrow C$ of length 3 with no RA in $\mathcal{G}$ is represented by path $Z \rightarrow B_2 \rightarrow A_3 \rightarrow \tau$ in $\mathcal{G}''$.*

Figure 5: Example of the improved extension of PICEF to model chains with and without RAs



Finally, we discuss the adaptation of Theorem 3 when chains are considered in KEP-RA.

Theorem 4. *There always exists an optimal KEP-RA solution in which no chains include an RA that is followed by $K - 1$ or fewer standard arcs.*

Proof. Proof. Given in Appendix C. □ □

Based on Theorem 4, one does not need to account for the possibility of using an RA in a chain unless it is followed by at least K standard arcs. This means that:

- A valid reduction procedure forbids two consecutive RAs in a chain. In the reduced PICEF adaptation, this means that arc (v, ω_l) only needs to be created if v is the head of at least one standard arc, whereas arc (ω_l, v) only needs to be created if v is the tail of at least one standard arc. For example in Figure 5, arcs (ω_2, C_2), (C_2, ω_3), (ω_3, D_3), (D_3, ω_4), and (ω_4, D_4) do not need to be generated, and therefore, nodes C_2, D_3, and D_4 can be removed;

- A valid reduction procedure forbids RAs to occur in the last K arcs of a chain (disregarding the final arc to τ). In the reduced PICEF adaptation, this means that only dummy vertices $\omega_2, \omega_3, \ldots, \omega_{L-K}$ and their associated arcs need to be generated. It also means that RAs do not need to be considered in chains unless $L \geq K + 2$. For example in Figure 5, it is not necessary to consider RAs in chains if $K \geq 3$. If $K = 2$, then one only needs to consider dummy vertex ω_2 in the chain structure. The chain corresponding to path $Z \rightarrow D_2 \rightarrow \omega_3 \rightarrow B_3 \rightarrow C_4 \rightarrow \tau$ in $\mathcal{G}''$ can be replaced by combining the chain corresponding to path $Z \rightarrow D_2 \rightarrow \tau$ of length 2 without RAs with cycle $B \rightarrow C \rightarrow B$ of size 2 with one RA.

7 Computational experiments

In this section, we first empirically evaluate the performance of each of the models and techniques introduced in Sections 3, 4, and 5 and we measure the improvement brought by reduced-cost variable fixing (see Section D of the Appendix for a description of reduced-cost variable fixing) on the most promising solution methods. We then compare the strategies presented in Section 6 to take non-directed donors into account. Finally, we analyze how the total number of transplants evolves when the number of allowed RAs changes. Our algorithms were all implemented in `C++` and can be downloaded from `https://github.com/mdelorme2/`

Mathematical_models_for_the_kidney_exchange_problem_with_reserve_arcs. All computational tests were executed on a virtual machine AMD EPYC-Rome Processor with 2.00 GHz and 64 GB of RAM memory, running under Ubuntu 20. The ILP models were solved using Gurobi 11.0.2. A single core was used for the tests and the barrier algorithm was used to solve the root nodes of the models. For the algorithms using reduced-cost variable fixing, we deactivated the crossover operation when solving the LP models. For every run, a time limit of 3600 seconds was imposed.

7.1 Evaluating the proposed solution methods for KEP-RA

We first evaluated the performance of the proposed adaptations of models CF, HCF, EEF, and PIEF to KEP-RA using the KEP instances introduced by Delorme et al. (2023) which were created using the instance generator of Delorme et al. (2022) available at https://wpettersson.github.io/kidney-webapp. This dataset contains 20 instance files for each size $n \in \{50, 100, 200, 400, 600, 800, 1000\}$. In our first set of experiments, we selected the instances with size up to 400 and ran these with $K \in \{3, 4\}$ and $B \in \{0, 1, 2, 3, 4, 5\}$, resulting in $4 \times 20 \times 2 \times 6 = 960$ KEP-RA instances in total. We solved these with 16 solution methods MODEL+CONFIGURATION where: MODEL is either (i) CF as described in Section 3.1, (ii) HCF as described in Section 3.2, (iii) EEF as described in Section 3.3, or (iv) PIEF as described in Section 3.4; and CONFIGURATION is either (i) NONE if the reduction procedure derived from Theorem 3 is not applied, (ii) 1RA if this reduction procedure is applied, (iii) PICORA if the structure introduced in Section 5 is used to model cycles with RAs, or (iv) PICORA+RCVF if this structure is used together with reduced-cost variable fixing.

We report in Table 2 the results obtained by each model in configurations NONE and 1RA on instances with $n = 50$ recipient-donor pairs. The first two columns of the table identify the model and the number of allowed RAs. The following columns provide, for each configuration and each K value, the number of optimal solutions found (column "#opt"), the average CPU time in seconds over each run of the dataset including the ones terminated by the time limit (column "T(s)"), and the average number of variables (column "#var") and constraints (column "#const") in the model.

Table 2: Results of the tested models in configurations NONE and 1RA for instances with $n = 50$

Model	B	$K = 3$								$K = 4$							
		NONE				1RA				NONE				1RA			
		#opt	T(s)	#var	#const	#opt	T(s)	#var	#const	#opt	T(s)	#var	#const	#opt	T(s)	#var	#const
CF	0	20	0	23	51	20	0	23	51	20	0	66	51	20	0	66	51
	1	20	0	983	51	20	0	983	51	20	0	3534	51	20	0	3534	51
	2	20	0.2	10,991	51	20	0	983	51	20	1	62,870	51	20	0.1	3534	51
	3	20	0.5	40,475	51	20	0	983	51	20	6.4	477,360	51	20	0.1	3534	51
	4	20	1.4	40,475	51	20	0	983	51	20	22	1,422,275	51	20	0.1	3534	51
	5	20	1	40,475	51	20	0	983	51	20	213	1,422,275	51	20	0.1	3534	51
HCF	0	20	0	90	68	20	0	90	68	20	0	136	78	20	0	136	78
	1	20	0	1250	331	20	0	1250	331	20	0.1	2935	385	20	0.1	2935	385
	2	20	0.3	12,216	1276	20	0	1250	331	20	1	32,234	1276	20	0.1	2935	385
	3	20	1.2	41,700	1276	20	0	1250	331	20	2	80,900	1276	20	0.1	2935	385
	4	20	1	41,700	1276	20	0	1250	331	20	3.8	80,900	1276	20	0.1	2935	385
	5	20	1	41,700	1276	20	0	1250	331	20	2.3	80,900	1276	20	0.1	2935	385
EEF	0	20	0	47	131	20	0	47	181	20	0	101	154	20	0	101	204
	1	20	0	1337	490	20	0	1337	540	20	0.1	2462	560	20	0.1	2462	610
	2	20	0.5	12,427	1376	20	0	1337	540	20	1	17,495	1376	20	0.2	2462	610
	3	20	1.5	42,925	1376	20	0.1	1337	540	20	2.3	42,925	1376	20	0.2	2462	610
	4	20	1.4	42,925	1376	20	0.1	1337	540	20	4.2	42,925	1376	20	0.2	2462	610
	5	20	1.6	42,925	1376	20	0.1	1337	540	20	8.3	42,925	1376	20	0.3	2462	610
PIEF	0	20	0	49	83	20	0	49	83	20	0	118	117	20	0	118	117
	1	20	0	1469	580	20	0	1469	580	20	0.1	3199	886	20	0.1	3199	886
	2	20	0.4	13,354	2253	20	0	1469	580	20	0.9	26,651	3041	20	0.1	3199	886
	3	20	1	44,150	2551	20	0	1469	580	20	2.1	80,101	3664	20	0.1	3199	886
	4	20	1.2	44,150	2551	20	0	1469	580	20	2.3	85,800	3776	20	0.1	3199	886
	5	20	1.7	44,150	2551	20	0	1469	580	20	2	85,800	3776	20	0.1	3199	886

First and foremost, the experiments clearly show that every model has a much lower computation time in configuration 1RA than in configuration NONE. This can be explained by the fact that the number of variables and constraints for all models stops increasing once $B = 1$ in configuration 1RA whereas it only stops increasing once $B = K$ in configuration NONE. Interestingly, we observe that the model size alone does not fully explain the poor empirical performance displayed by the models in configuration NONE: for example for instances where $n = 50$ and $K = 4$, CF+NONE took 22 seconds on average to solve an instance when $B = 4$ whereas it took 213 seconds on average to solve an instance when $B = 5$. Further investigation showed that this difference could mostly be attributed to the presolve phase of the solver, which was significantly longer when $B = 5$ than when

$B = 4$, while not being more effective. As far as comparing the models in configuration 1RA is concerned, not much can be said at this stage as all instances with 50 recipients could be solved within a fraction of a second.

We report in Table 3 (resp. Table 6 in the Appendix) the results obtained by each model in configurations 1RA, PICORA, and PICORA+RCVF on instances with $n \in \{50, 100, 200, 400\}$ recipient-donor pairs and cycle size limit $K = 4$ (resp. $K = 3$). The first two columns of the tables now identify the instance size and the number of allowed RAs whereas the following columns provide, for each model and for each configuration, the number of optimal solutions found and the average CPU time. Information related to the average model size is provided in Tables 7 and 8 of the Appendix.

Table 3: Results of the tested models in configurations 1RA, PICORA, and PICORA+RCVF for instances with $K = 4$

n	B	CF						HCF						EEF						PIEF					
		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF	
		#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)
50	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0
	1	20	0	20	0	20	0	20	0.1	20	0	20	0	20	0.1	20	0	20	0	20	0.1	20	0	20	0
	2	20	0.1	20	0	20	0	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.1	20	0	20	0
	3	20	0.1	20	0	20	0	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.1	20	0	20	0
	4	20	0.1	20	0	20	0	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.1	20	0	20	0
	5	20	0.1	20	0	20	0	20	0.1	20	0	20	0	20	0.3	20	0	20	0	20	0.1	20	0	20	0
100	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0.1	20	0.1	20	0	20	0.1	20	0.1	20	0.1
	1	20	0.5	20	0.1	20	0	20	1.1	20	0.1	20	0	20	29.1	20	0.2	20	0.1	20	1.2	20	0.2	20	0.1
	2	20	1.1	20	0.1	20	0.1	20	1.2	20	0.1	20	0.1	20	80	20	0.2	20	0.1	20	1.1	20	0.2	20	0.1
	3	20	1.1	20	0.1	20	0.1	20	1.2	20	0.1	20	0.1	20	85.6	20	0.2	20	0.1	20	1.4	20	0.2	20	0.1
	4	20	1	20	0.1	20	0.1	20	1.3	20	0.1	20	0.1	20	98.8	20	0.3	20	0.3	20	1.4	20	0.2	20	0.2
	5	20	1	20	0.1	20	0.1	20	1.2	20	0.1	20	0.1	20	54.7	20	0.2	20	0.2	20	1.5	20	0.2	20	0.1
200	0	20	0.4	20	0.4	20	0.2	20	0.4	20	0.4	20	0.3	20	1.6	20	1.5	20	1.4	20	0.9	20	1	20	0.7
	1	20	16.3	20	0.8	20	0.4	20	16.9	20	0.9	20	0.5	13	2007.1	20	2.2	20	1.6	20	36.5	20	1.6	20	0.9
	2	20	21	20	0.8	20	0.5	20	19	20	1	20	0.6	11	1971.7	20	2.1	20	1.4	20	45.4	20	1.5	20	1.1
	3	20	21	20	0.8	20	0.5	20	18.2	20	0.9	20	0.5	14	1662	20	1.9	20	1.3	20	39.6	20	1.7	20	1.1
	4	20	22.8	20	0.7	20	0.5	20	17.2	20	0.9	20	0.6	16	1429	20	2	20	1.4	20	49.1	20	1.6	20	1.1
	5	20	22.4	20	0.9	20	0.6	20	18.1	20	1	20	0.6	14	1585.1	20	1.9	20	1.4	20	52.8	20	1.6	20	1.2
400	0	20	8.7	20	9.9	20	5.1	20	17.4	20	18.6	20	9.4	20	64.4	20	58.9	20	45.8	20	34.3	20	34.7	20	19.1
	1	20	488.8	20	16.7	20	7.5	20	477.3	20	26	20	12.3	1	3561.4	20	102.9	20	66.2	19	1283.8	20	41.6	20	23.1
	2	20	457.7	20	14.5	20	7.7	20	460.1	20	31.6	20	13.2	1	3480.5	20	88.2	20	48.5	19	1334.3	20	46.2	20	23.9
	3	20	466	20	16	20	7.7	20	501.6	20	28.2	20	12.6	0	3600	20	91	20	51.9	19	1472.1	20	40.1	20	22.4
	4	20	487.9	20	16.3	20	8	20	521.5	20	28.2	20	11.8	2	3429.4	20	87.5	20	49.2	19	1519.8	20	42.1	20	23.4
	5	20	463.7	20	14.7	20	7.7	20	589.1	20	28.3	20	13.1	0	3600	20	72.3	20	44.1	19	1610	20	49.3	20	23.5

Indubitably, every model obtained better performance in configuration PICORA than in configuration 1RA, which can mostly be attributed to a sharp reduction in the model size when using the former configuration over the latter. As far as reduced-cost variable fixing is concerned, it appears that using the technique is beneficial for every model (e.g., HCF+PICORA needed 28.3s on average to solve an instance with $n = 400, K = 4$, and $B = 5$ whereas HCF+PICORA+RCVF only needed 13.1s).

Finally, we point out that the average computation time increase caused by RAs (i.e., when B goes from 0 to 1) is greatly reduced in configurations PICORA and PICORA+RCVF: for instances with $n = 400$ and $K = 4$, the average solving time jumped from 8.7s to 488.8s for CF+1RA, whereas it only went from 9.9s to 16.7s for CF+PICORA and from 5.1s to 7.5s for CF+PICORA+RCVF. This indicates that RAs can be taken into account without significantly increasing the average solving time, and that, therefore, we should be able to solve KEP-RA instances with the same order of magnitude as solvable KEP instances. We confirm this conclusion in Table 9 of the Appendix, where we evaluate the performance of each model in configuration PICORA+RCVF for instances with $n = \{600, 800, 1000\}$. Clearly, CF, HCF, and PIEF are able to solve large size KEP-RA instances in configuration PICORA+RCVF, but this is not the case for EEF when $K = 4$.

7.2 Evaluating the proposed solution methods for KEP-RA with non-directed donors

In this second set of experiments, we compared two strategies presented in Section 6 to include non-directed donors: one that uses the direct extension of PICEF (corresponding to Figure 4) to model the chains, which is referred to as DPICEF, and one that uses the improved extension of PICEF (corresponding to Figure 5) together with the preprocessing derived from Theorem 4, which is referred to as IPICEFTH4. We used CF+PICORA to model the cycles with and without RAs as it obtained the best performance in our previous set of experiments, and we also used reduced-cost variable fixing to reduce the size of the overall model. We created new KEP instances using the same instance generator as before, in which the field "Proportion of donors who are altruistic" was set to 0.1. An instance with 600 recipient-donor pairs would have size $n = 667$ in total, which includes 67 non-directed donors, as $\frac{67}{667} \approx 0.1$. We report in Table 4 the results obtained by CF+PICORA+DPICEF+RCVF and by CF+PICORA+IPICEFTH4+RCVF on instances with $n \in \{600+67, 800+89, 1000+111\}$, $K \in \{3,4\}$, $L \in \{K, 2K\}$, and $B \in \{0,1,2,3,4,5\}$. A similar table with the average model size is provided in Table 10 of the Appendix.

Table 4: Results of CF+PICORA+DPICEF+RCVF and CF+PICORA+IPICEFTH4+RCVF for large instances with chains

n	B	CF+PICORA+DPICEF+RCVF								CF+PICORA+IPICEFTH4+RCVF							
		$K=3$				$K=4$				$K=3$				$K=4$			
		$L=3$		$L=6$		$L=4$		$L=8$		$L=3$		$L=6$		$L=4$		$L=8$	
		#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)
667	0	20	1.8	20	16.7	20	63.4	20	128.6	20	1.8	20	17.4	20	63.8	20	130.6
	1	20	11.5	20	124	20	134.9	20	698.5	20	4.8	20	27.4	20	86	20	161.9
	2	20	11.1	20	104	20	135.5	20	580.9	20	4.5	20	24.7	20	85.5	20	166.7
	3	20	11.1	20	98.7	20	131.1	20	439.1	20	4.7	20	25.8	20	85.5	20	146.5
	4	20	11	20	90.5	20	130.8	20	522.8	20	4.6	20	25.8	20	78.7	20	146.3
	5	20	10.9	20	94.7	20	130.5	20	457.7	20	4.8	20	24.6	20	85.8	20	155.1
889	0	20	4.5	20	46.1	20	179.4	20	319.8	20	4.4	20	47.4	20	153.5	20	308.8
	1	20	27	20	334.5	20	453.7	19	2655	20	11.3	20	74.2	20	204.1	20	359.8
	2	20	27.1	20	311.7	20	382.7	20	2479.8	20	11.5	20	77.1	20	197.3	20	435.1
	3	20	26.7	20	297.2	20	355.2	20	1889.3	20	11.2	20	68.9	20	201.7	20	411.8
	4	20	26	20	318	20	375.8	20	1412.6	20	11.4	20	69.8	20	229	20	432.7
	5	20	25.9	20	302.1	20	402.7	20	1517	20	11.6	20	72	20	198.7	20	407.1
1111	0	20	10.2	20	92.7	20	511	20	850.9	20	9.3	20	88.4	20	470	20	830.4
	1	20	55.5	20	697.6	20	2149.3	3	3400.6	20	26.7	20	138.8	20	606.2	20	933.4
	2	20	53.9	20	746.2	20	1417.3	0	3600	20	24.2	20	132.2	20	576.5	20	965.5
	3	20	52.7	20	667	20	892.4	2	3570	20	24.5	20	138.3	20	562.9	20	906.2
	4	20	53.1	20	621.5	20	1029.6	10	3076	20	24	20	145.8	20	554.7	20	903.2
	5	20	53.1	20	582.6	20	1047.1	18	2779.1	20	25.6	20	138.7	20	644.3	20	1103.1

Even though CF+PICORA+DPICEF+RCVF was able to solve every large size instance within the time limit, except those where $K=4$ and $L=8$, the performance increase resulting from the improved PICEF structure and the preprocessing derived from Theorem 4 is significant as witnessed by the sharp reduction in the computation time needed by CF+PICORA+IPICEFTH4+RCVF, and the fact that the approach is able to solve all tested large size instances. Such a speed-up can be explained by the significant difference in the number of variables required by the two approaches when RAs are considered (especially for larger K and L values, as shown in Table 10 in the Appendix). Interestingly, we observe that CF+PICORA+IPICEFTH4+RCVF is only mildly impacted by the presence of RAs, with an average computation time around a third higher when $B > 0$ compared to when $B = 0$, whereas CF+PICORA+DPICEF+RCVF is strongly impacted by the presence of RAs, as witnessed by its results on instances with $n = 1111, K = 4$, and $L = 8$: all instances could be solved when $B = 0$, but only 3 could be solved when $B = 1$. We also notice that the difficulty of a KEP-RA instance did not increase linearly with B for the two approaches.

7.3 Measuring the number of additional transplants brought by RAs

Our last set of experiments is aimed at empirically evaluating the impact of RAs in various settings. To do so, we generated smaller KEP instances with a 0.1 proportion of non-directed donors and $n \in \{50+6, 100+11, 200+22, 400+44\}$, and solved them with maximum cycle size $K \in \{3,4\}$, maximum chain length $L \in \{1, K, 2K\}$, and maximum number of allowed RAs $B \in \{0,1,2,3,4,5\}$. The average number of transplants in each setting is reported in Table 5 in column "#trans", whereas the average number of extra transplants brought by the addition of the last RA is displayed in column "#extra". Note that when $L = 1$, each non-directed donor is involved in a chain that links them directly to the terminal node τ, and therefore counts towards 1 unit in the objective function. The latter can be seen as not allowing chains in KEP-RA.

Interestingly, we observe that RAs have a much larger impact when the instance size is small: for instances with $n = 222$, even the 5th RA still brings 2.8 additional transplants on average when $K = 3$ and $L = 1$, whereas it only brings 2.1 additional transplants on average for instances with $n = 400$. We also notice that RAs have a smaller impact when the maximum cycle size increases: for instances with $n = 222$, the 1st RA brings 2.1 additional transplants on average when $K = L = 3$, whereas it only brings 1.1 additional transplants on average when $K = L = 4$. We also observe that the presence of RAs is less impactful when long chains are considered: for instances with $n = 56$ and $K = 3$, the 5th RA brings 2.7 additional transplants on average when $L = 1$, whereas it only brings 1.9 additional transplants on average when $L = 3$, and 1.2 when $L = 6$. Finally, we observe that in several settings, especially for large instances where long chains are allowed, each

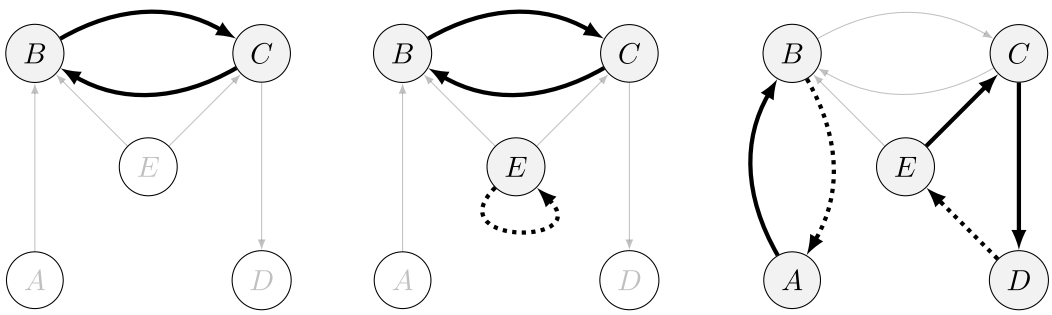
Table 5: Average number of transplants in KEP-RA instances with chains depending on n, B, K, and L

n	B	$K=3$						$K=4$					
		$L=1$		$L=3$		$L=6$		$L=1$		$L=4$		$L=8$	
		#trans	#extra	#trans	#extra	#trans	#extra	#trans	#extra	#trans	#extra	#trans	#extra
56	0	22.9	-	31.4	-	37.6	-	27.2	-	36.2	-	38.7	-
	1	25.9	3	34.4	3	40.2	2.6	31	3.8	39.2	3	41	2.3
	2	28.9	3	37.2	2.8	42.1	1.8	34.3	3.3	41.4	2.3	42.7	1.8
	3	31.7	2.9	39.7	2.6	43.5	1.4	37.2	3	43.1	1.7	43.9	1.2
	4	34.5	2.8	42.1	2.4	44.8	1.3	39.7	2.5	44.6	1.5	45	1.1
	5	37.2	2.7	44	1.9	46	1.2	41.5	1.9	45.8	1.2	46	1
111	0	53.7	-	67.7	-	77.7	-	63.1	-	77	-	78.3	-
	1	56.7	3	70.6	2.9	79.5	1.8	66.9	3.8	79	2	79.6	1.3
	2	59.7	3	73.6	3	80.7	1.2	70.3	3.5	80.5	1.6	80.8	1.2
	3	62.7	3	76.4	2.9	81.8	1.1	73.3	3	81.6	1.1	81.8	1
	4	65.7	3	79	2.6	82.8	1	76	2.8	82.7	1.1	82.8	1
	5	68.6	2.9	81.4	2.4	83.8	1	78.2	2.2	83.7	1	83.8	1
222	0	136.9	-	164	-	166.4	-	148.8	-	166.4	-	166.4	-
	1	139.9	3	166.1	2.1	167.4	1	151.4	2.7	167.4	1.1	167.4	1
	2	142.9	3	167.8	1.7	168.4	1	153.6	2.2	168.4	1	168.4	1
	3	145.8	3	169	1.2	169.4	1	155.8	2.2	169.4	1	169.4	1
	4	148.7	2.9	170.2	1.2	170.4	1	157.6	1.8	170.4	1	170.4	1
	5	151.5	2.8	171.2	1	171.4	1	159.3	1.7	171.4	1	171.4	1
444	0	297.4	-	331.3	-	331.4	-	301.7	-	331.4	-	331.4	-
	1	300.3	2.9	332.4	1.1	332.4	1	304	2.3	332.4	1	332.4	1
	2	303.1	2.8	333.4	1	333.4	1	306	2	333.4	1	333.4	1
	3	305.5	2.4	334.4	1	334.4	1	307.9	1.9	334.4	1	334.4	1
	4	307.8	2.3	335.4	1	335.4	1	309.8	1.9	335.4	1	335.4	1
	5	309.9	2.1	336.4	1	336.4	1	311.6	1.8	336.4	1	336.4	1

extra RA only triggers one additional transplant (the one brought by the RA itself), indicating that the overall benefit of allowing RAs in those settings is limited. If one had to rank the impact of each feature based on the outcomes of these experiments, it appears that allowing chains results in the highest number of extra transplants, whereas considering longer cycles, longer chains, and RAs all seem to have an effect with the same order of magnitude. Note, however, that not all these features pose the same level of logistical and legal challenges in a kidney exchange programme. These conclusions differ from the ones reported by Andersson and Kratz (2020) in the context of KEP-HCA, as the latter found that the inclusion of HCAs was more impactful than the inclusion of chains or longer cycles. Such a difference could be due to the fact that the chain size was limited to 2 in the experiments of Andersson and Kratz 2020, whereas the budget was unlimited.

As a final remark, we point out that these experiments showed that each unit increase in the number of allowed RAs had a diminishing return in terms of extra number of transplants they enabled (which, this time, is in line with the findings of Aziz et al. 2021 in the context of KEP-HCA). One could postulate that this finding is true for every KEP-RA instance. However, we present in Figure 6 an instance with $n = 5$ and $K = 3$ for which this is not the case: with no RAs (in the left part of the figure), an optimal solution consists of cycle $B \rightarrow C \rightarrow B$ for a total of 2 transplants; with one RA (in the middle), an optimal solution consists of cycles $B \rightarrow C \rightarrow B$ and E for a total of 3 transplants, and with two RAs (in the right part of the figure), an optimal solution consists of cycles $B \rightarrow A \rightarrow B$ and $E \rightarrow C \rightarrow D \rightarrow E$ for a total of 5 transplants. In this instance, the first RA enables one extra transplant whereas the second RA enables two extra transplants.

Figure 6: A KEP-RA instance for which the RAs do not have a diminishing return



8 Conclusion

We studied the Kidney Exchange Problem with Reserve Arcs (KEP-RA), an extension of the well known Kidney Exchange Problem (KEP) in which a fixed number of transplants that are deemed incompatible can be made feasible (e.g., with immunosuppressants). We introduced and empirically tested various modelling strategies for the problem, and showed that our best approaches, which use the fact that there always exists an optimal KEP-RA solution where each cycle contains at most one Reserve Arc (RA), could solve instances with up to 1,000 recipient-donor pairs. We also discussed how non-directed donors could be taken into account in the proposed approaches. Finally, we measured the number of additional transplants enabled by the presence of RAs and outlined that the latter was particularly impactful in small size instances and in instances without non-directed donors. We also showed that each supplementary RA had a diminishing return in the general case, but that one could also construct artificial KEP-RA instances for which this was not the case. As future work, we plan to study KEP with half-compatible arcs and KEP with costs, two generalizations of KEP-RA. Another research direction involves analyzing how concepts such as stability, robustness, and the multi-agent setting, which were recently considered for KEP, could be extended to KEP-RA.

Acknowledgements

David Manlove was supported by the Engineering and Physical Sciences Research Council, grant number EP/X013618/1. The computational resources were provided by SURF Research Cloud through the pilot grant framework.

References

Abraham, D., Blum, A., and Sandholm, T. (2007). Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 295–304. ACM.

Andersson, T. and Kratz, J. (2020). Pairwise kidney exchange over the blood group barrier. *The Review of Economic Studies*, 87:1091–1133.

Aziz, H., Cseh, A., Dickerson, J., and McElfresh, D. (2021). Optimal kidney exchange with immunosuppressants. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 21–29.

Berger, A., Bonifaci, V., Grandoni, F., and Schäfer, G. (2011). Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128:355–372.

Bikbov et al., B. (2020). Global, regional, and national burden of chronic kidney disease, 1990-2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 395:709–733.

Biró, P., Haase-Kromwijk, B., Andersson, T., Ásgeirsson, E. I., Baltesová, T., Boletis, I., Bolotinha, C., Bond, G., Böhmig, G., Burnapp, L., Cechlárová, K., Ciaccio, P. D., Fronek, J., Hadaya, K., Hemke, A., Jacquelinet, C., Johnson, R., Kieszek, R., Kuypers, D. R., Leishman, R., Macher, M.-A., Manlove, D., Menoudakou, G., Salonen, M., Smeulders, B., Sparacino, V., Spieksma, F. C. R., Valentín, M. O., Wilson, N., and van der Klundert, J. (2019). Building kidney exchange programmes in Europe–an overview of exchange practice and activities. *Transplantation*, 103:1514–1522.

Biró, P., van de Klundert, J., Manlove, D., Pettersson, W., Andersson, T., Burnapp, L., Chromy, P., Delgado, P., Dworczak, P., Haase, B., Hemke, A., Johnson, R., Klimentova, X., Kuypers, D., Costa, A. N., Smeulders, B., Spieksma, F., Valentín, M. O., and Viana, A. (2020). Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research*, 291:447–456.

Blom, D., Hojny, C., and Smeulders, B. (2024). Cutting plane approaches for the robust kidney exchange problem. *Computers & Operations Research*, 162:106470.

Carvalho, M. and Lodi, A. (2023). A theoretical and computational equilibria analysis of a multi-player kidney exchange program. *European Journal of Operational Research*, 305:373–385.

Constantino, M., Klimentova, X., Viana, A., and Rais, A. (2013). New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research*, 231:57–68.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., and Pettersson, W. (2024). New algorithms for hierarchical optimisation in kidney exchange programs. *Operations Research*, 72(4):1654–1673.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., Pettersson, W., and Trimble, J. (2022). Improved instance generation for kidney exchange programmes. *Computers & Operations Research*, 141:105707.

Delorme, M., Manlove, D., and Smeets, T. (2023). Half-cycle: A new formulation for modelling kidney exchange problems. *Operations Research Letters*, 51:234–241.

Dickerson, J., Manlove, D., Plaut, B., Sandholm, T., and Trimble, J. (2016). Position-indexed formulations for kidney exchange. In *Proceedings of EC '16: the 17th ACM Conference on Economics and Computation*, pages 25–42. ACM.

Duan, R., Pettie, S., and Su, H.-H. (2018). Scaling algorithms for weighted matching in general graphs. *ACM Trans. Algorithms*, 14:1–35.

Heo, E., Hong, S., and Chun, Y. (2021). Kidney exchange with immunosuppressants. *Economic Theory*, 72:1–19.

Janssen, V. (2022). Matching problems with an additional knapsack constraint. Master's thesis, Eindhoven University of Technology. Available from `https://research.tue.nl/en/studentTheses/matching-problems-with-an-additional-knapsack-constraint` (accessed 23 July 2024).

Klimentova, X., Alvelos, F., and Viana, A. (2014). A new branch-and-price approach for the kidney exchange problem. In *Computational Science and Its Applications–ICCSA 2014: 14th International Conference, Guimarães, Portugal, June 30–July 3, 2014, Proceedings, Part II 14*, pages 237–252. Springer.

Klimentova, X., Biró, P., Viana, A., Costa, V., and Pedroso, J. (2023). Novel integer programming models for the stable kidney exchange problem. *European Journal of Operational Research*, 307:1391–1407.

MacMillan, S., Hosgood, S., and Nicholson, M. (2023). Enzymatic blood group conversion of human kidneys during ex vivo normothermic machine perfusion. *British Journal of Surgery*, 110:133–137.

Mak-Hau, V. H. (2017). On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *Journal of Combinatorial Optimization*, 33:35–59.

Manlove, D. and O'Malley, G. (2015). Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *ACM Journal of Experimental Algorithmics*, 19:1–21.

National Kidney Foundation (accessed 19 June 2024). Blood tests for transplant. `https://www.kidney.org/atoz/content/BloodTests-for-Transplant`.

Riascos-Álvarez, L., Bodur, M., and Aleman, D. (2024). A branch-and-price algorithm enhanced by decision diagrams for the kidney exchange problem. *Manufacturing & Service Operations Management*, 26:485–499.

Roth, A., Sönmez, T., and Ünver, M. (2005). Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151–188.

Roth, A., Sönmez, T., and Ünver, M. (2007). Efficient kidney exchange: Coincidence of wants in a market with compatibility-based preferences. *American Economic Review*, 97(3):828–851.

Roth, A., Sönmez, T., Ünver, M., Delmonico, F., and Saidman, S. (2006). Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *American Journal of Transplantation*, 6(11):2694–2705.

Sönmez, T., Ünver, M., and Yılmaz, O. (2018). How (not) to integrate blood subtyping technology to kidney exchange. *Journal of Economic Theory*, 176:193–231.

APPENDIX

A Proof of Theorem 1

Let us start by showing that KEP $\propto$ KEP-RA. Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ and K is an instance of KEP. One can easily build a corresponding KEP-RA instance $\mathcal{G}' = (\mathcal{V}', \mathcal{A}' = \{\mathcal{A}'_s \cup \mathcal{A}'_r\})$, K', and B' by setting $\mathcal{V}' = \mathcal{V}$, $\mathcal{A}'_s = \mathcal{A}$, $\mathcal{A}'_r = \emptyset$, $K' = K$ and $B' = 0$. In other words, a KEP instance can be seen as a KEP-RA instance in which the budget is set to 0.

Let us continue by showing that KEP-RA $\propto$ KEP-HCA. Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{A} = \{\mathcal{A}_s \cup \mathcal{A}_r\})$, K, and B is an instance of KEP-RA. One can easily build a corresponding KEP-HCA instance $\mathcal{G}' = (\mathcal{V}', \mathcal{A}' = \{\mathcal{A}'_s \cup \mathcal{A}'_h \cup \mathcal{A}'_i\})$, K', and B' by setting $\mathcal{V}' = \mathcal{V}$, $\mathcal{A}'_s = \mathcal{A}_s$, $\mathcal{A}'_h = \mathcal{A}_r$, $\mathcal{A}'_i = \emptyset$, $K' = K$, and $B' = B$. In other words, a KEP-RA instance can be seen as a KEP-HCA instance in which the set of infeasible arcs is empty.

Let us finish by showing that KEP-HCA $\propto$ KEP-C. Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{A} = \{\mathcal{A}_s \cup \mathcal{A}_h \cup \mathcal{A}_i\})$, K, and B is an instance of KEP-HCA. One can easily build a corresponding KEP-C instance $\mathcal{G}' = (\mathcal{V}', \mathcal{A}')$, c'_a $(a \in \mathcal{A}')$, K', and B' by setting $\mathcal{V}' = \mathcal{V}$, $\mathcal{A}' = \mathcal{A}_s \cup \mathcal{A}_h \cup \mathcal{A}_i$, $c'_a = 0$ $(a \in \mathcal{A}_s)$, $c'_a = 1$ $(a \in \mathcal{A}_h)$, $c'_a = B + 1$ $(a \in \mathcal{A}_i)$, $K' = K$, and $B' = B$. In other words, a KEP-HCA instance can be seen as a KEP-C instance in which the feasible arcs have cost 0, the half-compatible arcs have cost 1, and the infeasible arcs have cost $B + 1$.

B Proof of Theorem 2

Let us first recall that, given a non-directed graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ and a vector cost c' that associates each edge $e \in \mathcal{E}'$ to cost c'_e, a minimum cost perfect matching $\mathcal{M}'$ of $\mathcal{G}'$ can be computed in polynomial time (Duan et al. 2018). Such a problem is referred to as MCPM hereafter. Given $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, c, $K = 2$, and B an instance of KEP-HCA, we build a corresponding MCPM instance by:

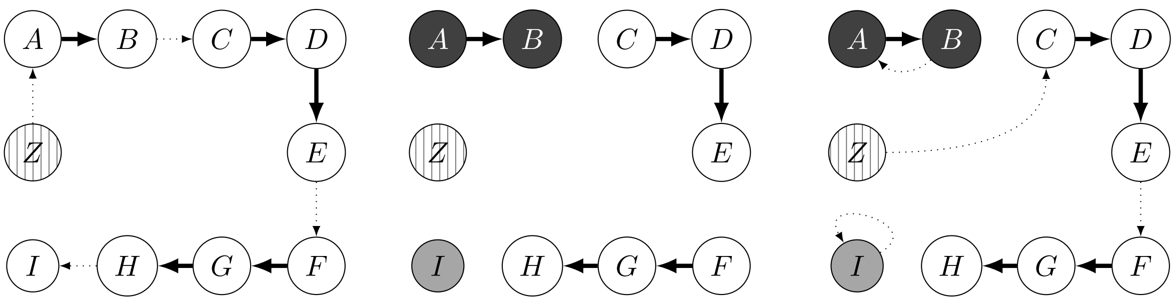
- Setting $\mathcal{V}$' = $\mathcal{V}'_o \cup \mathcal{V}'_d$ where $\mathcal{V}'_o = \mathcal{V}$ is the original set of vertices and $\mathcal{V}'_d$ is a set of $|\mathcal{V}|$ dummy nodes, one for each vertex in $\mathcal{V}$, to account for the fact that a vertex may be matched with itself (allowing for compatible pairs). Note that there are therefore always an even number of nodes in $\mathcal{V}'$;
- Setting $\mathcal{E}' = \{\{p_1, p_2\} \mid p_1, p_2 \in \mathcal{V}' \text{ and } p_1 \neq p_2\}$ with cost c'_{p_1,p_2} equals to:
  - $c_{p_1,p_2} + c_{p_2,p_1}$ if $p_1, p_2 \in \mathcal{V}'_o$,
  - c_{p_1,p_1} if $p_1 \in \mathcal{V}'_o, p_2 \in \mathcal{V}'_d$, and p_2 is the dummy node associated with p_1,
  - $B + 2$ if $p_1 \in \mathcal{V}'_o, p_2 \in \mathcal{V}'_d$, and p_2 is not the dummy node associated with p_1,
  - 0 if $p_1, p_2 \in \mathcal{V}'_d$.

After finding an optimal solution for the resulting MCPM instance, say with objective value z', one can derive the minimum budget $B' = z'$ that is needed to match all $|\mathcal{V}|$ recipient-donor pairs. If $B' \leq B$, then an optimal solution for KEP-C was found. Otherwise, we know for a fact that it is impossible to include all $|\mathcal{V}|$ recipient-donor pairs in the solution and $|\mathcal{V}| - 1$ becomes a valid upper bound for the KEP-C instance. We therefore transform the MCPM instance such that one original vertex and its associated dummy node are now allowed to not be included in the solution. To permit this, we add (i) one vertex v to set $\mathcal{V}'_o$ together with an edge $\{p, v\}$ with cost 0 for every vertex $p \in \mathcal{V}'_o \setminus \{v\}$ to $\mathcal{E}$' and (ii) one vertex w to set $\mathcal{V}'_d$ together with an edge $\{p, w\}$ with cost 0 for every vertex $p \in \mathcal{V}'_d \setminus \{w\}$ to $\mathcal{E}$'. After finding an optimal solution for the resulting MCPM instance, one can derive the minimum budget B' that is needed to match all but one recipient-donor pair. If $B' \leq B$, then an optimal solution for KEP-C was found. Otherwise, we know for a fact that it is impossible to include $|\mathcal{V}| - 1$ recipient-donor pairs in the solution and $|\mathcal{V}| - 2$ becomes a valid upper bound for the KEP-C instance. We therefore transform the MCPM instance such that two original vertices and their associated dummy nodes are now allowed to not be included in the solution and we iterate the procedure until $B' \leq B$. As the total number of iterations is bounded by $|\mathcal{V}|$, the overall complexity of the resulting routine is polynomial.

C Proof of Theorem 4

Let us consider an optimal KEP-RA solution s in which at least one chain, say c, contains at least one RA that is followed by $K-1$ or fewer standard arcs (see the left part of Figure 7, where RAs are represented as dotted arcs). After removing all the RAs from c, one obtains a set of k weakly connected components, each of those either being composed of $v \geq 2$ vertices linked by a set of $v-1$ arcs forming a directed path, or consisting of a single vertex (see the middle part of Figure 7). We group in $\Pi_{\geq K}$ the components that contain K vertices or more and we group in $\Pi_{<K}$ the remaining components. Following our assumption, it holds that $|\Pi_{<K}| \geq 1$. One can then add an arc to each of the $|\Pi_{<K}|$ weakly connected components composed of $K-1$ vertices or fewer so as to obtain $|\Pi_{<K}|$ cycles, either by linking the last and first vertices of the component's directed path if it has two vertices or more, or by adding a self-loop if the component consists of a single vertex. One finishes by linking the $\Pi_{\geq K}$ remaining components together to form a new chain (see the right part of Figure 7). This equivalent KEP-RA solution s' is feasible because: (i) it satisfies the budget constraint as it contains no more RAs than solution s, as the solution was constructed by removing k RAs and adding k arcs, which are either standard arcs or RAs, (ii) it satisfies the capacity and cardinality constraints as each new cycle generated is of size at most K and the new chain generated is of size at most the size of c. Finally, observe that since solution s' includes the exact same recipient-donor pairs as solution s, it is therefore also optimal.

Figure 7: Transforming a chain that includes an RA followed by $K-1$ standard arcs or fewer in a KEP-RA instance where $K=2$ and $L=9$



D Description of reduced-cost variable fixing

Reduced-cost variable fixing is a technique arising from linear optimization that is used to reduce the number of variables that needs to be considered in an ILP model. In the context of KEP, Delorme et al. (2024, 2023) showed that using reduced-cost variable fixing was empirically beneficial when using the CF, PICEF, PIEF, and HCF models. To apply reduced-cost variable fixing for KEP-RA, one first solves the LP-relaxation of one of the four models considered, saves the objective value, say $\hat{z}$, and uses it to derive a valid upper bound $U = \lfloor \hat{z} \rfloor$ on the optimal solution value. From now on, we are only looking for an integer solution with objective value U. Delorme et al. (2024) showed that any LP solution in which a variable v with reduced cost $\hat{s}_v \leq U - \hat{z} - \epsilon$ takes value 1 or above must have objective value strictly below U (ϵ is a very small number used to avoid precision errors). Therefore, no integer solution where $x_v = 1$ can have objective value U, meaning that x_v can be deactivated (i.e., set to 0 or not created at all). If we find a solution with value U for the reduced model, then that solution is optimal. If that is not the case, then it means that no solution with objective value U exists, so $U-1$ becomes a valid upper bound. U is thus decremented and all the variables are reactivated except the ones with reduced cost smaller than or equal to the updated $U - \hat{z} - \epsilon$ value. The algorithm is iterated until a solution with value U is found (which may occur immediately after updating the upper bound if the solution obtained at the previous iteration had value $U-1$ for example).

E Supplementary tables

Table 6: Results of the tested models in configurations 1RA, PICORA, and PICORA+RCVF for instances with $K = 3$

n	B	CF						HCF						EEF						PIEF					
		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF	
		#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)
50	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0
	1	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0
	2	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0
	3	20	0	20	0	20	0	20	0	20	0	20	0	20	0.1	20	0	20	0	20	0	20	0	20	0
	4	20	0	20	0	20	0	20	0	20	0	20	0	20	0.1	20	0	20	0	20	0	20	0	20	0
	5	20	0	20	0	20	0	20	0	20	0	20	0	20	0.1	20	0	20	0	20	0	20	0	20	0
100	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0	20	0
	1	20	0	20	0.1	20	0	20	0.2	20	0.1	20	0	20	0.8	20	0.1	20	0	20	0.3	20	0.1	20	0.1
	2	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.8	20	0	20	0	20	0.3	20	0.1	20	0.1
	3	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.8	20	0	20	0	20	0.3	20	0.1	20	0.1
	4	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.9	20	0	20	0	20	0.3	20	0.1	20	0.1
	5	20	0.1	20	0	20	0	20	0.2	20	0	20	0	20	0.9	20	0.1	20	0	20	0.3	20	0.1	20	0.1
200	0	20	0	20	0	20	0	20	0.1	20	0.1	20	0	20	0.1	20	0.1	20	0.1	20	0.4	20	0.4	20	0.3
	1	20	0.4	20	0.2	20	0.1	20	1.8	20	0.2	20	0.1	20	52.4	20	0.3	20	0.2	20	2.9	20	0.6	20	0.5
	2	20	1.1	20	0.2	20	0.1	20	1.8	20	0.2	20	0.1	20	53.4	20	0.3	20	0.2	20	2.7	20	0.6	20	0.4
	3	20	1.1	20	0.2	20	0.1	20	1.8	20	0.2	20	0.1	20	54.3	20	0.3	20	0.2	20	2.6	20	0.5	20	0.5
	4	20	1.1	20	0.2	20	0.1	20	1.9	20	0.2	20	0.2	20	136.3	20	0.3	20	0.2	20	2.7	20	0.5	20	0.5
	5	20	1.1	20	0.2	20	0.1	20	1.9	20	0.2	20	0.2	20	90.4	20	0.3	20	0.2	20	3.1	20	0.6	20	0.5
400	0	20	0.3	20	0.3	20	0.2	20	0.4	20	0.3	20	0.3	20	0.7	20	0.7	20	0.5	20	3	20	2.9	20	2.8
	1	20	9.4	20	1	20	0.5	20	19.5	20	1.2	20	0.6	2	3525.1	20	1.7	20	0.8	20	33.2	20	4	20	3.1
	2	20	9.4	20	0.9	20	0.5	20	18.9	20	1.1	20	0.7	3	3452.3	20	1.7	20	1	20	30.6	20	4	20	3.2
	3	20	8.9	20	1	20	0.5	20	18.4	20	1.1	20	0.7	2	3488.3	20	1.8	20	0.9	20	29.8	20	4.1	20	3.2
	4	20	8.8	20	1	20	0.6	20	19.2	20	1.2	20	0.8	5	3128.1	20	1.8	20	1	20	30.9	20	4.1	20	3.3
	5	20	8.8	20	1	20	0.6	20	18	20	1.1	20	0.8	6	3034.5	20	1.9	20	1	20	32.4	20	4.1	20	3.3

Table 7: Average number of variables and constraints (in thousands) for the tested models in configurations 1RA, PICORA, and PICORA+RCVF for instances with $K = 3$

n	B	CF						HCF						EEF						PIEF					
		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF	
		#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const
50	0	0	0.1	0	0.1	0	0.1	0.1	0.1	0	0.1	0	0.1	0	0.2	0	0.1	0	0.1	0	0.1	0	0.1	0	0.1
	1	1	0.1	0.6	0.1	0.2	0.1	1.2	0.3	0.6	0.2	0.2	0.1	1.3	0.5	0.6	0.2	0.2	0.2	1.5	0.6	0.6	0.2	0.2	0.2
	2	1	0.1	0.6	0.1	0.2	0.1	1.2	0.3	0.6	0.2	0.2	0.1	1.3	0.5	0.6	0.2	0.2	0.2	1.5	0.6	0.6	0.2	0.2	0.2
	3	1	0.1	0.6	0.1	0.2	0.1	1.2	0.3	0.6	0.2	0.2	0.1	1.3	0.5	0.6	0.2	0.2	0.2	1.5	0.6	0.6	0.2	0.2	0.2
	4	1	0.1	0.6	0.1	0.2	0.1	1.2	0.3	0.6	0.2	0.2	0.1	1.3	0.5	0.6	0.2	0.2	0.2	1.5	0.6	0.6	0.2	0.2	0.2
	5	1	0.1	0.6	0.1	0.2	0.1	1.2	0.3	0.6	0.2	0.2	0.1	1.3	0.5	0.6	0.2	0.2	0.2	1.5	0.6	0.6	0.2	0.2	0.2
100	0	0.1	0.1	0.1	0.1	0	0.1	0.3	0.2	0.2	0.2	0.1	0.2	0.3	0.5	0.3	0.4	0.2	0.3	0.3	0.3	0.3	0.3	0.2	0.3
	1	6.9	0.1	2.2	0.3	0.5	0.2	8.2	1.6	2.3	0.4	0.6	0.3	8.9	2.2	2.4	0.6	0.6	0.5	9.5	2.7	2.4	0.5	0.6	0.4
	2	6.9	0.1	2.2	0.3	0.5	0.2	8.2	1.6	2.3	0.4	0.6	0.3	8.9	2.2	2.4	0.6	0.6	0.5	9.5	2.7	2.4	0.5	0.6	0.4
	3	6.9	0.1	2.2	0.3	0.5	0.2	8.2	1.6	2.3	0.4	0.6	0.3	8.9	2.2	2.4	0.6	0.6	0.5	9.5	2.7	2.4	0.5	0.6	0.4
	4	6.9	0.1	2.2	0.3	0.5	0.2	8.2	1.6	2.3	0.4	0.6	0.3	8.9	2.2	2.4	0.6	0.6	0.5	9.5	2.7	2.4	0.5	0.6	0.4
	5	6.9	0.1	2.2	0.3	0.5	0.2	8.2	1.6	2.3	0.4	0.6	0.3	8.9	2.2	2.4	0.6	0.6	0.5	9.5	2.7	2.4	0.5	0.6	0.4
200	0	1.1	0.2	1.1	0.2	0.3	0.2	1.9	0.8	1.7	0.8	0.8	0.7	1.9	1.5	1.9	1.3	1.1	1.2	2	1.2	2	1.2	1.2	1.1
	1	52.3	0.2	8.9	0.6	2.1	0.5	57.8	7.4	9.5	1.2	2.5	1	62.6	10	9.7	1.7	2.9	1.5	65.3	13.4	9.7	1.6	3	1.4
	2	52.3	0.2	8.9	0.6	2.2	0.5	57.8	7.4	9.5	1.2	2.6	1	62.6	10	9.7	1.7	3	1.5	65.3	13.4	9.7	1.6	3	1.4
	3	52.3	0.2	8.9	0.6	2.4	0.5	57.8	7.4	9.5	1.2	2.9	1	62.6	10	9.7	1.7	3.3	1.5	65.3	13.4	9.7	1.6	3.3	1.4
	4	52.3	0.2	8.9	0.6	2.4	0.5	57.8	7.4	9.5	1.2	2.8	1	62.6	10	9.7	1.7	3.1	1.5	65.3	13.4	9.7	1.6	3.2	1.4
	5	52.3	0.2	8.9	0.6	1.9	0.5	57.8	7.4	9.5	1.2	2.4	1	62.6	10	9.7	1.7	2.8	1.5	65.3	13.4	9.7	1.6	2.7	1.4
400	0	7.6	0.4	7.6	0.4	2.7	0.4	11.4	3.8	11	3.8	5.4	3	12.4	6.1	12.4	5.7	7.1	4.9	12.7	5.6	12.7	5.6	7	5
	1	387.2	0.4	36.9	1.2	2.8	0.4	410.5	36.8	40.3	4.6	5.4	3	441	46.6	41.7	6.5	7.1	4.9	452.8	66	42	6.4	7.1	5
	2	387.2	0.4	36.9	1.2	5.3	0.6	410.5	36.8	40.3	4.6	7.6	3.3	441	46.6	41.7	6.5	9.4	5.2	452.8	66	42	6.4	9.4	5.2
	3	387.2	0.4	36.9	1.2	4.7	0.5	410.5	36.8	40.3	4.6	7.4	3.2	441	46.6	41.7	6.5	9.1	5	452.8	66	42	6.4	9.1	5.1
	4	387.2	0.4	36.9	1.2	6.2	0.6	410.5	36.8	40.3	4.6	8.9	3.4	441	46.6	41.7	6.5	10.9	5.3	452.8	66	42	6.4	10.6	5.3
	5	387.2	0.4	36.9	1.2	6.2	0.6	410.5	36.8	40.3	4.6	8.8	3.3	441	46.6	41.7	6.5	10.9	5.2	452.8	66	42	6.4	10.7	5.3

Table 8: Average number of variables and constraints (in thousands) for the tested models in configurations 1RA, PICORA, and PICORA+RCVF for instances with $K = 4$

n	B	CF						HCF						EEF						PIEF					
		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF		1RA		PICORA		PICORA+RCVF	
		#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const
50	0	0.1	0.1	0.1	0.1	0	0.1	0.1	0.1	0.1	0.1	0	0.1	0.1	0.2	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	1	3.5	0.1	0.8	0.2	0.1	0.1	2.9	0.4	0.9	0.2	0.1	0.2	2.5	0.6	0.9	0.3	0.2	0.2	3.2	0.9	0.9	0.2	0.2	0.2
	2	3.5	0.1	0.8	0.2	0.2	0.1	2.9	0.4	0.9	0.2	0.2	0.2	2.5	0.6	0.9	0.3	0.2	0.3	3.2	0.9	0.9	0.2	0.2	0.2
	3	3.5	0.1	0.8	0.2	0.2	0.2	2.9	0.4	0.9	0.2	0.2	0.2	2.5	0.6	0.9	0.3	0.3	0.3	3.2	0.9	0.9	0.2	0.3	0.2
	4	3.5	0.1	0.8	0.2	0.2	0.2	2.9	0.4	0.9	0.2	0.2	0.2	2.5	0.6	0.9	0.3	0.3	0.3	3.2	0.9	0.9	0.2	0.2	0.2
	5	3.5	0.1	0.8	0.2	0.3	0.2	2.9	0.4	0.9	0.2	0.3	0.2	2.5	0.6	0.9	0.3	0.4	0.3	3.2	0.9	0.9	0.2	0.4	0.2
100	0	0.7	0.1	0.7	0.1	0.1	0.1	0.9	0.3	0.8	0.3	0.3	0.3	1	0.7	1	0.6	0.6	0.6	1.1	0.6	1.1	0.6	0.6	0.6
	1	47.4	0.1	3.8	0.4	0.6	0.3	30.3	1.9	3.9	0.6	0.7	0.4	24.4	2.8	4	0.9	1.2	0.7	30.7	5.1	4.2	0.9	0.9	0.7
	2	47.4	0.1	3.8	0.4	0.7	0.3	30.3	1.9	3.9	0.6	0.9	0.5	24.4	2.8	4	0.9	1.3	0.8	30.7	5.1	4.2	0.9	1.1	0.8
	3	47.4	0.1	3.8	0.4	0.7	0.3	30.3	1.9	3.9	0.6	0.9	0.5	24.4	2.8	4	0.9	1.3	0.8	30.7	5.1	4.2	0.9	1.1	0.8
	4	47.4	0.1	3.8	0.4	1.3	0.3	30.3	1.9	3.9	0.6	1.5	0.5	24.4	2.8	4	0.9	2	0.8	30.7	5.1	4.2	0.9	1.8	0.8
	5	47.4	0.1	3.8	0.4	1.5	0.4	30.3	1.9	3.9	0.6	1.7	0.5	24.4	2.8	4	0.9	2	0.8	30.7	5.1	4.2	0.9	1.9	0.8
200	0	11.5	0.2	11.5	0.2	3.4	0.2	9.7	2	9.5	2	4.1	1.5	10.2	3.5	10.2	3.3	6.8	3	11.4	3.9	11.4	3.9	5.7	3.3
	1	741.5	0.2	23.1	0.8	6.1	0.5	334.5	9.3	21.1	2.6	6.7	1.8	251.5	12.7	21.8	3.9	9.3	3.3	307.5	26.9	22.9	4.5	8.1	3.6
	2	741.5	0.2	23.1	0.8	8	0.6	334.5	9.3	21.1	2.6	8.4	2	251.5	12.7	21.8	3.9	10.9	3.4	307.5	26.9	22.9	4.5	9.8	3.8
	3	741.5	0.2	23.1	0.8	8	0.6	334.5	9.3	21.1	2.6	8.4	2	251.5	12.7	21.8	3.9	11	3.4	307.5	26.9	22.9	4.5	10	3.8
	4	741.5	0.2	23.1	0.8	8.2	0.6	334.5	9.3	21.1	2.6	8.6	2	251.5	12.7	21.8	3.9	11.6	3.4	307.5	26.9	22.9	4.5	10.2	3.8
	5	741.5	0.2	23.1	0.8	8.9	0.6	334.5	9.3	21.1	2.6	9.3	2	251.5	12.7	21.8	3.9	11.9	3.4	307.5	26.9	22.9	4.5	10.9	3.9
400	0	159.1	0.4	159.1	0.4	58.3	0.4	100	13.2	99.6	13.2	49.7	10.1	100.3	19.2	100.3	18.8	70.5	16.6	109.2	23	109.2	23	58.4	19.7
	1	10,713.9	0.4	202.9	1.6	69.2	0.8	3447.8	45.3	143.4	14.4	60.3	10.6	2538.5	56.4	144.1	20	79.9	17	3066.5	133.2	153	24.2	69	20.2
	2	10,713.9	0.4	202.9	1.6	73.5	1	3447.8	45.3	143.4	14.4	64.6	10.8	2538.5	56.4	144.1	20	83.6	17.2	3066.5	133.2	153	24.2	73.2	20.4
	3	10,713.9	0.4	202.9	1.6	73.5	1	3447.8	45.3	143.4	14.4	64.6	10.8	2538.5	56.4	144.1	20	83.6	17.2	3066.5	133.2	153	24.2	73.2	20.4
	4	10,713.9	0.4	202.9	1.6	73.5	1	3447.8	45.3	143.4	14.4	64.6	10.8	2538.5	56.4	144.1	20	83.6	17.2	3066.5	133.2	153	24.2	73.2	20.4
	5	10,713.9	0.4	202.9	1.6	74.5	1	3447.8	45.3	143.4	14.4	65.6	10.8	2538.5	56.4	144.1	20	84.6	17.3	3066.5	133.2	153	24.2	74.3	20.5

Table 9: Results of the tested models in configuration PICORA+RCVF for large instances

n	B	CF+PICORA+RCVF				HCF+PICORA+RCVF				EEF+PICORA+RCVF				PIEF+PICORA+RCVF			
		$K=3$		$K=4$		$K=3$		$K=4$		$K=3$		$K=4$		$K=3$		$K=4$	
		#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)	#opt	T(s)
600	0	20	0.9	20	48.6	20	1.2	20	48.9	20	2.7	20	602.3	20	10.4	20	138.4
	1	20	2.1	20	63.2	20	2.6	20	58.4	20	3.7	20	695.9	20	11.7	20	161.5
	2	20	2.3	20	66	20	2.7	20	61.4	20	4.3	20	606.9	20	11.8	20	146
	3	20	2.4	20	68.1	20	2.9	20	64.7	20	4.4	20	531.4	20	12	20	145.6
	4	20	2.3	20	64.3	20	2.9	20	59.2	20	4.3	20	504.1	20	12	20	159.7
	5	20	2.5	20	65.7	20	3.1	20	63	20	4.9	20	632.2	20	12.3	20	151.8
800	0	20	2.4	20	112.5	20	3.7	20	137.1	20	9	13	2883.3	20	26.1	20	594.8
	1	20	4.6	20	143.3	20	5.8	20	153.1	20	12.1	7	3308	20	28.4	20	486.5
	2	20	5.2	20	134	20	6.9	20	160.8	20	13.4	7	3079.5	20	29.6	20	521
	3	20	5.5	20	154.9	20	7.2	20	168	20	13.7	11	2829.9	20	29.7	20	590.6
	4	20	5.1	20	180.1	20	7.2	20	167.1	20	13.6	8	3259.5	20	30.2	20	547.3
	5	20	5.4	20	173.1	20	6.9	20	176.5	20	14	9	2944.1	20	30.4	20	610.6
1000	0	20	5.3	20	317.6	20	7.5	20	387.9	20	27.7	0	3600	20	55.4	20	978
	1	20	9.6	20	394.1	20	12.6	20	493.1	20	32.8	0	3600	20	61.1	17	1481.1
	2	20	11.3	20	384.3	20	16.9	20	469.2	20	38.5	0	3600	20	63.8	18	1382.8
	3	20	11.6	20	421.7	20	16.1	20	455.5	20	35.2	0	3600	20	62.9	17	1419.5
	4	20	11.2	20	405.5	20	15.6	20	478.9	20	36.2	0	3600	20	62.7	19	1323.5
	5	20	10.9	20	428.3	20	16	20	533.8	20	36.2	0	3600	20	63	17	1408.1

Table 10: Average number of variables and constraints (in thousands) for CF+PICORA+DPICEF+RCVF and CF+PICORA+IPICEFTH4+RCVF for large instances with chains

n	B	CF+PICORA+DPICEF+RCVF								CF+PICORA+IPICEFTH4+RCVF							
		$K=3$				$K=4$				$K=3$				$K=4$			
		$L=3$		$L=6$		$L=4$		$L=8$		$L=3$		$L=6$		$L=4$		$L=8$	
		#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const	#var	#const
667	0	37.9	1.9	123.7	3.9	407.7	2.6	522.6	5.2	37.9	1.9	125.5	3.9	407.7	2.6	525.1	5.2
	1	195.2	2.9	785.6	4.9	756.4	4.2	1548.8	6.8	74.4	2.9	161.9	4.9	469.6	4.2	587.2	6.8
	2	197.9	2.9	789.6	4.9	760.5	4.2	1552.8	6.8	74.4	2.9	161.9	4.9	469.6	4.2	587.2	6.8
	3	197.9	2.9	789.6	4.9	760.5	4.2	1552.8	6.8	74.4	2.9	161.9	4.9	469.6	4.2	587.2	6.8
	4	197.9	2.9	789.6	4.9	760.5	4.2	1552.8	6.8	74.4	2.9	161.9	4.9	469.6	4.2	587.2	6.8
	5	197.9	2.9	789.6	4.9	760.5	4.2	1552.8	6.8	74.4	2.9	161.9	4.9	469.6	4.2	587.2	6.8
889	0	77	2.5	232.8	5.2	1254	3.4	1462.2	7	77	2.5	235	5.2	1254	3.4	1465.5	7
	1	360.1	3.9	1424.5	6.5	1884.1	5.5	3308.3	9.1	143.5	3.9	301.5	6.5	1366.8	5.5	1578.4	9.1
	2	361.9	3.9	1427.2	6.5	1886.8	5.5	3311	9.1	143.5	3.9	301.5	6.5	1366.8	5.5	1578.4	9.1
	3	361.9	3.9	1427.2	6.5	1886.8	5.5	3311	9.1	143.5	3.9	301.5	6.5	1366.8	5.5	1578.4	9.1
	4	361.9	3.9	1427.2	6.5	1886.8	5.5	3311	9.1	143.5	3.9	301.5	6.5	1366.8	5.5	1578.4	9.1
	5	361.9	3.9	1427.2	6.5	1886.8	5.5	3311	9.1	143.5	3.9	301.5	6.5	1366.8	5.5	1578.4	9.1
1111	0	138.5	3.2	386.3	6.5	3188.1	4.3	3519.2	8.8	138.5	3.2	388.7	6.5	3188.1	4.3	3522.9	8.8
	1	584.1	4.8	2243.3	8.2	4175.8	6.9	6393.5	11.4	243.8	4.8	494	8.2	3367.4	6.9	3702.3	11.4
	2	585.4	4.8	2245.3	8.2	4177.8	6.9	6395.4	11.4	243.8	4.8	494	8.2	3367.4	6.9	3702.3	11.4
	3	585.4	4.8	2245.3	8.2	4177.8	6.9	6395.4	11.4	243.8	4.8	494	8.2	3367.4	6.9	3702.3	11.4
	4	585.4	4.8	2245.3	8.2	4177.8	6.9	6395.4	11.4	243.8	4.8	494	8.2	3367.4	6.9	3702.3	11.4
	5	585.4	4.8	2245.3	8.2	4177.8	6.9	6395.4	11.4	243.8	4.8	494	8.2	3367.4	6.9	3702.3	11.4