Fairness in Kidney Exchange Programs through Optimal Solutions Enumeration

Golnoosh Farnadi[1,2*], **Behrouz Babaki**[3], **Margarida Carvalho**[2]
[1]Mila, [2]Université de Montréal, [3]Polytechnique Montréal
farnadig@mila.quebec, behrouz.babaki@polymtl.ca, carvalho@iro.umontreal.ca

Abstract

Not all patients who need kidney transplant can find a donor with compatible characteristics. Kidney exchange programs (KEPs) seek to match such incompatible patient-donor pairs together, usually with the objective of maximizing the total number of transplants. We propose a randomized policy for selecting an optimal solution in which patients' equity of 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. We empirically demonstrate the advantages of our proposed method over the common practice of using the first optimal solution obtained by a solver.

1 Introduction

Kidney exchange programs. Chronic kidney disease is a condition that leads to a slow loss of kidney function with no cure. In Canada, more than 4 million people suffer from it [20]. The size of the world population suffering from chronic kidney failure is increasing at an annual rate of around 6% [14]. These patients require a renal replacement therapy: kidney transplantation or dialysis. Dialysis alone corresponds to significant expenditures in the countries health systems. For instance, it is approximately 1.1% of total health expenditures in Canada [4]. Transplantation 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 longevity. Typically, patients are registered in a waiting list for a deceased donor transplantation or they can receive a direct transplantation from a compatible donor who is a friend or a relative. To overcome the long waiting times in these lists and to take advantage of incompatible patient-donor pairs, *Kidney Exchange Programs* (KEPs) have been implemented in several countries such as United Kingdom [19], The Netherlands [7], and Canada [18]). By gathering incompatible patient-donor pairs as well as altruistic donors, KEPs seek to maximize the patients benefit through the exchange of donors. The literature on the development of algorithms to compute exchange plans that maximize the patients benefit has focused in integer programming approaches [1; 17; 9]. In concrete, the typical objective is to maximize the number of transplants. Some KEPs, like the Dutch Program, use additional hierarchical criteria [16].

The problem. In [5], using simulated instances of the Canadian program, it is showed that very frequently there is a large number of optimal solutions for a current pool of incompatible pairs. 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. For instance, in the Canadian program, an accumulation of *hard-to-match* patients has been observed. Solution selection has a significant impact on the lives of the patients and evolution of the KEP pool overtime. Consequently, special care must be taken to ensure fairness in this process.

Contributions. Recently, in the context of machine learning, concepts of individual and group fairness have been broadly discussed [12]. While individual fairness aims to treat similar individuals similarly, group fairness aims to establish similar treatment among groups (subsets) of individuals. To the best of our knowledge, the literature has exclusively focused on group fairness on KEPs, generally, associated with the set of *highly-sensitized* (hard-to-match) patients, and subject to a potential detioration on the maximum number of transplants. In this paper, we introduce for the first time individual fairness as the supporting policy on solution selection. In concrete, we develop randomized processes among the sets of optimal solutions with the goal of favouring equity on the patients' likelihood of being in a selected solution. In particular, our processes avoid the ethical subjectivity of defining patients priority directly through their characteristics. Furthermore, we propose different methodologies to tackle the difficult problem of computing all optimal solutions.

2 Kidney exchange program

Mathematically, a KEP instance can be represented as a graph $G = (V, A)$, where V is the union of the set of incompatible pairs P together with 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 guarantees that the patient associated with a donor donating a kidney also receives a transplant. A chain is a path $(v_0, v_1, \ldots, v_n)$ in G starting in an altruistic donor $v_0 \in N$ and with $\{v_1, \ldots, v_n\} \subset P$. 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, $(1, 3, 2)$ 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. Furthermore, in Europe, their lengths are limited to three

*Contact Author

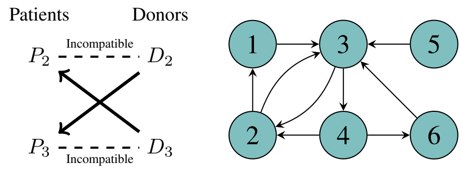

Figure 1: Left: Kidney exchange. Right: Compatibility graph for a KEP.

or four while, *e.g.* in the US, there is no limit on chains length [3]. In [10] it is argued that there is no advantage on considering chains larger than three. Therefore, based on the practical interest of cycles and chains of length at most 3, and for sake of simplicity, in this paper, we concentrate in this case. Nevertheless, our framework can be generalized to any length limit, although large lengths must result in slower running times.

The goal of KEP is to determine a set of disjoint chains and cycles representing the kidney exchanges to be performed such that the number of the patients receiving a transplant is maximized.

Integer programming formulations have been broadly used to model KEP: cycle formulation [1], edge formulation [21], position-indexed formulation [9]. 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 that each donor and patient participates 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 w_c is the number of patients involved on exchange c.

3 Related literature

In what follows, we review the literature on fairness associated with KEPs. As mentioned before, the main goal of KEPs is to maximize the benefit of the patients. In this way, weights are associated with each feasible exchange in order to reflect some utilitarian and priority criteria. The baseline is to associate weights that allow to maximize the number of transplants while prioritizing certain patients. For instance, highly-sensitized patients have a low probability of being compatible with a random kidney. In this context, Dickerson *et al.* [11] 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. Freedman *et al.* [13] focus on the fact that such prioritization can depend on human values. In this work, we do not use weights in the objective function. Instead of empirically bringing fairness to the solution, we effectively enforce it by characterizing the set of all optimal solutions (this is, exchange plans maximizing the number of transplants). Gao [15] provides a theoretical study on fairness also considering that KEPs run over time. Two prioritizing rules are analyzed, prioritization of *critical* patients (patients in a critical situation with high mortality rates) or highly sensitized patients in each run of KEP, and compared in terms of their fairness over time (defined as the number of losses, *i.e.*, patients who perished).

All the works mentioned above concentrate on group fairness, namely hard-to-match and critical patients. In this paper, we propose a new direction that does not consider the characteristics of groups of patients, but their individual chances. Furthermore, our approach has no extra cost of fairness (as discussed by [11]) since we always select an optimal solution (*i.e.*, it guarantees the maximum number of transplants).

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).

4 Fairness

When a KEP instance has multiple solutions (such that they differ in terms of patients who receive transplants), 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.

Definition 1. *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.

Definition 2. *A subset of optimal solutions $\mathcal{S}_k \subseteq \mathcal{S}$ is called* covering *if every vertex which appears in some solution in $\mathcal{S}$ also appears in some solution in $\mathcal{S}_k$.*

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.

4.1 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. We will now show that for certain criteria, the optimal strategy can be obtained by solving a linear program.

Minimizing the Mean Absolute Deviation. The goal is to determine the selection strategy δ such that it minimizes the absolute mean deviation (MAD) of each vertex in a selected solution. In other words, we aim to find the distribution δ that minimizes $\sum_{v\in P}|y_v - \frac{1}{|P|}\sum_{v\in P} y_v|$ where $y_v = \sum_{S:v\in S}\delta(S)$ is the probability of the vertex v in a selected solution.

The minimization of the mean absolute deviation problem can be formulated as a linear program. 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 linear program:

$$\min \sum_{v\in P} d_v \quad (2a)$$

$$\text{s.t.} \sum_{S\in\mathcal{S}} \delta_S = 1 \quad (2b)$$

$$y_v = \sum_{S:v\in S} \delta_S \quad \forall v \in P \quad (2c)$$

$$\sum_{v\in P} y_v = |P|\cdot z \quad (2d)$$

$$d_v \geq y_v - z \quad \forall v \in P \quad (2e)$$

$$d_v \geq z - y_v \quad \forall v \in P \quad (2f)$$

$$0 \leq \delta_S \leq 1 \quad \forall S \in \mathcal{S} \quad (2g)$$

$$0 \leq y_v \leq 1 \quad \forall v \in P. \quad (2h)$$

Constraints (2b) and (2g) ensure that δ is a probability distribution over $\mathcal{S}$. Constraints (2c) and (2h) establish y_v as the probability of vertex v. Constraint (2d) makes z equal to the mean over y. Finally, Constraints (2e) and (2f) lead to $d_v \geq |y_v - z|$, which together with the minimization of the objective function implies $d_v = |y_v - z|$.

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:

$$\max \quad z \quad (3a)$$

$$\text{s.t.} \quad z \leq \sum_{S:v\in S} \delta_S \quad \forall v \in P \quad (3b)$$

$$\sum_{S\in\mathcal{S}} \delta_S = 1 \quad (3c)$$

$$0 \leq \delta_S \leq 1 \quad \forall S \in \mathcal{S}. \quad (3d)$$

Again, constraints (3c) and (3d) ensure that δ is a probability distribution over $\mathcal{S}$. Constraints (3b) enforce z to be upper bounded by the smallest probability for a vertex v, while the objective function (3a) maximizes z (and thus, the lowest vertex probability).

5 Methodology

The selection strategies described in the previous section require us to compute $\mathcal{S}$, *i.e.*, the set of all optimal solutions. However, even computing the size of $\mathcal{S}$ when the KEP is restricted to exchanges of length at most 2 is known to be a #P-complete problem [22]. In what follows, we describe three techniques to perform this task.

5.1 No-good cuts

Let *OPT* be the optimal solution of $\mathcal{P}(\mathcal{C})$, then all its optimal solutions (*i.e.*, $\mathcal{S}$) can be determined in an iterative way by solving the original KEP formulation in Section 2 in iteration $K+1$ while adding the following constraints:

$$\sum_{c\in\mathcal{C}:x_c^k=0} x_c + \sum_{c\in\mathcal{C}:x_c^k=1}(1-x_c) \geq 1 \quad k=1,\ldots,K \quad (4a)$$

where x^k is an optimal solution of $\mathcal{P}(\mathcal{C})$ determined in a previous iterations. Constraints (4a) are the so-called *no-good cuts* [2] and, in order to be satisfied, the solution must differ in at least one entry for each x^k. Clearly, once Problem (4) becomes infeasible, then all optimal solutions have been determined. Furthermore, 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*.

5.2 Constraint programming formulation

We will now present a *Constraint Programming* (CP) model for the problem of enumerating the optimal solutions of KEP. 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 constraints of the CP model are as follows:

$$\textit{AllDifferent}(X) \quad (5a)$$

$$\big(X[v]=v\big)\vee\big(X[X[v]]=v\big)\vee\big(X[X[X[v]]]=v\big) \;\; \forall v\in V \quad (5b)$$

$$\sum_{v\in V}\big(X[v]\neq v\big) = \textit{OPT}. \quad (5c)$$

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.

Recall that the length of every cycle in the solution should be at most three. The constraint set (5b) enforces that each vertex is either in a self-loop, or a cycle of length two or three. Finally, given the optimal solution *OPT*, constraint (5c) ensures that the solution of the model is an optimal matching.

We also designed a specialized constraint for KEP problems, which replaces constraints 5a-5b in the CP model. Our propagator builds on the ideas and principles employed by the Compact-Table (CT) algorithm for filtering the *Table* constraint [8].

5.3 Finding a covering set of solutions

In Section 4 we argued that instead of enumerating all solutions, one can use any *covering* subset of solutions. We can use constraint programming for generating 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 specialized constraint to this setting.

6 Experimental evaluation

We investigate three research questions in our experiments: **Q1** How scalable are the proposed methods for enumerating the optimal solutions?, **Q2** What is the effect of graph size on the number of optimal solutions?, and **Q3** To what extent does our proposed method enhance the equity of chances for receiving a transplant?

Datasets For all the empirical evaluations that are presented in this section, we use two datasets which we call them the *Canada* and the *US* dataset. For the *Canada* dataset, we use a generator that is described in [5] to create instances for our experiments. For the *US* dataset, we use the same samples as [6][1]. In both of our dataset, the number of graph vertices ranges over 9 values: 20, 30, ..., 100. There are 50 different instances per graph size, amounting to a total of 900 different graphs in total.

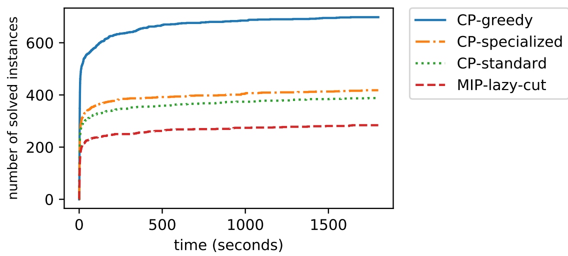

Figure 2: Performance profiles of different approaches for enumerating optimal solutions

In Section 5, we proposed four approaches to enumerate optimal solutions. To address Q1, we present the performance profiles of those methods in Figure 2. These include the no-good-cut approach of Section 5.1 (*MIP-lazy-cut*), the CP model of Section 5.2 (*CP-standard*), the CP model equipped with a specialized propagator (*CP-specialized*), and the CP approach for greedy covering (*CP-greedy*) of Section 5.3.

Among the three methods which enumerate all optimal solutions, *CP-specialized* performs the best in enumerating 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. Figure 2 shows that the *CP-greedy* method successfully finds a solution set for the majority of instances, often within a short amount of time.

To answer Q2, we inspect the optimal solutions obtained by *CP-specialized*. The average number of solutions over graphs of different sizes are reported in the top half of Table 2. Increase in 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. Observe that two different solutions can be equivalent in terms of patients that they cover (i.e. the same set of patients receive transplants in different cycles). For this reason, in a pre-processing step, we extract the unique set of solutions. The bottom part of Table 2 shows the average counts of these unique solutions. It is notable that even after removing the equivalent solutions, we could still have hundreds of thousands of unique optimal solutions.

	20	30	40	50	60	70
Canada	32	2,358	267,830	815,822	206,492	640,351
US	247	7,213	231,854	750,636	559,414	807,649
Canada	20	505	16,031	19,051	26,810	114,389
US	155	5024	142,517	264,419	164,292	231,347

Table 2: The average number of optimal solutions per graph size, before extracting the unique solutions (top) and after (bottom).

The large number of solutions in Table 2 indicates the importance of employing a selection procedure to ensure fairness. To address Q3 and measure the impact of our fairness-aware selection strategies, we analyze the quality of these strategies with the first-best and uniform solution selectors using both *CP-specialized* and *CP-greedy* approaches. We compare the probability of receiving a transplant for all vertices based on the *Maxmin* and *MAD* measures (for details see Section 4). The results are summarized in four plots that are presented in Figure 3.

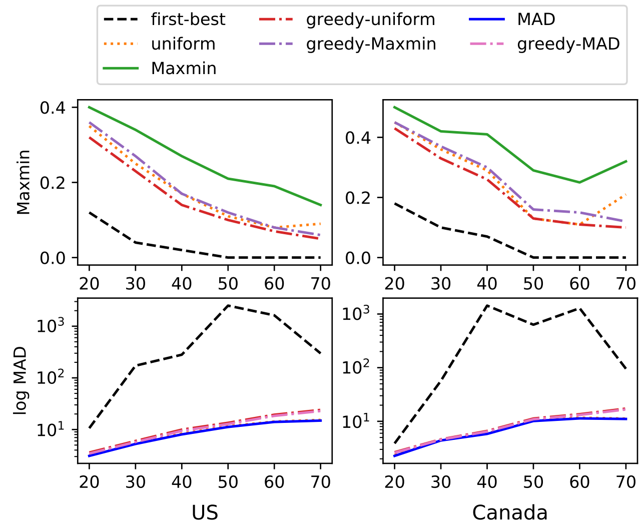

Figure 3: Comparing different selection strategies in terms of the *Maxmin* and *MAD* measures. The horizontal axis shows the graph size.

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 50 instances per graph size. It is observed that the enumeration-based strategies significantly improve the chances of the least well-off vertex, especially in larger graphs (where this chance approaches zero). The two plots on the bottom of Figure 3 compare the strategies in terms of the mean absolute deivation of all vertex probabilities (lower is better). Again, the enumeration-based methods dramatically outperform the baseline .

Two observations are in order. First, the uniform strategy is consistently competitive with the optimal strategy, while being much easier to calculate. Second, the greedy method provides high-quality results despite enumerating only a small subset of the optimal solutions. While our *CP-specialized* can scale up to 70 vertices, it is worth recalling that the *CP-greedy* scales to larger graphs which makes our approach more practical and scalable for various sizes of KEP.

7 Conclusion & future directions

In this work, the problem of individual fairness on KEPs is investigated for the first time. In order to tackle this issue, we propose two fair random processes for solution selection. Our computational experiments demonstrate the importance of our proposed approach in ensuring fairness among the patients. We believe this work has the potential to spark more research on fairness for solution selection in healthcare. An open question worth future investigation is the impact of our proposed selection strategies on the KEP pool over time.

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

Acknowledgments

The authors wish to thank the support of the Institut de valorisation des données and Fonds de Recherche du Québec through the FRQ–IVADO Research Chair in Data Science for Combinatorial Game Theory, and the Natural Sciences and Engineering Research Council of Canada through the discovery grant 2019-04557. Golnoosh Farnadi and Behrouz Babaki are supported by postdoctoral scholarships 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).

References

[1] David J. Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *EC*, pages 295–304. ACM, 2007.

[2] Egon Balas and Robert Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69, 1972.

[3] P. Biró and et al. Modelling and optimisation in european kidney exchange programmes. *Accepted for publication in the European Journal of Operational Research*, 2019.

[4] Canadian Institute for Health Information. High risk and high cost: Focus on opportunities to reduce hospitalizations of dialysis patients in Canada, 2016.

[5] Margarida Carvalho and Andrea Lodi. Game theoretical analysis of kidney exchange programs. *arXiv preprint arXiv:1911.09207*, 2019.

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

[7] Marry De Klerk, Karin M Keizer, Frans HJ Claas, Marian Witvliet, Bernadette JJM Haase-Kromwijk, and Willem Weimar. The dutch national living donor kidney exchange program. *American Journal of Transplantation*, 5(9):2302–2305, 2005.

[8] Jordan Demeulenaere, Renaud Hartert, Christophe Lecoutre, Guillaume Perez, Laurent Perron, Jean-Charles Régin, and Pierre Schaus. Compact-table: Efficiently filtering table constraints with reversible sparse bit-sets. In *CP*, 2016.

[9] John P. Dickerson, David F. Manlove, Benjamin Plaut, Tuomas Sandholm, and James Trimble. Position-indexed formulations for kidney exchange. *CoRR*, abs/1606.01623, 2016.

[10] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Optimizing kidney exchange with transplant chains: theory and reality. In *AAMAS*, pages 711–718. IFAAMAS, 2012.

[11] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. In *AAMAS*, pages 1013–1020, 2014.

[12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. *CoRR*, abs/1104.3913, 2011.

[13] Rachel Freedman, Jana Schaich Borg, Walter Sinnott-Armstrong, John P. Dickerson, and Vincent Conitzer. Adapting a kidney exchange algorithm to align with human values. In *AIES*, page 115, 2018.

[14] Fresenius Medical Care. Annual report 2018 - care and live, 2018.

[15] Irena Gao. Fair matching in dynamic kidney exchange, 2019.

[16] Kristiaan Glorie. *Clearing Barter Exchange Markets: Kidney Exchange and Beyond*. PhD thesis, Erasmus University Rotterdam, 12 2014.

[17] Xenia Klimentova, Filipe Pereira Alvelos, and Ana Viana. A new branch-and-price approach for the kidney exchange problem. In *ICCSA (2)*, volume 8580 of *Lecture Notes in Computer Science*, pages 237–252. Springer, 2014.

[18] Shafi Malik and Edward Cole. Foundations and principles of the canadian living donor paired exchange program. *Canadian journal of kidney health and disease*, 1(6), 2014.

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

[20] Braden Manns, Susan Q McKenzie, Flora Au, Pamela M Gignac, Lawrence Ian Geller, et al. The financial impact of advanced kidney disease on canada pension plan and private disability insurance costs. *Canadian Journal of Kidney Health and Disease*, 4, 2017.

[21] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney Exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 05 2004.

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