Individual Fairness in Kidney Exchange Programs

Golnoosh Farnadi[1, 2, 3 *], **William St-Arnaud**[1 *], **Behrouz Babaki**[4*], **Margarida Carvalho**[1]

[1] CIRRELT and Université de Montréal, [2] HEC Montréal, [3] Mila, [4] Polytechnique Montréal

Abstract

Kidney transplant is the preferred method of treatment for patients suffering from kidney failure. However, not all patients can find a donor that matches their physiological characteristics. Kidney exchange programs (KEPs) seek to match such incompatible patient-donor pairs together, usually with the main objective of maximizing the total number of transplants. Since selecting one optimal solution translates to a decision on who receives a transplant, it has a major effect on the lives of patients. The current practice in selecting an optimal solution does not necessarily ensure fairness in the selection process. In this paper, the existence of multiple optimal plans for a KEP is explored as a mean to achieve individual fairness. We propose the use of randomized policies for selecting an optimal solution in which patients' equal opportunity to receive a transplant is promoted. Our approach gives rise to the problem of enumerating all optimal solutions, which we tackle using a hybrid of constraint programming and linear programming. The advantages of our proposed method over the common practice of using the optimal solution obtained by a solver are stressed through computational experiments. Our methodology enables decision makers to fully control KEP outcomes, overcoming any potential bias or vulnerability intrinsic to a deterministic solver.

1 Introduction

Chronic kidney disease is an incurable condition that leads to a slow loss of kidney function. The worldwide population of patients suffering from this condition reached 700 million in 2017 (Bikbov et al. 2020) and is increasing at an annual rate of around 6% (Fresenius Medical Care 2018). These patients require a renal replacement therapy: kidney transplantation or dialysis. The former is generally the preferable treatment because it reduces the economic burden of dialysis, it has the potential to improve the patient life quality and to yield longer lifespan.

A complicating factor in transplantation is that many patients have donors which are not medically compatible with them. *Kidney Exchange Programs (KEPs)* allow such patients to exchange donors. These programs have been implemented in several countries such as South Korea (Park et al. 2004), The Netherlands (De Klerk et al. 2005), United Kingdom (Manlove and O'Malley 2012), and Canada (Malik and Cole 2014). The underlying idea in KEP is to form cycles: a donor donates a kidney if their corresponding patient receives one. This motivates the problem of computing exchange plans that maximize the overall patients benefit, which is usually tackled by Integer Programming approaches (Roth, Sönmez, and Ünver 2007; Abraham, Blum, and Sandholm 2007; Klimentova, Alvelos, and Viana 2014; Dickerson et al. 2016). The typical objective is to maximize the number of transplants. However, some KEPs (like the Dutch Program) use additional hierarchical criteria (Glorie 2014), while others associate weights to reflect the benefit and priority of an exchange (UNOS 2020).

KEP is an example of algorithmic decision making with a significant impact on the lives of individuals. This makes it critical to ensure *fairness* in the process. To the best of our knowledge, the literature has exclusively focused on group fairness in KEPs, and the dominant perspective in this area is to ensure fairness towards pre-defined *groups* of patients (*e.g.* those deemed to be *hard-to-match*), leading to potential deterioration of the maximum number of transplants (Dickerson, Procaccia, and Sandholm 2014). There is no previous work that studies *individual fairness* in KEPs. Individual fairness (Dwork et al. 2012) ensures similar treatment for patients without grouping them *a priori*. This goal calls for a new methodology. After all, in any exchange plan, only some patients will receive a transplant. We address this problem by designing randomized policies for selecting a solution from a set of *desirable* solutions, such that individuals have the same *chances* of receiving a transplant.

Existing works suggest that KEPs frequently have many optimal solutions (Carvalho and Lodi 2019). Currently, the KEP solution computed by the algorithms in place is the one being carried out. Note that such a solution highly depends on the order in which the instance is inputted in the system and how the optimization is implemented. Such dependence is not controlled by the user and thus, it can deeply affect the evolution of the KEP pool (Canadian Blood Services 2014). Hence, solution selection has a significant impact on the lives of the patients and special care must be taken to ensure fairness in this process. Designing a randomized policy to select a solution from the set of optimal solutions allows us to enforce individual fairness without compromising optimality. However, constructing the set of all optimal solu-

[*] Equal Contribution

tions is a challenging task that requires new approaches to enumerate them.

This paper makes several contributions: 1) We introduce for the first time individual fairness as the supporting policy for solution selection in KEPs. We introduce several definitions of individual fairness in KEPs and propose randomized policies for enforcing them. 2) We present and evaluate alternative methods for enumerating the solutions of KEPs, using Integer Linear Programming (ILP) and Constraint Programming (CP). 3) We design a specialized CP propagator for efficient enumeration of solutions. 4) We present extensions of our method which allow to control the trade-off of fairness vs. scalability or optimality. 5) We evaluate the performance of our method through extensive experiments. We compare our method against the current practice of deploying the optimal solution returned by the solver. We also inspect the effect of individual fairness on a well-known group fairness criterion, that is, the welfare of *highly-sensitized* (hard-to-match) patients.

The rest of the paper is structured as follows: We introduce the KEP and review fairness in KEP in sections 2 and 3. After establishing the connection between individual fairness and solution enumeration, we introduce several methods for enumerating solutions in Section 4. The randomized policies for enforcing individual fairness are presented in Section 5. We evaluate our proposed approach in Section 6. We present conclusions and future directions in Section 7.

2 Kidney exchange program

In this section, the classic compatibility graph for KEP is described, together with the standard integer program that models the optimization of the patients' benefit.

A KEP instance can be represented as a graph $G = (V, A)$, where V is the union of the set of incompatible pairs P and the set of altruistic donors N, and A is the set of arcs representing compatibilities. There are two types of possible exchanges: *cycles* and *chains*. A cycle in G among vertices in P guarantees that the patient associated with a donor (donating a kidney) also receives a transplant. A chain is a path $(v_0, v_1, \ldots, v_K)$ in G starting in an altruistic donor $v_0 \in N$ and with $\{v_1, \ldots, v_K\} \subset P$. The donor of the last involved pair in a chain becomes an altruistic donor for the next KEP or he/she donates in the waiting list for deceased donation. In the right of Figure 1, $(1, 3, 2, 1)$ is a cycle of length 3 and $(5, 3, 4)$ is a chain of length 3. Selected chains and cycles must be disjoint since donors can only donate one kidney. In Europe, their lengths are typically limited to three or four while, *e.g.* in the US, there is no limit on chains length (Biró et al. 2019). Roth, Sönmez, and Ünver (2007) showed that most of the cycles potential is achieved by limiting their length to 3 and Dickerson, Procaccia, and Sandholm (2012) argued that there is no advantage on considering chains larger than 3. Therefore, based on the practical interest of cycles and chains of length at most 3 and for the sake of simplicity, in this paper, we concentrate on this case. Nevertheless, our framework can be generalized to any length limit, although large length bounds can result in slower running times.

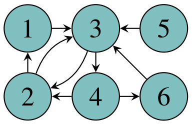

Figure 1: Compatibility graph for a KEP.

The goal of KEPs is to determine a set of disjoint chains and cycles representing the kidney exchanges to be performed such that the benefit of the patients receiving a transplant is maximized. ILPs have been broadly used to model KEPs: *e.g.* cycle formulation (Abraham, Blum, and Sandholm 2007; Roth, Sönmez, and Ünver 2007), edge formulation (Roth, Sönmez, and Ünver 2007), position-indexed formulation (Dickerson et al. 2016). The cycle formulation is the simplest to describe. Let $\mathcal{C}$ be the set of all allowed cycles and chains of G, and w_c for $c \in \mathcal{C}$, the benefit of the exchange c. Then, the formulation is the following:

$$\mathcal{P}(\mathcal{C}): \quad \max \quad \sum_{c \in \mathcal{C}} w_c x_c \tag{1a}$$

$$\text{s.t.} \quad \sum_{c: v \in c} x_c \leq 1 \qquad \forall v \in V \tag{1b}$$

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

The decision variables x_c take value 1 if c is a selected exchange and 0 otherwise. Constraints (1b) enforce each donor to participate in at most one exchange. Constraints (1c) introduce the binary requirement for x. The objective function (1a) maximizes the benefit of the selected exchanges. The main goal is to maximize the number of transplants ($w_c = |c|$). Additionally, KEPs frequently account for the optimization of other utilitarian and equity criteria. This is implemented either through hierarchical optimization or a match-point system establishing the value of w_c (Biró et al. 2017; Biró et al. 2019).

Definition 1. *The optimality criteria of a KEP is defined by a rank $\mathcal{W} = \left(\mathbf{w}^1, \ldots, \mathbf{w}^m\right)$ of $|\mathcal{C}|$-dimensional vectors. The optimal value of the optimality criteria is a list $\mathcal{P}^* = (OPT_1, \ldots, OPT_m)$ obtained through the hierarchical optimization of the linear objectives given by $\mathcal{W}$. A $|\mathcal{C}|$-dimensional vector x is said to be an optimal solution if it is feasible and $(w^i)^T x = OPT_i$, for $i = 1, \ldots, m$.*

The most commonly used utilitarian criteria are the number of transplants, the number of back-arcs on selected cycles and chains, and the number of cycles. See Appendix for an illustration.

Definition 2. *A $|\mathcal{C}|$-dimensional vector x is said to be a t-relaxed solution if it is feasible and $(w^i)^T x \geq OPT_i - t$, for $i = 1, \ldots, m$.*

The framework proposed in this paper takes the optimality criteria as input and proposes mechanisms for the selection of an optimal solution promoting individual fairness.

3 Related literature

As mentioned before, the main goal of KEPs is to maximize the benefit of the patients. The baseline is to associate weights that allow to maximize the number of transplants while prioritizing certain groups of patients. For instance, highly-sensitized patients have a low probability of being compatible with a random kidney. In this context, Dickerson, Procaccia, and Sandholm (2014) concentrate on the trade-off of moving from maximizing the number of transplants (utilitarian objective function) towards maximizing the number of highly-sensitized patients receiving a kidney. However, McElfresh, Bidkhori, and Dickerson (2018) show that such an approach can sacrifice efficiency significantly and thus propose the use of a threshold to balance group fairness and the number of transplants. Freedman et al. (2018) focus on the fact that such prioritization can depend on human values and use it to break ties between solutions achieving the maximum number of transplants. Anticipation of future kidney exchanges is taken into account in (Dickerson and Sandholm 2014) by assigning weights to certain exchanges, such as the ones involving highly-sensitized patients, and also assigning a chance of failure to arcs in a matching.

In (Klimentova et al. 2019) and (Biró et al. 2020), cross-border programs are considered. Instead of patient fairness, these works concentrate on fairness between countries, namely in terms of the contribution of each country to an international KEP pool. Likewise, *e.g.* (Sönmez and Ünver 2013; Ashlagi and Roth 2014; Carvalho and Lodi 2019) investigate multi-agent programs but through the lens of non-cooperative game theory.

The majority of current studies on fairness in KEPs focus on group fairness. The closest work to individual fairness is on egalitarian mechanisms seeking the so-called Lorenz-dominance[1](*e.g.*, (Roth, Sönmez, and Utku Ünver 2005; Li et al. 2014)), which, simply put, focuses on equalizing the patients individual matching probabilities. Our work is more general since it is not particularly tailored for exploring the mathematical structure of pairwise exchanges and we present a variety of fairness selection policies.

4 Individual fairness by enumeration

In many optimization problems, deploying different optimal solutions leads to different a distribution of benefits among some individuals. For example, when a KEP has multiple optimal solutions (in terms of the patients who receive transplants), by picking any solution we are inevitably favoring the individuals included in that solution over the excluded ones who appear in another solution.

Our purpose is to design a randomized process for selecting an optimal solution which promotes similar *chances* of receiving a transplant among the individuals. For a KEP instance $\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ with optimal value OPT, this process involves two steps: 1) Constructing the set of optimal solutions $\mathcal{S} = \{\mathbf{x} \in \mathcal{X} : f(x) = OPT\}$. 2) Assigning a weight to each solution in such a way that random selection according to those weights results in similar chances for the individuals. It is worth noting that if we are willing to pay a price in terms of optimality, we can relax the definition of the solution set by including the set of t-relaxed solutions.

The computationally-demanding part of this process is the first step, *i.e.* enumeration of the optimal solutions. Even in the simple forms of KEPs (exchanges of length 2 and optimizing for number of transplants), this task is known to be #P-complete (Valiant 1979). We will next study different methods for tackling this challenging problem.

In this section we propose three approaches based on ILP and CP methodologies. For the sake of simplicity, in what follows, we assume a single optimality criterion: the number of transplants (denoted by *OPT*).

4.1 Enumeration through no-good cuts in ILP

All optimal solutions (*i.e.*, $\mathcal{S}$) can be determined in an iterative way by solving the following problem in iteration $K+1$:

$$\sum_{c\in\mathcal{C}} |c| x_c = OPT \tag{2a}$$

$$\sum_{c\in\mathcal{C}:x_c^k=0} x_c + \sum_{c\in\mathcal{C}:x_c^k=1} (1-x_c) \geq 1 \quad k = 1,\ldots,K \tag{2b}$$

(1b) and (1c)

where x^k is an optimal solution in $\mathcal{S}$ determined in a previous iteration k. Constraint (2a) implies that the computed solution must be optimal. Constraints (2b) are the so-called *no-good cuts* (Balas and Jeroslow 1972) and, in order to be satisfied, the solution must differ in at least one entry for each x^k. Once Problem (2) becomes infeasible, then all optimal solutions have been determined. Since there is a finite number of feasible exchanges, this iterative method will stop in finite time. We implemented the no-good cuts using lazy constraints. In this way, the solver takes advantage of the search tree being built, avoiding to reset the search for each iteration.

An alternative to adding the no-good cuts is to systematically exhaust the search tree. This functionality is supported by constraint satisfaction paradigms such as CP.

4.2 Enumeration through CP

We will now present a CP model for the problem of enumerating the optimal solutions of KEP. A CP problem is a tuple $(\mathcal{V}, D, \mathcal{C})$ where $\mathcal{V}$ is a set of finite-domain variables, D specifies the set of values that each variable can take (*i.e.* their domains), and $\mathcal{C}$ is a set of constraints. Each constraint is defined over a subset of variables and dictates the valid assignments for that subset. A solution is an assignment to each variable from its domain such that all constraints are satisfied. A CP problem is solved by a combination of *search* and *propagation*. Search consists of assigning values to variables from their domains. Assigning a value to a variable triggers propagation by constraints which include that variable. This involves eliminating values from the domains of variables that cannot appear in a solution. Each constraint is equipped with a *propagator*, which is a specialized algorithm for performing those eliminations.

We will now present the CP formulation. First, to facilitate modeling, we slightly modify the KEP compatibility graph by adding a self-loop to every vertex. We define an array of variables X indexed by the vertices $v \in V$. The variable $X[v]$ represents the successor of v in some path, and its domain is defined as $\{v\} \cup \{u : (v,u) \in A\}$. A self-loop is represented by assigning v to $X[v]$. In Figure 1, the array X has 6 entries and domain of $X[v_3]$ is $\{v_2, v_3, v_4\}$. The

[1] A Lorenz-dominant policy is not guarantee to exist for KEPs considering exchanges larger than 2.

constraints of the CP model are as follows:

$$\textit{AllDifferent}(X) \tag{3a}$$

$$\big(X[v] = v\big) \vee \big(X[X[v]] = v\big) \vee \big(X[X[X[v]]] = v\big) \;\; \forall v \in V \tag{3b}$$

$$\sum_{v \in V} \big(X[v] \neq v\big) = OPT. \tag{3c}$$

The constraint *AllDifferent*(X) requires that all variables in X take different values. It is easy to show that under this constraint, the successor variables define a set of cycles (including self-loops). Observe that when a vertex appears in a self-loop, it is excluded from the matching. The constraint set (3b) enforces that each vertex is either in a self-loop or a cycle of length two or three. Finally, constraint (3c) ensures that the solution of the model is an optimal matching.

This formulation can be extended to KEP instances which include chains starting with an altruistic donor. One way to do this is to add a dummy vertex for each altruistic donor, with incoming arcs from all vertices in P and only one outgoing arc to its corresponding altruistic donor.

Algorithm 1: Class *KepConstraint*

```
method updateCycles()
    foreach $v \in V$ do
        mask ← 0
        foreach $u \in \Delta_v \setminus \{v\}$ do
            mask ← mask | contains[$v, u$]
        mask ← ∼ mask
        validCycles ← validCycles & mask
method filterDomains()
    foreach $v \in V$ do
        foreach $u \in dom(X[v]) \setminus \{v\}$ do
            if validCycles & contains[$v, u$] $= 0$ then
                $dom(X[v]) \leftarrow dom(X[v]) \setminus \{u\}$
method propagate()
    updateCycles()
    if validCycles $= 0$ then
        return Backtrack
    UB ← relaxedKEP()
    if UB < OPT then
        return Backtrack
    filterDomains()
```

4.3 A global constraint for KEP

We will now introduce a new constraint which we have designed specifically for KEP problems. This constraint will replace constraints (3a)-(3b) in the CP model. Using our knowledge of the problem structure, we introduce a propagator which shrinks the domains of the variables and prunes the search space more efficiently than the combination of simpler constraints.

The propagation mechanism is performed by the *KepConstraint* class. At initialization, all cycles with length two and three are generated, and the bitset contains$[v, u]$ is calculated for each arc $(u, v) \in A$. This bitset indicates the cycles that contain that arc. Figure 2 illustrates this on the running example. Another important data structure is the validCycles bitset. At any node of the search tree, this bitset maintains the cycles which can be part of the solution given the current domains of variables.

arc	c_1	c_2	c_3	c_4
$(1,3)$	1	0	0	0
$(2,1)$	1	0	0	0
$(2,3)$	0	1	1	0
$(3,2)$	1	0	0	0
$(3,4)$	0	0	1	1
		...		

c_1: $(1 - 3 - 2)$
c_2: $(2 - 3)$
c_3: $(2 - 3 - 4)$
c_4: $(3 - 4 - 6)$

Figure 2: The entries of the contains array for some of the arcs in the running example. Each entry is a bitset representing the cycles which contain the arc.

Algorithm 1 shows the main methods of the *KepConstraint* class. The propagation starts at the *propagate()* method. Note that removing the value u from the domain of $X[v]$ is equivalent to removing the arc (v, u). This in turn invalidates every cycle that contains this arc. In method *updateCycles*, we identify such cycles and set their bit in validCycles to zero. We assume that Δ_v stores the values removed from the domain of $X[v]$ since the last call to the propagator (at this node or upstream in the search tree). Next, in the method *relaxedKEP()*, we obtain an upper bound on the number of vertices covered by any solution given the current domains of variables. This bound is obtained by solving the linear relaxation of problem $\mathcal{P}(\mathcal{C})$ (1). Before solving this relaxation, we set the x_c variables corresponding to invalid cycles (according to validCycles bitset) to zero. Finally, the method *filterDomains()* checks for each arc (v, u) whether it is included in a valid cycle. If not, u is removed from the domain of $X[v]$.

Our propagator builds on the ideas and principles employed by the Compact-Table (CT) algorithm for filtering the *Table* constraint. Similar to CT, our propagator relies on the *reversible bitset* data structure which is optimized for backtracking search in CP solvers. We refer the reader to (Demeulenaere et al. 2016) for details.

4.4 Finding a covering set of solutions

For the sake of comparison and scalability, we describe a *greedy* approach. Here, we restrict enumeration of solutions to a *covering* $\mathcal{S}'$, *i.e.* a subset of $\mathcal{S}$ such that if $v \in P$ appears in some solution of $\mathcal{S}$, it also appears in $\mathcal{S}'$.

Next, we show how to adapt the CP methodology to generate a cover. Let V' denote the set of indices of vertices which are not yet covered by any solution during the search. After finding each solution, we update V' and add the following constraint $\bigvee_{v \in V'} X[v] \neq v$ which requires that the next solution contains some vertex which has not appeared in the previous solutions. We can also adapt the KEP constraint to this setting by adding the following constraint $\sum_{c\colon V' \cap c \neq \emptyset} x_c \geq 1$ to the linear relaxation after finding each solution.

5 Fair selection strategy

When a KEP instance has multiple solutions (such that they differ in terms of patients who receive transplants and ex-

δ		v_1	v_2	v_3	v_4	v_5	v_6
0.4	S_1	1	1	1	0	0	0
0.2	S_2	0	1	1	1	0	0
0.4	S_3	0	0	1	1	0	1
	p_δ	0.4	0.6	1	0.6	0	0.4

Table 1: Optimal solutions for the compatibility graph of Figure 1. The selection strategy (left) determines the vertex probabilities (bottom).

changes established), by picking any solution we are inevitably favoring the individuals included in the solution over the excluded individuals. Our proposal for enforcing fairness in this process is to design a randomized process for selecting an optimal solution which promotes equal *chances* of receiving a transplant among the patients. Next, we introduce a selection strategy that allows us to design a randomized process.

Definition 3. *Let $\mathcal{S}$ be the set of optimal solutions of a KEP instance. A* selection strategy *δ is a distribution over $\mathcal{S}$. The probability that the patient v receives a transplant according to the strategy δ is called a* vertex probability *and is denoted by $p_\delta(v) = \sum_{S:v\in S} \delta_S$.*

Table 1 shows a selection strategy for the running example. The common practice is to select the first optimal solution returned by a solver. We call this the *first-best* strategy. The strategy which assigns equal probabilities to all solutions (*i.e.* $\delta_S = \frac{1}{|\mathcal{S}|}$) is called the *uniform* strategy.

Intuitively, we prefer a selection strategy which yields similar chances for the patients. We can quantify this quality in terms of the dispersion between the vertex probabilities. An example is the *mean absolute deviation*, *i.e.* $\sum_v |p_\delta(v) - m|$ where $m = \frac{1}{|V|}\sum_v p_\delta(v)$. Alternatively, one can measure the quality of a strategy in terms of the vertex with the lowest probability, *i.e.* $\min_v p_\delta(v)$.

A strategy can be defined over a subset of solutions by assigning zero to every solution which is not included in the subset. This is the case for coverings. In Table 1, $\{S_1, S_3\}$ is a covering subset, while $\{S_1, S_2\}$ is not. Any strategy defined over the latter will reduce the chances of v_6 to zero.

5.1 Computing the strategies

Assuming that we have the set of optimal solutions, we are interested in obtaining the best strategy according to some criterion that reflects fairness. Next, we describe popular criteria that can drive the selection strategy.

Minimizing the L_p-norm. The goal is to determine the selection strategy δ such that it minimizes the L_p-norm mean deviation of each vertex. In other words, we aim to find the distribution δ that minimizes $\left(\sum_{v\in P} |p_\delta(v) - \frac{1}{|P|}\sum_{v\in P} p_\delta(v)|^p\right)^{\frac{1}{p}}$. Let variables δ_S and y_v denote the probabilities of solution S and vertex v. Let z represent the mean vertex probability and d_v represent the deviation of y_v from the mean. The optimal strategy is obtained by solving the following optimization program:

$$\min \left(\sum_{v\in P} d_v^p\right)^{\frac{1}{p}} \tag{4a}$$

$$\text{s.t.} \sum_{S\in\mathcal{S}} \delta_S = 1 \tag{4b}$$

$$y_v = \sum_{S:v\in S} \delta_S \qquad \forall v \in P \tag{4c}$$

$$\sum_{v\in P} y_v = |P| \cdot z \tag{4d}$$

$$d_v \geq y_v - z \quad \forall v \in P \tag{4e}$$

$$d_v \geq z - y_v \quad \forall v \in P \tag{4f}$$

$$0 \leq \delta_S \leq 1 \quad \forall S \in \mathcal{S} \tag{4g}$$

$$0 \leq y_v \leq 1 \quad \forall v \in P. \tag{4h}$$

Constraints (4b) and (4g) ensure that δ is a probability distribution over $\mathcal{S}$. Constraints (4c) and (4h) establish y_v equal to $p_\delta(v)$. Constraint (4d) makes z equal to the mean over y. Finally, Constraints (4e) and (4f) lead to $d_v \geq |y_v - z|$, which together with the minimization of the objective function implies $d_v = |y_v - z|$. Since minimizing L_1 and L_2 correspond to a linear and convex programs, respectively, we will use them in our experiments.

Maximizing the minimum vertex probability. In this case, the goal is to maximize the probability of the vertex with the least chance of being in a selected solution (Maxmin). Mathematically, it means to find the selection strategy δ that maximizes $\min_{v\in P}\sum_{S\in\mathcal{S}} \delta_S$. Next, we formulate the maximization of the mininum vertex probability as a linear program. Keeping the same definition as above for variable δ_S and letting variable z denote the probability of the least well-off vertex. The strategy which maximizes this smallest probability is obtained using the following formulation:

$$\max \quad z \tag{5a}$$

$$\text{s.t.} \quad z \leq \sum_{S:v\in S} \delta_S \qquad \forall v \in P \tag{5b}$$

$$\text{(4b) and (4g).}$$

Constraints (5b) enforce z to be upper bounded by the smallest probability for a vertex v, while the objective function (5a) maximizes z (thus, the lowest vertex probability).

5.2 Projection: removing redundancy

In Section 4, we present three approaches to determine all optimal solutions (*i.e.*, $\mathcal{S}$). Next, we show that $\mathcal{S}$ can be reduced without any lost in the optimality of the selection policies described in this section. To this end, we need the following definition characterizing *redundancy* among optimal solutions.

Definition 4. *Two solutions in $\mathcal{S}$ are said to be KEP-equivalent if they contain exactly the same set of transplanted patients. A set $\bar{\mathcal{S}} \subseteq \mathcal{S}$ is said to be a projection of $\mathcal{S}$, if no two solutions in $\bar{\mathcal{S}}$ are KEP-equivalent and for any solution in $\mathcal{S}$ there is a KEP-equivalent solution in $\bar{\mathcal{S}}$.*

If an optimal selection strategy δ^* for problems (4) or (5) assigns a strictly positive probability to some strategy $S \in \mathcal{S} \setminus \bar{\mathcal{S}}$, *i.e.* $\delta^*_S > 0$, then this value can be passed to $\delta^*_{\hat{S}}$ for some $\hat{S} \in \bar{\mathcal{S}}$ which is KEP-equivalent to S. Consequently, the following result holds, contributing to the scalability of the fairness selection policies described earlier (see Appedix for detailed proof):

Lemma 5.1. *Let $\bar{\mathcal{S}}$ be a projection of $\mathcal{S}$. The optimal value of problems* (4) *(L_p-norm) and* (5) *(Maxmin) is equal to their optimal value when restricted to $\bar{\mathcal{S}}$.*

In this way, we can adapt both our ILP and CP methodologies to restrict the enumeration to projections. Note that a projection is a covering, but the reverse does not hold.

6 Experimental evaluation

In this section, we show the effectiveness of our proposed methods by performing an empirical evaluation. For all the empirical evaluations that are presented in this section, we use two datasets which we name the *US* dataset (Saidman et al. 2006) and the *Canada* dataset (Carvalho and Lodi 2019). In both of our datasets, the number of graph vertices ranges over 6 values: 20, 30, . . . , 70. There are 50 different instances per graph size, amounting to a total of 600 different graphs in total (See Appendix for further details on our experimental setup and datasets used). We investigate five research questions in our experiments:

Q1: How scalable are our optimal solution enumeration approaches? We proposed four approaches to enumerate the optimal solutions. These include the no-good-cut approach of Section 4.1 (*MIP-lazy-cut*), the CP model of Section 4.2 (*CP-standard*), the CP model equipped with a specialized propagator as described in Section 4.3 (*CP-specialized*), and the CP approach for greedy covering (*CP-greedy*) of Section 4.4. The performance profiles of these methods are presented in Figure 3.

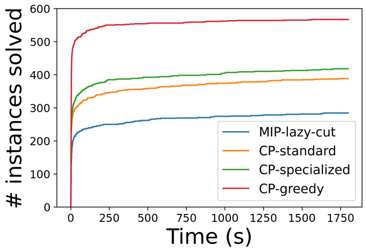

Figure 3: Performance profiles.

Among the three methods which enumerate all optimal solutions, *CP-specialized* performs the best and enumerates the solutions of graphs with as many as 70 vertices. This is a considerable improvement over the *MIP-lazy-cut* method which scales to graphs with at most 40 vertices. However, for half of the instances of larger graphs, enumerating all optimal solutions within the timeout (30 minutes) is still not possible. This is alleviated by enumerating a subset of solutions using the *CP-greedy* method. Figure 3 shows that this method successfully finds a solution set for the majority of instances within a short amount of time.

Q2: What is the effect of graph size on the number of optimal solutions? We inspect the optimal solutions obtained by *CP-specialized*: the average number of all solutions over graphs of different sizes is reported in the first row of Table 2. For both datasets, increasing the graph size is accompanied by a significant increase in the count of optimal solutions, which in turn correlates with the increased difficulty of the enumeration task. We observe that two different solutions can be KEP-equivalent, *i.e.*. the same set of patients receive transplants in different cycles. For this reason, in the second row of Table 2, we present the projected solutions. It is notable that even after removing the equivalent solutions, we could still have hundreds of thousands of optimal solutions. In the last row, we present the results when the utilitarian hierarchical criteria mentioned in Section 2 are used. It is interesting to observe that even after applying this filter, thousands of optimal solutions exist which emphasizes the importance of selection strategies such as those introduced in Section 5 (See Appendix for additional results).

Graph size	20	30	40	50	60	70
All (Canada)	35	2820	180155	363764	354277	616120
All (US)	256	8524	220181	352953	594217	788673
Projected (Canada)	12	30	92	658	1718	2750
Projected (US)	31	86	2522	6570	7001	10953
Hierarchical (Canada)	8	144	892	1487	3778	9318
Hierarchical (US)	76	495	54012	142137	62569	42478

Table 2: Average number of solutions per graph size.

Q3: To what extent does our individual fairness method enhance the equality of chances for receiving a transplant? The large number of solutions in Table 2 indicates the importance of employing a selection procedure to ensure fairness. To address this question, we compare the enumeration-based policies *uniform*, *Maxmin*, L_1 and L_2, with the *first-best* baseline. We use the enumeration approaches *CP-specialized* and *CP-greedy*. The results are summarized in the plots presented in Figure 4 (See Appendix for additional comparison among the measures).

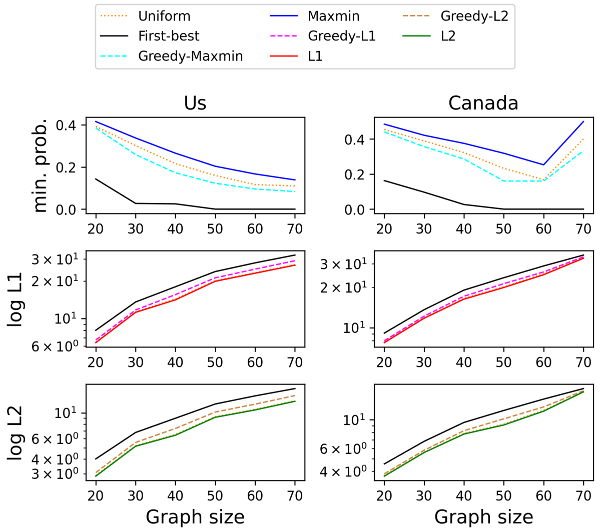

Figure 4: Comparing different selection strategies.

The two plots on the top compare different selection strategies with respect to the probability of the most disadvantaged vertex (higher is better) These probabilities are averaged over all solved instances per graph size (See Appendix for additional experiments that present the variances per graph sizes). It is observed that the enumeration-based strategies significantly improve the chances of the least well-off patients compared to the baseline (first-best), especially in larger graphs where this chance approaches zero.

The four plots on the bottom compare the strategies in terms of L_1 and L_2 norms (lower is better). Similar to Maxmin, significant improvement over the baseline is observed (Notice the logarithmic y axis). An interesting observation is the good performance of the uniform policy across multiple fairness criteria.

It is worth noting that we considered changing the solver's random seed to obtain different solutions, however we observed no variability in the returned solutions.

Q4: How does our individual fairness method compare with existing work? We first remark that comparison with the existent egalitarian mechanisms (Roth, Sönmez, and Utku Ünver 2005; Li et al. 2014) is not possible, since they 1) are restricted to cycles of length two, 2) use the Lorentz-dominance metric which is not guaranteed to exist for KEPs with cycles of length greater than 2, and 3) can only handle the number of transplants as the optimal criteria.

This leaves us with the group fairness approach which is defined in terms of the number of *hard-to-match patients* (*i.e.* PRA - panel reactive antibody - above 80%) included in a solution (Dickerson, Procaccia, and Sandholm 2014). To enforce group fairness according to this definition, we add the constraint $\sum_{c \in \mathcal{C}} h(c) x_c \geq \alpha^* \, |V_H|$ to Problem (1), where $V_H = \{v \in P : v \text{ has PRA } \geq 80\%\}$, $h(c) = |\{v \in c : v \in V_H\}|$, and α^* is the maximum possible fraction of hard-to-match patients that can receive a transplant among all solutions.

Graph size	20	30	40	50	60	70
Group fairness measure: α value						
First best	0.30±0.31	0.29±0.24	0.43±0.18	0.46±0.16	0.44±0.16	0.46±0.17
IF	0.31±0.30	0.29±0.23	0.42±0.17	0.45±0.17	0.44±0.15	0.49±0.18
GF	**0.37±0.32**	**0.34±0.23**	**0.52±0.17**	**0.54±0.16**	**0.49±0.15**	**0.58±0.17**
Optimality measure: number of transplants						
IF	**8.05±4.02**	**11.81±3.46**	**16.03 ±5.36**	**21.16±5.02**	**24.08±6.51**	**24.33±2.50**
GF	7.90±3.97	11.69±3.41	15.69±5.36	20.84±4.83	24.08±6.51	23.67±2.66
Individual fairness measure: Log L_2						
First best	4.02±0.95	6.77±0.90	8.91±1.34	11.73±1.06	13.76±1.48	15.80±0.78
IF	**2.86±0.97**	**5.18±1.04**	**6.50±1.34**	**9.22±1.67**	**10.38±1.52**	**12.11±1.40**
GF	4.01±0.95	6.76±0.89	8.84±1.41	11.71±1.09	13.76±1.48	15.58±0.85

Table 3: Comparing group fairness method (GF) with individual fairness method (IF), for US only.

In Table 3, we compare our individual fairness selector based on L_2-norm (IF) with *first-best* and the group fairness approach (GF). The table presents the expected value of the *group fairness measure* (α: the number of hard-to-match patients in a solution), the *optimally measure* (expected maximum number of transplants), and the *individual fairness measure* ($\log L_2$). We remark that the results of IF do not degrade the group fairness measure too much (α value). They are similar to the *first-best* case and even slightly better in some cases. We can see however that in terms of the individual fairness measure, our approach significantly outperforms the *first-best* and group fairness while we do not pay the *price of fairness* (Dickerson, Procaccia, and Sandholm 2014) wrt the total number of transplants (See Appendix for similar comparison between other individual fairness approaches, i.e., Maxmin and L_1).

Q5: What is the price of enforcing both individual and group fairness?

We use the enumeration of t-relaxed solutions to control the price of including *hard-to-match* patients. The main challenge is that the number of such solutions can become quite large, *e.g.*, the number of solutions for a graph of size 40 expanded from 164 to 213,250 solutions. Thus, we used CP-greedy to determine a covering of t-relaxed solutions. Figure 5 presents these results for graphs of size 70 (See Appendix for similar results based on the smaller graphs). We observe that in all instances the 3-relaxed solutions include all patients in the pool. Interestingly, by paying a minimum cost, *i.e.*, OPT-1, not only we are able to cover 98-99% of all patients in the pool, we also cover 98-99% of the *hard-to-match* patients. This result suggests that we are able to enforce both individual fairness and group fairness by paying a minimum cost. Another interesting observation is that by using our approach, we are able to give the equal chance of receiving a transplants to $\%12$ more patients that were not in the optimal solutions while only 1% of them are labeled as hard-to-match.

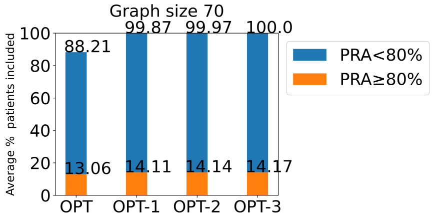

Figure 5: Effect of t-relaxation on the (relative) number of hard-to-match patients in solutions.

7 Conclusion & future directions

This work investigates for the first time the problem of individual fairness on KEPs. To balance patients' likelihood of being in a selected solution, we propose a two-phases framework: *(1)* the enumeration of solutions not deteriorating the utilitarian goals of KEPs, and *(2)* the determination of a probability distribution among those solutions that optimizes an individual fairness metric. For the first phase, 4 techniques are developed using integer and constraint programming. Whereas the second phase presents selection strategies. The computational experiments demonstrate the crucial role of our selection strategies in ensuring fairness among patients. As a byproduct, our approach *(i)* guarantees full control over selected solutions by decision makers, instead of relying on a solver's output, and *(ii)* it does not significantly damage group fairness. In fact, for the latter, a small relaxation of the utilitarian criteria can allow our selection strategies to better incorporate group fairness. Investigating other approaches that are able to scale to larger graphs remains a promising direction for future work. Likewise, another interesting path to explore is on how our approach behaves over time: How much will it avoid the accumulation of the most-disadvantaged patients and reduce the overall health cost? Does individual fairness remains stable?

Ethical impact

Kidney exchange programs are running in several countries. Their *modus operandi*, such as frequency, optimization criteria and maximum length for chains, varies from country to country. Nevertheless, the methodology described in this work can be incorporated on top of the current practices as all it requires is a description of the valid exchange plans (solutions). Furthermore, this work contributes to our responsibility as computer scientists to empower users (in this case, physicians) with a complete control of the selected solution, *i.e.*of the patients receiving a transplant.

It is worth highlighting the fact that our selection strategies promoting individual fairness do not use an inputted medical definition. Contrary to group fairness, which is defined for patients satisfying certain characteristics independent of the patient-donor pool, the individual fairness introduced in this paper depends on the patient-donor pool and not explicitly on patients' attributes. This observation opens an important dialog with practitioners about the identification of the most disadvantaged patients by our mathematical procedure and hence, on the importance of context (KEP pool) for this definition.

Gao (2019) has raised attention for priority rules involving patients in a critical situation (high mortality rate). Additionally, over time, the health status of patients can worsen. These are elements that cannot be mathematically captured by considering static KEPs. Therefore, it is of the utmost importance to further investigate the impact of individual fairness on KEP pools over time and to adapt our enumeration procedures to capture dynamic solutions (*i.e.*, exchange plans over time). In this way, we can better anticipate and control the impact of algorithms used by kidney exchange programs.

Last but not least, healthcare is a protected domain under the anti-discrimination law. However, there is no comprehensive guidance for healthcare research and subsequent deployment yet, and studies such as ours are an opportunity to provide insight into fairness in KEP problems to facilitate regulating models.

References

Abraham, D. J.; Blum, A.; and Sandholm, T. 2007. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *EC*, 295–304. ACM.

Ashlagi, I., and Roth, A. E. 2014. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics* 9(3):817–863.

Balas, E., and Jeroslow, R. 1972. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics* 23(1):61–69.

Bikbov, B.; Purcell, C.; Levey, A.; Smith, M.; Abdoli, A.; Abebe, M.; Adebayo, O.; Afarideh, M.; Agarwal, S.; Agudelo-Botero, M.; Ahmadian, E.; Al-Aly, Z.; Alipour, V.; Almasi-Hashiani, A.; Al-Raddadi, R.; Alvis Guzmán, N.; Amini, S.; Andrei, T.; and Andrei, C. 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(10225):709 – 733.

Biró, P.; Burnapp, L.; Haase, B. J.; Hemke, A.; Johnson, R.; van de Klundert, J.; and Manlove, D. 2017. Kidney exchange practices in Europe. *First Handbook of the COST Action CA15210: European Network for Collaboration on Kidney Exchange Programmes (ENCKEP).*

Biró, P.; Gyetvai, M.; Klimentova, X.; Pedroso, J. a. P.; Pettersson, W.; and Viana, A. 2020. Compensation scheme with Shapley value for multi-country kidney exchange programmes. *In: Proceedings of the 34th International ECMS Conference on Modelling and SimulationECMS 2020. European Council for Modelling and Simulation (ECMS).*

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.; Nanni Costa, A.; Smeulders, B.; Spieksma, F.; Valentín, M. O.; and Viana, A. 2019. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*.

Canadian Blood Services. 2014. Donation and transplantation kidney paired donation program data report 2009 - 2013. https://professionaleducation.blood.ca/sites/msi/files/ Canadian-Blood-Services-KPD-Program-Data-Report-2009-2013.pdf.

Carvalho, M., and Lodi, A. 2019. Game theoretical analysis of kidney exchange programs. *arXiv preprint arXiv:1911.09207.*

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(1):57–68.

De Klerk, M.; Keizer, K. M.; Claas, F. H.; Witvliet, M.; Haase-Kromwijk, B. J.; and Weimar, W. 2005. The dutch national living donor kidney exchange program. *American Journal of Transplantation* 5(9):2302–2305.

Demeulenaere, J.; Hartert, R.; Lecoutre, C.; Perez, G.; Perron, L.; Régin, J.; and Schaus, P. 2016. Compact-table: Efficiently filtering table constraints with reversible sparse bit-sets. In *CP*.

Dickerson, J. P., and Sandholm, T. 2014. Balancing efficiency and fairness in dynamic kidney exchange. In *AAAI Workshop: Modern Artificial Intelligence for Health Analytics*, volume WS-14-08 of *AAAI Workshops*. AAAI.

Dickerson, J. P.; Manlove, D. F.; Plaut, B.; Sandholm, T.; and Trimble, J. 2016. Position-indexed formulations for kidney exchange. *CoRR* abs/1606.01623.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2012. Optimizing kidney exchange with transplant chains: theory and reality. In *AAMAS*, 711–718. IFAAMAS.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2014. Price of fairness in kidney exchange. In *AAMAS*, 1013–1020.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.

Freedman, R.; Borg, J. S.; Sinnott-Armstrong, W.; Dickerson, J. P.; and Conitzer, V. 2018. Adapting a kidney exchange algorithm to align with human values. In *AIES*, 115.

Fresenius Medical Care. 2018. Annual report 2018 - care and live.

Gao, I. 2019. Fair matching in dynamic kidney exchange.

Glorie, K. 2014. *Clearing Barter Exchange Markets: Kidney Exchange and Beyond*. Ph.D. Dissertation, Erasmus University Rotterdam.

Klimentova, X.; Alvelos, F. P.; and Viana, A. 2014. A new branch-and-price approach for the kidney exchange problem. In *ICCSA (2)*, volume 8580 of *Lecture Notes in Computer Science*, 237–252. Springer.

Klimentova, X.; Viana, A.; ao Pedro Pedroso, J.; and Santos, N. 2019. Fairness models for multi-agent kidney exchange programmes. working paper.

Li, J.; Liu, Y.; Huang, L.; and Tang, P. 2014. Egalitarian pairwise kidney exchange: Fast algorithms vialinear programming and parametric flow. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, 445–452. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Malik, S., and Cole, E. 2014. Foundations and principles of the canadian living donor paired exchange program. *Canadian journal of kidney health and disease* 1(6).

Manlove, D., and O'Malley, G. 2012. Paired and altruistic kidney donation in the UK: algorithms and experimentation. In *SEA*, volume 7276 of *Lecture Notes in Computer Science*, 271–282. Springer.

McElfresh, D. C.; Bidkhori, H.; and Dickerson, J. P. 2018. Scalable robust kidney exchange. *CoRR* abs/1811.03532.

Park, K.; Lee, J.; Huh, K.; Kim, S.; and Kim, Y. 2004. Exchange living-donor kidney transplantation: diminution of donor organ shortage. *Transplantation proceedings* 36(10):2949–295.

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

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

Saidman, S.; Roth, A.; Sonmez, T.; Unver, U.; and Delmonico, F. 2006. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation* 81(5):773–782.

Sönmez, T., and Ünver, M. U. 2013. Market design for kidney exchange. In *The Handbook of Market Design*. Oxford University Press.

UNOS. 2020. OPTN: Organ procurement and transplantation network. Policies. https://optn.transplant.hrsa.gov/media/1200/optn_policies.pdf. Last visited 03/08/2020.

Valiant, L. 1979. The complexity of computing the permanent. *Theoretical Computer Science* 8(2):189 – 201.

Acknowledgments

This work was partially funded by FRQ-IVADO Research Chair in Data Science for Combinatorial Game Theory, and NSERC grant 2019-04557. Behrouz Babaki was supported by a postdoctoral scholarship from IVADO through the Canada First Research Excellence Fund (CFREF) grant.

This research was enabled in part by support provided by Calcul Québec (www.calculquebec.ca) and Compute Canada (www.computecanada.ca).

Appendix

A Hierarchical optimization

Three of the most used optimality criteria for kidney exchange programs (Biró et al. 2019) are the number of patients receiving a transplant, the number of back-arcs on the selected cycles and chains, and the number of cycles. Note that these are utilitarian criteria. In accordingly with the notation introduced in definition 1, we have $\mathcal{W} = (\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3)$ with $\mathbf{w}^1 = (|c_1|, \ldots, |c_k|)$, $\mathbf{w}^2 = (c_1^b, \ldots, c_k^b)$, and $\mathbf{w}^3 = (1, \ldots, 1)$, for $c_i \in \mathcal{C}$ and c_i^b equal to the number of back-arcs for cycle c_i.

Recall our running example of Figure 1 and respective cycles' labels in Figure 2. For sake of simplicity, we remove the altruistic donor (vertex 5). In this case, we have $\mathbf{w}^1 = (3, 2, 3, 3)$, $\mathbf{w}^2 = (1, 0, 1, 0)$ and $\mathbf{w}^3 = (1, 1, 1, 1)$. Starting by maximizing the first criterion (number of patients receiving a kidney), we obtain $OPT_1 = 3$ with cycles c_1 and c_4 as two distinct solutions leading to this optimal value. Next, we maximize the number of back-arcs while keeping $OPT_1 = 3$; thus, $OPT_2 = 1$ with cycle c_1 selected. Finally, we maximize the number of cycles while maintaining $OPT_1 = 3$ and $OPT_2 = 1$; then, $OPT_3 = 1$ with only cycle c_1 satisfying these criteria.

B Proof of Lemma 5.1

Lemma. *Let $\bar{\mathcal{S}}$ be a projection of $\mathcal{S}$. The optimal value of problems* (4) *(L_p-norm) and* (5) *(Maxmin) is equal to their optimal value when restricted to $\bar{\mathcal{S}}$.*

Proof. We make the proof for problem (4). The other is analogous. The restriction of problem (4) to $\bar{\mathcal{S}}$ is reflected by a decrease in the dimension of the decision vector δ. Therefore, the restricted problem provides an upper bound for problem (4). Next, we show how to build a feasible solution to the restricted problem from an optimal solution of problem (4), and therefore, establish the lemma result. For the sake of clarity, let us rename δ in the restricted problem by $\bar{\delta}$. Denote by (δ^*, y^*, z^*) the optimal solution of problem (4). In its restricted version, we must have $\bar{\delta}_S = 0$ for all $S \in \mathcal{S} \setminus \bar{\mathcal{S}}$. Hence, if there is $S \in \mathcal{S} \setminus \bar{\mathcal{S}}$ such that $\delta_S^* > 0$, we make $\bar{\delta}_{\hat{S}} = \delta_{\hat{S}}^* + \delta_S^*$ for the $\hat{S} \in \bar{\mathcal{S}}$ which is KEP-equivalent to S. Note that by construction, this step is always feasible. It is easy to see that $(\bar{\delta}, y^*, z^*)$ is feasible for the restricted problem. Thus, the restricted problem has the same optimal value as problem (4). □

C Experimental Setup

Our experiments were performed using an Intel E5-2683 2.1 GHz processors. The memory limit was set to 8 GB, and the time limit was 30 minutes. The MIP models were solved using Gurobi-8.1.1[2]. We implemented the CP models and propagators using OscarR-4.1.0[3]. The linear relaxations in the propagation algorithm were solved using Cplex-12.9[4]. Our code for all the experiments is available at [5].

D Datasets

We used two datasets which we refer to them by the *US* dataset and the *Canada* dataset.

The *US* dataset. The literature typically uses the US population characteristics presented in (Saidman et al. 2006) for simulating KEP instances. In (Constantino et al. 2013) such instances were generated for various graph sizes and we use them in our computational analysis[6].

The *Canada* dataset. We use the generator described in (Carvalho and Lodi 2019) to create instances for our experiments. This generator is based on data from the Canadian Kidney Paired Donation Program and it is restricted to patient-donor pairs (no altruistic donors are generated).

E Additional Experiments for Section 6

Additional experiments for Q2. What is the effect of graph size on the number of optimal solutions? As the graph size (number of verices) increases, more optimal solutions are expected to exist. Table 4 presents the average number of optimal solutions for different graph sizes when using the enumeration of projected hierarchical (with the criteria described in Appendix A), 3-relaxed solutions and 3-relaxed solutions using CP-Greedy approach.

Note that for a given instance, the set of projected hierarchical solutions is contained in the set of 3-relaxed solutions. We were able to fully enumerate all projected hierarchical solutions in Table 4. However, note that given the potentially very large set of 3-relaxed solutions, the number of instances for which we were able to perform their full enumeration is lower than the number of instances for which we present the results in Table 2. This explains why the average number of all solutions for some graph sizes in Table 2 is higher than the corresponding numbers in 3-relaxed enumeration. In particular, the complete enumeration of 3-relaxed solutions was not possible for any graph of size 70.

In the last row of the table, we present the average number of 3-relaxed solutions using CP-Greedy approach in which we only include a minimal set of solutions that covers all the patients in the pool. Using this approach, we were able to enumerate solutions of all instances in our datasets.

Graph size	20	30	40	50	60	70
Projected Hierarchical (Canada)	4	9	28	152	77	686
Projected Hierarchical (US)	17	50	1030	1446	15701	5525
3-relaxed (Canada)	997	17592	67402	226676	25586	N/A
3-relaxed (US)	2950	37592	92988	94703	3106	N/A
Greedy 3-relaxed (Canada)	4	6	7	9	10	12
Greedy 3-relaxed (US)	6	9	13	16	17	20

Table 4: Average number of solutions per graph size.

[2] https://www.gurobi.com

[3] https://bitbucket.org/oscarlib/oscar

[4] https://www.ibm.com/products/ilog-cplex-optimization-studio

[5] https://github.com/stawaway/IF-KEP

[6] Available at https://rdm.inesctec.pt/dataset/ii-2019-001

Additional experiments for Q3. To what extent does our individual fairness method enhance the equality of chances for receiving a transplant? Figure 6 provides the standard deviation for our selection strategies and first-best, complementing Figure 4. Figure 7 enables us to compare our different individual fairness selection strategies, demonstrating the L_2-norm selection strategy as the more robust among all the used fairness metrics.

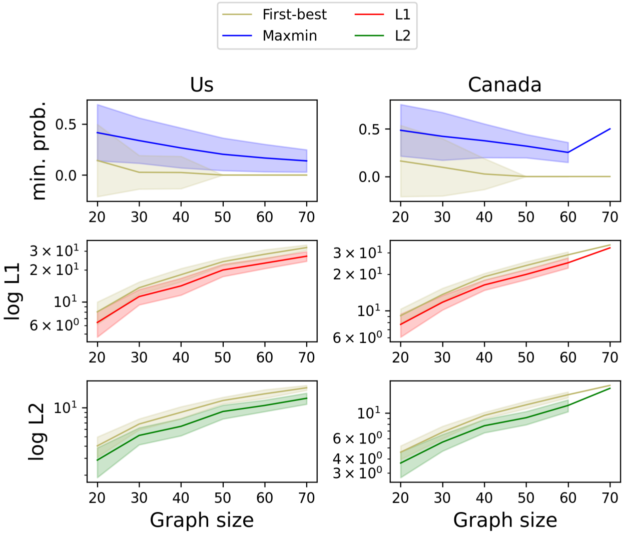

Figure 6: Comparison between individual fairness methods and the first-best case for both datasets over various losses (shaded area represents std).

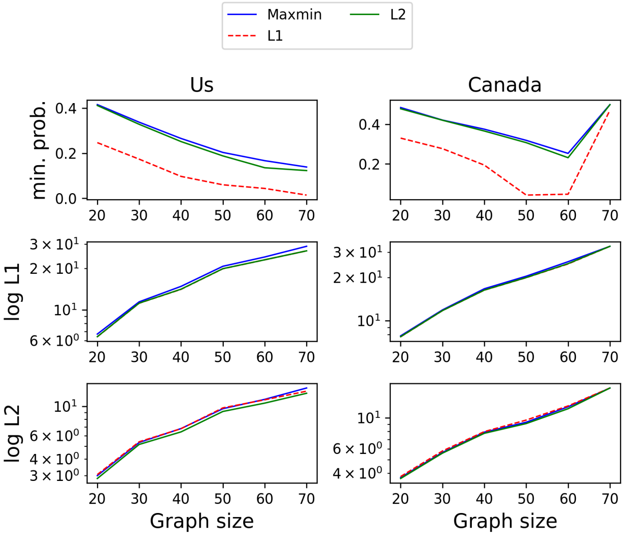

Figure 7: Comparison between the different individual fairness methods for both datasets over various losses.

Additional experiments for Q4. How does our individual fairness method compare with existing work? Table 5 complements Table 3 by adding α value for the Uniform, Maxmin and L_1-norm selection strategies. Similarly, Table 5 complements Table 3 by comparing GF with IF methods based on minimum probability and log L_1-norm.

The results are similar to those presented in Table 3 for all IF methods based on different losses. Also, our IF approaches outperform GF and best-first approaches with respect to both IF measures, *i.e.*, log L_1 and minimum probability.

Note that we only had access to PRA of patients in the US dataset to compare our approach with the group fairness approach.

Graph size	20	30	40	50	60	70
Group fairness measure: α value						
First best	0.30±0.31	0.29±0.24	0.43±0.18	0.46±0.16	0.44±0.16	0.46±0.17
IF (Uniform)	0.31±0.30	0.29±0.23	0.41±0.18	0.45±0.17	0.43±0.16	0.49±0.19
IF (Maxmin)	0.30±0.30	0.29±0.23	0.41±0.17	0.44±0.17	0.43±0.16	0.48±0.19
IF (L_1-norm)	0.31±0.29	0.29±0.23	0.42±0.16	0.44±0.17	0.43±0.16	0.49±0.19
IF (L_2-norm)	0.31±0.30	0.29±0.23	0.42±0.17	0.45±0.17	0.44±0.15	0.49±0.18
GF	**0.37±0.32**	**0.34±0.23**	**0.52±0.17**	**0.54±0.16**	**0.49±0.15**	**0.58±0.17**
Individual fairness measure: minimum probability						
First best	0.14±0.35	0.03±0.17	0.03±0.17	0.00±0.00	0.00±0.00	0.00±0.00
IF (L_1-norm)	**0.42±0.28**	**0.34±0.23**	**0.27±0.21**	**0.22±0.18**	**0.14±0.09**	**0.11±0.09**
GF	0.21±0.42	0.11±0.32	0.06±0.24	0.11±0.32	0.08±0.28	0.00±0.00
Individual fairness measure: Log L_1						
First best	8.04±1.90	13.54±1.80	17.82±2.69	23.45±2.11	27.53±2.96	31.60±1.56
IF (Maxmin)	**6.39±1.73**	**11.20±1.86**	**14.14±2.67**	**19.81±2.95**	**22.72±2.72**	**26.10±2.31**
GF	8.02±1.90	13.52±1.79	17.68±2.82	23.42±2.18	27.53±2.96	31.16±1.70

Table 5: Comparing group fairness method (GF) with individual fairness method (IF), for US only.

Additional experiments for Q5. What is the price of enforcing both individual and group fairness? To better clarify the effect of extending solution enumeration to t-relaxed solutions on graphs of different sizes, we present Figure 8. The blue bars represent the patients with $PRA < 80\%$ and the orange bars represent the patients with $PRA \geq 80\%$. The numbers on the blue and orange bars represent the percentage of covered patients and hard-to-match patients in the pool, respectively. Note that for graphs with the pool size less than 60, we were able to cover all patients in the pool by $OPT - 2$ and only for graphs of size 60 and 70, we had to include $OPT - 3$ to cover all patients.

Additional information about Q5. To what extent does the global constraint reduce the number of search tree nodes and failures? In section 4.3 we introduced a new constraint designed specifically for KEP problems. The goal of designing a specific propagator was to use our knowledge of the problem structure to prune the search space more efficiently. Tables 6 and 7 compare the number of nodes and failures in the search tree in the standard CP model and the specialized model (i.e. the one equipped with the global constraint). The values in these tables are averages over the instances solved by both models. The results demonstrate substantial reductions in terms of number of nodes and failures across different data instances.

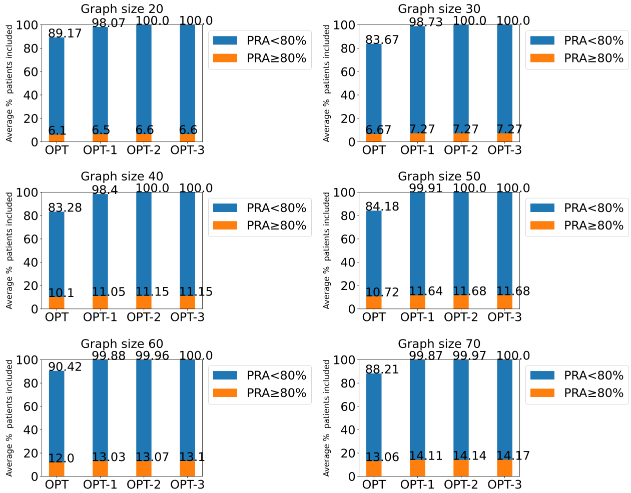

Figure 8: Effect of relaxation on the (relative) number of hard-to-match patients in solutions.

	standard		**specialized**	
size	**#failures**	**#nodes**	**#failures**	**#nodes**
20	104.38	206.84	64.6	127.28
30	11241.36	22480.76	5342.3	10682.64
40	1.46e+06	2.92e+06	3.57e+05	7.13e+05
50	4.98e+06	9.96e+06	4.12e+05	8.24e+05
60	2.74e+06	5.47e+06	5.48e+05	1.10e+06
70	2.77e+06	5.55e+06	1.83e+06	3.66e+06

Table 6: The average number of search tree nodes and failures for different graph sizes in *Canada* dataset.

	standard		**specialized**	
size	**#failures**	**#nodes**	**#failures**	**#nodes**
20	605.64	1209.36	334.76	667.6
30	3.39e+05	6.78e+05	12865.04	25728.08
40	2.11e+06	4.22e+06	4.29e+05	8.57e+05
50	4.83e+06	9.66e+06	8.67e+05	1.73e+06
60	8.23e+06	1.65e+07	1.41e+06	2.81e+06
70	8.52e+06	1.70e+07	1.29e+05	2.59e+05

Table 7: The average number of search tree nodes and failures for different graph sizes in *US* dataset.