Université de Montréal

Towards Fairness in Kidney Exchange Programs

par

William St-Arnaud

Département d'informatique et recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Discipline

August 27, 2021

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Towards Fairness in Kidney Exchange Programs

présenté par

William St-Arnaud

a été évalué par un jury composé des personnes suivantes :

Fabian Bastin

(président-rapporteur)

Margarida Carvalho

(directrice de recherche)

Andrea Lodi

(membre du jury)

Résumé

Le traitement médical de choix pour la maladie rénale chronique est la transplantation d'organe. Cependant, plusieurs patients ne sont en mesure que de trouver un donneur direct avec lequel ils ne sont pas compatibles. Les Programmes de Don Croisé de Reins peuvent aider plusieurs paires donneur-patient incompatibles à échanger leur donneur entre elles. Typiquement, l'objectif principal d'un tel programme est de maximiser le nombre total de transplantations qui seront effectuées grâce à un plan d'échange. Plusieurs solutions optimales peuvent co-exister et comme la plupart correspondent à différents ensembles de patients obtenant un donneur compatible, il devient important de considérer quels individus seront sélectionnés. Fréquemment, ce problème n'est pas abordé et la première solution fournie par un solveur est choisie comme plan d'échange. Ceci peut mener à des parti-pris en faveur ou défaveur de certains patients, ce qui n'est pas considéré une approche juste. De plus, il est de la responsabilité des informaticiens de s'assurer du contrôle des résultats fournis par leurs algorithmes. Pour répondre à ce besoin, nous explorons l'emploi de multiples solutions optimales ainsi que la manière dont il est possible de sélectionner un plan d'échange parmi celles-ci. Nous proposons l'emploi de politiques aléatoires pour la sélection de solutions optimales suite à leur enumération. Cette tâche est accomplie grâce à la programmation en nombres entiers et à la programmation par contraintes. Nous introduisons aussi un nouveau concept intitulé équité individuelle. Ceci a pour but de trouver une politique juste pouvant être utilisée en collaboration avec les solutions énumerées. La mise à disposition de plusieurs métriques fait partie intégrante de la méthode. En faisant usage de la génération de colonnes en combinaison au métrique L_1, nous parvenons à appliquer la méthode à de plus larges graphes. Lors de l'évaluation de l'équité individuelle, nous analysons de façon systématique d'autres schémas d'équité tels que le principle d'Aristote, la justice Rawlsienne, le principe d'équité de Nash et les valeurs de Shapley. Nous étudions leur description mathématiques ainsi que leurs avantages et désavantages. Finalement, nous soulignons le besoin de considérer de multiples solutions, incluant des solutions non optimales en ce qui concerne le nombre de transplantations d'un plan d'échange. Pour la sélection d'une politique équitable ayant comme domaine un tel ensemble de solutions, nous notons l'importance de trouver un équilibre entre les mesures d'utilité et d'équité d'une solution. Nous utilisons le Programme de Bien-être Social de Nash afin de satisfaire à un tel objectif. Nous proposons aussi une méthodologie de décomposition qui permet d'étendre le système sous-jacent et de faciliter l'énumeration de solutions.

Mots clés: Programmes de Don Croisé de Reins; Équité; Programmation en nombres entiers

Abstract

The preferred treatment for chronic kidney disease is transplantation. However, many patients can only find direct donors that are not fully compatible with them. Kidney Exchange Programs (KEPs) can help these patients by swapping the donors of multiple patient-donor pairs in order to accommodate them. Usually, the objective is to maximize the total number of transplants that can be realized as part of an exchange plan. Many optimal solutions can co-exist and since a large part of them features different subsets of patients that obtain a compatible donor, the question of who is selected becomes relevant. Often, this problem is not even addressed and the first solution returned by a solver is chosen as the exchange plan to be performed. This can lead to bias against some patients and thus is not considered a fair approach. Moreover, it is of the responsibility of computer scientists to have control of the output of the algorithms they design. To resolve this issue, we explore the use of multiple optimal solutions and how to pick an exchange plan among them. We propose the use of randomized policies for selecting an optimal solution, first by enumerating them. This task is achieved through both integer programming and constraint programming methods. We also introduce a new concept called individual fairness in a bid to find a fair policy over the enumerated solutions by making use of multiple metrics. We scale the method to larger instances by adding column generation as part of the enumeration with the L_1 metric. When evaluating individual fairness, we systematically review other fairness schemes such as Aristotle's principle, Rawlsian justice, Nash's principle of fairness, and Shapley values. We analyze their mathematical descriptions and their pros and cons. Finally, we motivate the need to consider solutions that are not optimal in the number of transplants. For the selection of a good policy over this larger set of solutions, we motivate the need to balance utility and our individual fairness measure. We use the Nash Social Welfare Program in order to achieve this, and we also propose a decomposition methodology to extend the machinery for an efficient enumeration of solutions.

Keywords: Kidney Exchange Programs; Fairness; Integer programming

Contents

List of tables

List of figures

Notation and abbreviations

KEP	Kidney Exchange Program
IP	Integer programming
IF	Individual fairness
GF	Group fairness
CP	Constraint programming
SOCP	Second-ordre conic programming
MOOP	Multi-objective optimization
SWP	Social Welfare Program
NSWP	Nash Social Welfare Program
$\mathcal{P}(X)$	Set of probability distributions over the set X

2^X	Power set of set X

Introduction

Kidney Exchange Program

Every year, chronic kidney disease plagues millions of people worldwide. Dialysis is an effective treatment but can prove costly [1]. Kidney transplantation provides a more affordable alternative as well as enhancing the life quality of the patients and their life expectancy: this is why transplantation is often preferred. However, many patients do not have a compatible donor. In most countries, a compatible donor eligible for direct transplantation can either be a friend or a relative. The organ can also come from a compatible deceased donor; patients are registered in a waiting list managing their transplant priority whenever a deceased donor's kidney becomes available. However, patients in these lists can often wait years before a donor is presented with a compatible kidney, which would simply be inviable as patient health deteriorates over time. Kidney Exchange Programs (KEPs) allow patients that have an incompatible donor[1] to "trade" or exchange (in a figurative sense) their donor's kidney with another person. This facilitates the search of an organ for patients suffering from chronic kidney disease.

Explicitly, incompatible patient-donor pairs register in a KEP; these are implemented in several countries, e.g. South Korea [2], United Kingdom [3], Canada [4] and The Netherlands [5]. The system will try to swap each patient's donor with another from the available pool in order to form an exchange plan satisfactory to everyone involved, i.e. patients are matched with a compatible donor. If this cannot be done for a certain incompatible patient-donor pair, the pair is left untouched and considered again in the next KEP run. In other words, a patient only relinquishes their donor if they manage to obtain a new donor in return. As a simple example, we can consider two incompatible patient-donor pairs. If the patient of pair A is compatible with the donor of pair B, and vice versa, we can perform an exchange with these two pairs. Every donor in this exchange plan gives a kidney and every patient in the exchange plan receives a kidney. We can also further extend the basic description of a KEP by allowing chains starting with non-directed donors (NDDs), also known as altruistic

[1]This donor must be a friend or relative of the patient.

donors. These are donors that are not part of an incompatible patient-donor pair. Non-directed donors can give to whoever is compatible, without necessitating the receipt of a kidney from another pair or NDD; hence the term non-directed donor.

As we will see in the next chapter, a KEP instance motivates a mathematical optimization formulation, namely an integer program (IP). We can then lift tools from this particular field and find an optimal exchange plan. By optimal, we mean maximizing the patients' benefits. We must point out that we are referring to an exchange **plan**, therefore nothing is set in stone and no actual medical operation has been performed at this point. Also, for ethical reasons, the exchange of all kidneys and operation on each patient must be performed simultaneously when dealing with cycles. The reason behind this remark is that each patient would only be willing to let go of their incompatible donor if they can ensure receiving a compatible donor in exchange. Also, issues might arise if a patient or donor drops out before the actual medical procedure, thus invalidating part or the entirety of the exchange plan. An exchange plan must simply be seen as a tool that helps doctors in finding compatible donors for their patients, to the mutual benefit of other patients. When looking at Figure 0.1, it is possible

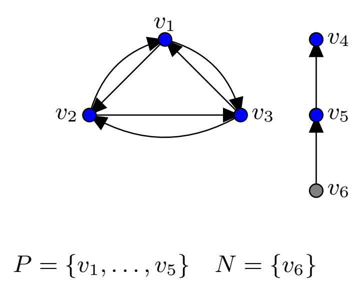


Fig. 0.1. Example of a KEP graph

to observe the key elements that form a KEP exchange plan. Vertices in the tuple (v_1,v_2,v_3) form a cycle, while the tuple (v_6,v_5,v_4) forms a chain. The sizes of the cycle and the chain are 3 and 2 respectively. In addition to the arcs forming the cycle (v_1,v_2,v_3), it can be seen that there are other arcs between the same set of vertices. For example, (v_3,v_2) is such an arc. It is referred to as a back-arc. In other words, a back-arc is any arc between two vertices of a cycle, that is not itself counted among the arcs of the cycle.

Contributions

Chapter 2 is based on the work of Farnadi et al. [6]. I contributed scientifically to this article, specifically in the methodology development, computational implementation and in the writing process. Chapter 4 is based on an article by Bertsimas et al. [7] and on the work of Lloyd Shapley [8]. My contribution consists of transposing the discussion related to fairness schemes to KEPs. I analyze the benefits and tradeoffs of the various schemes and from this discussion, I motivate the need to use the Nash Social Welfare Program [9]. Chapter 5 is based on the work of Charkhgarda et al. [9]. All the experiments were developed by my own hand as well as the algorithmic enhancement to the basic Nash Social Welfare Program.

Structure of this thesis

This thesis is structured in various chapters. Chapter 1 defines two important mathematical formulations of a KEP: the cycle formulation and the position-indexed edge formulation [10]. It also surveys the datasets that are used over the course of the experiments of this thesis and the experimental setup used to run them. In Chapter 2, the need to retrieve multiple solutions is motivated, and two different approaches are introduced: enumeration through integer programming and through constraint programming. The two approaches are then evaluated in terms of their efficiency to enumerate solutions. Finally, we motivate the need to have a good solution selection policy, i.e. a policy that is considered "fair" by the participants in the KEP. Chapter 3 introduces the concept of individual fairness (IF) [6]. Various metrics that can be used for this approach are described and evaluated experimentally. The chapter also compares IF to group fairness and lays out a method to scale the IF approach to larger graphs using column generation. Chapter 4 describes a useful set of concepts when discussing fairness in KEPs. First, Price of Fairness (POF) helps the reader to understand the tradeoff between a solution maximizing the number of transplants and selecting a fair solution. Then, various fairness schemes are introduced to further the discussion surrounding fairness in KEPs and ultimately to motivate the need of IF in the first place. These fairness schemes are: Aristotle's equity principle [7], Nash's standard of comparison [11], Rawlsian justice (or the *Veil of Ignorance*) [12], and Shapley values [8]. We also explain the difference between a fair outcome and a fair procedure as it is relevant to the introduction of distributions in the solution selection process of IF. Finally, the limitations of the utilitarian approach are discussed and we consider an alternative that seeks to balance both the social utility and fairness, i.e. the *Nash Social Welfare Program* (NSWP) [9]. Chapter 5 details the NSWP and how it can be applied to KEPs. An algorithmic enhancement similar to the column generation approach of Chapter 3 is provided, although in the context of second-order conic programming (SOCP). Experiments detailing the efficiency of the method, both in terms of the number of transplants that can be realized and the running time, are listed. Chapter 6 discusses some weaknesses of the methods introduced in the thesis and future improvements that can be realized. It also highlights interesting research avenues that could expand the scope of the current literature on fairness in KEPs.

Chapter 1

Preliminaries

1.1. Mathematical formulation

A KEP instance is a graph $G = (V, A)$ where $V = P \cup N$ is the set of incompatible patient-donor pairs P together with the non-directed donors N and where A is the set of arcs between these vertices. There is an arc (i,j) from vertex i to vertex j if and only if the donor of vertex i is compatible with the patient from vertex j. It can readily be seen that no $(i,j) \in A$ exists for $j \in N$, i.e. no altruistic donor can have an arc pointing to them.

The simplest way of formulating the instance mathematically with IP is to use the *cycle formulation*. We first need to compute the set $\mathcal{C}$ of cycles or chains of G. We introduce a binary variable $x_c \in \{0,1\}$ for each cycle (or chain) $c \in \mathcal{C}$. We denote the weight (benefit) of a cycle c by w_c. As a remark, it can be observed that generally, $w_c = |c|$ for cycles and $w_c = |c| - 1$ for chains, i.e. the number of involved patients. Different coefficient weights might correspond to utility values when receiving a kidney from a particular donor. We then have the following IP [13, 14]:

$$\begin{aligned} \max \sum_{c \in \mathcal{C}} w_c x_c & & \\ \text{s.t.} \quad \sum_{c \in \mathcal{C} | v \in c} x_c \leq 1 & \qquad \forall v \in V \\ x_c \in \{0,1\} & \quad \forall c \in \mathcal{C}. \end{aligned} \tag{CF}$$

The first constraint ensures that each vertex is in at most 1 cycle or chain. We need this constraint as performing an exchange c means that each donor's kidney in c will be matched with a patient, the one from the adjacent vertex in c. The constraint also ensures that a donor only donates one kidney. In other words, selected cycles or chains must be disjoint. We can modify the program (CF) by restricting the set V to P and eliminate NDDs. There are other ways of formulating a KEP in mathematical form. One drawback that we can see right away in the cycle and chain formulation is that its size can increase exponentially in terms of $|V|$.

Dickerson et al. [10] introduce the *position-indexed edge formulation* (PIEF) for KEPs. The main objective of this formulation is to tackle the exponentially large size that a KEP instance can take when described using (CF). The authors present a polynomial-size description by cleverly getting rid of symmetries in the choice of cycles via the introduction of graph "copies". First, they define $G^l = (V^l, A^l), \forall l \in V$ to be the subgraph of G induced by $\{i \in V : i \geq l\}$. They also define

$$\mathcal{K}(i,j,l) = \begin{cases} \{1\} & i = 1 \\ \{2, \ldots, K-1\} & i,j > l \\ \{2, \ldots, K\} & j = l, \end{cases}$$

the set of positions k at which edge $(i,j) \in A$ can be selected in the l^{th} copy. The value K is the limit to the length of chains or cycles and it is fixed beforehand. For the KEPs where cycles or chains are uncapped, the value of K can bet set to $|V|$. The set A^l can be thought as the set of arcs in the l^{th} copy of the graph. It consists of the set of arcs $\{(i,j) \in A : i,j \geq l\}$. Hence, the formulation looks like

$$\max \sum_{l \in V} \sum_{(i,j) \in A^l} \sum_{k \in \mathcal{K}(i,j,l)} w_{ij} x^l_{ijk} \tag{PIEF}$$

$$\text{s.t.} \sum_{l \in V} \sum_{j:(j,i) \in A^l} \sum_{k \in \mathcal{K}(j,i,l)} x^l_{jik} \leq 1 \qquad \forall i \in V \tag{1.1.1}$$

$$\sum_{\substack{j:(j,i) \in A^l \wedge \\ k \in \mathcal{K}(j,i,l)}} x^l_{jik} = \sum_{\substack{j:(i,j) \in A^l \wedge \\ k+1 \in \mathcal{K}(i,j,l)}} x^l_{ijk} \qquad \substack{\forall l \in V \\ i \in \{l+1,\ldots,n\} \\ k \in \{1,\ldots,K-1\}} \tag{1.1.2}$$

$$x^l_{ijk} \in \{0,1\} \qquad \substack{\forall l \in V \\ (i,j) \in A^l \\ k \in \mathcal{K}(i,j,l)}$$

The Constraint (1.1.1) ensures that all selected vertices can only appear once among all the graph copies. It also ensures that it has at most one adjacent vertex, i.e. its donor cannot donate more than one kidney or its patient cannot receive multiple kidneys. Finally, this constraint also ensures that each vertex is selected in at most one position in a cycle or chain. The Constraint (1.1.2) ensures that the patient of the incompatible patient-donor pair i receives a kidney if and only if its donor gives their own away. Otherwise, the pair does not participate in an exchange; it will be available for a future KEP run. The formulation is polynomial because $|\mathcal{K}(i,j,l)| \leq K \leq n$ for all arcs $(i,j) \in A$ and $l \in V$. We thus have $O(n^3)$ variables (remark, $n := |V|$). One interesting property of PIEF is that it has a very good LP relaxation when no chains are involved: it is equal to the LP relaxation for (CF) [10]. When chains are involved, the relaxation is not as tight and can even be arbitrarily bad. Dickerson et al. [10] show that this is not the case experimentally, especially on real and generated data. Furthermore, the number of constraints is also polynomially bounded since there are n inequality constraints and $O(n^3)$ equality constraints. PIEF can thus become a more tractable alternative when the size of the instance graph becomes large. Over the course of this thesis, both the cycle formulation and PIEF will be used. As a remark, it must be noted that other formulations exist [13, 14, 15, 16]. We chose the cycle formulation because it is simple and has good relaxation bounds. It is therefore the standard for small graphs with limited cycle and chain length. As for the PIEF, it is the state-of-the-art in KEPs and its use was warranted in this thesis.

1.2. Hierarchical optimization

Over the course of this thesis, whenever referring to *hierarachical optimization*, this will consist in optimizing a KEP formulation such as (CF) or (PIEF) in an iterative way with different objective functions, thus refining the set of solutions; see [17, 18]) for examples of objectives used in European KEPs. Explicitly, a KEP formulation is solved with a sequence of linear objectives $(w^1, \ldots, w^m)$. For the sake of simplicity, we suppose all of the objectives must be maximized. After, solving for w^i, a constraint of the form $(w^i)^T x \geq OPT_i$ is added to the model, where OPT_i is the optimal value obtained after optimizing the i^{th} objective (i.e. lexicographic optimization). Some natural candidates for the sequence of objectives are:

- Maximizing the number of transplants;
- Maximizing the number of cycles. This ensures that the selected cycles are small and if a pair drops out of an exchange, it will minimize its impact on the rest of the exchange plan;
- Maximizing the number of back-arcs. For example, if a pair drops out of a three-way exchange, and if there were two back-arcs between the other two pairs, then they will still be able to be matched together. Thus, by having more flexibility using back-arcs, an exchange plan is more robust to pairs dropping out of the KEP pool. One can look at Figure 0.1 for reference. If the edges $\{(v_1,v_2),(v_2,v_3),(v_3,v_1)\}$ are selected as part of the exchange plan, then the arc (v_1,v_3) is a back-arc.

1.3. Datasets

In this section, the reader can find a description of the datasets that are used in the experimental results of this thesis. These datasets mimic real-world graph topology of KEPs and have been widely used in the literature. One can now explore how these datasets are generated to validate their use when analyzing the concept of fairness in KEPs, a notion that will be introduced in Chapter 2.

1.3.1. INESC TEC / PrefLib (Saidman Generator)

The Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)[1] and PrefLib[2] data is generated using the Saidman generator [19]. Based on the ABO model (see Figure 1.1), every generated patient is coupled with one generated donor. There are conditionally independent probability distributions of the patient-donor pair properties:

(1) blood types for the patient and donor are generated according to a predefined distribution (ABO model);
(2) The sex of each patient and donor follows a predetermined probability distribution based on the general population;
(3) The relationship between the patient and donor is either "spouse" or "other" and is generated using a predetermined distribution;
(4) Patients were attributed a low, medium, or high Panel-Reactive Antibody (or PRA, see definition below) level and a compatible positive cross-match probability with a random donor (1 value per category). In the event of a couple with a female patient, the positive cross-match probability is updated to $100 - 0.75 \times (100 - PRA)$.

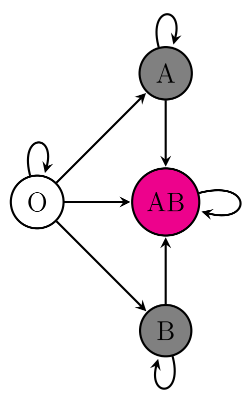


Fig. 1.1. ABO model: an arc indicates donation compatibility. blood type O is universal donor and blood type AB is universal recipient.

Definition 1.3.1. *The PRA percentage corresponds to the chance of a patient having antibodies that would reject a donor's organ when that donor is drawn at random from the general population.*

Thus, having a low PRA value increases a patient chance of finding a match when compared to an individual that has a very high PRA. The PRA is thus a value between 0 and 1 (0 and 100%). Many pairs are generated and those that have compatible blood types or a positive cross-match are discarded. Amongst the incompatible patient-donor pairs that are left, the compatibility arcs are determined according to the positive cross-match probability of each

[1] `https://rdm.inesctec.pt/dataset/ii-2019-001`
[2] `https://www.preflib.org/data/MD/00001`

patient. All the values for the default generator (i.e. the one used for the INESC TEC data) can be found on James Trimble's GitHub page[3] and in Saidman et al. [19].

Instances for the INESC TEC dataset were generated for various graph sizes, i.e. the size $|V|$. No NDDs were provided as part of the graphs. Thus, all the experiments involving this dataset do no make use of chains. The graph sizes in this dataset are: 20, 30, 40, 50, 60 and 70. In total, 50 instances per graph size are provided, for a total of 300 instances.

1.3.2. UNOS dataset

In Dickerson et al. [10], the authors use both real and synthetic data. The two real datasets that are used are the UNOS (United Network for Organ Sharing) US-wide exchange and the NLDKSS UK-wide exchange. The synthetic data is obtained from a generator seeded with the real UNOS data. The code for the generator can be found on Dickerson's GitHub page[4].

After inspecting the code, we can observe that there are multiple samplers available to us. The one that is used by Dickerson et al. [10] is the *RealSplitUNOSSAmpler*. It takes the set of all UNOS pairs and altruistic donors, and samples randomly (with replacement). Although all match runs of the UNOS programs are used, it is possible to only use a subset of them. We can thus generate graphs of the desired size that will conform to a distribution induced by the UNOS data (a real dataset).

In this thesis, all experiments involving the UNOS dataset were done over instances of sizes: 16, 32, 64, 128, 256, and 512 vertices. By having exponentially increasing sizes, it will be possible to validate the scaling of experiments to larger instance sizes than for the INESC TEC data, especially in Chapter 5. Each graph size has instances with no NDDs and instances with varying amounts of NDDs (5,10 and 15% of the graph size). There are thirty instances of size 16 and forty instances for the other graph sizes for a total of 230 instances.

1.4. Experimental setup

In all experiments realized as part of this thesis, the solvers used were Gurobi 9.0.1 (Chapters 2 and 3) and Mosek 9.2 (Chapter 5). The only exception is for Section 2.2.3, where the constraint programming solver used was OscaR-CP 4.1.0 (with Scala 2.12.8[5]) combined with CPLEX 12.8 (relaxation of linear programs). Nodes from the Compute Canada clusters[6] (Cedar) were used to run the experiments. For each job, one CPU and one thread were used. A time limit of one hour was set and 8 gigabytes of RAM was allocated

[3] https://jamestrimble.github.io/kidney-webapp/#/generator

[4] https://github.com/JohnDickerson/KidneyExchange/blob/master/src/edu/cmu/cs/dickerson/kpd/structure/generator/UNOSGenerator.java

[5] https://www.scala-lang.org/

[6] https://www.computecanada.ca

for each task. The detailed list of parameters used for each section can be found in Table 1.1. It also contains the cycle and chain cap K used for each experiment. All the experiments

Section	Dataset	RAM	Time	Language	Solver	Cycle cap	Chain cap
2.2.3	INESC TEC	8GB	1h	Scala 2.12.8	Oscar-CP 4.1.0	3	3
2.3	INESC TEC	8GB	1h	Python 3.7.4	Gurobi 9.0.1	3	3
3.3	INESC TEC	8GB	1h	Python 3.7.4	Gurobi 9.0.1	3	3
3.4	INESC TEC	8GB	1h	Python 3.7.4	Gurobi 9.0.1	3	3
3.5	INESC TEC	8GB	1h	Python 3.7.4	Gurobi 9.0.1	3	3
5.5.1	UNOS	8GB	1h	Julia 1.4.1	Mosek 9.2	3	0
5.5.2	UNOS	8GB	1h	Julia 1.4.1	Mosek 9.2	3	3

Table 1.1. Parameters used for the experiments[7]

in Chapters 2 and 3 were programmed using Python 3.7.4[8], except for the Section 3.5, which used Julia 1.4.1[9] combined with JuMP 0.21.8[10]. Experiments in Chapter 5 also made use of Julia and JuMP.

[7]The code for all the experiments can be found at `https://github.com/stawaway/William-s-Master-Thesis`.
[8]`https://www.python.org`
[9]`https://www.julialang.org`
[10]`https://jump.dev`

Chapter 2

Methodology

In this chapter, we explore the need to strengthen our grasp on the choice of solution for (CF). Using this newly gained control over the choice of solution, we aim to further our objective of developing responsible algorithms that do not exhibit some undesirable properties such as biases anchored to individuals' characteristics. In Section 2.1, we motivate the need for fairness in KEPs. Section 2.2 explores different mechanisms to enumerate solutions of a KEP. Finally, in Section 2.3, we propose a solution selection policy that takes advantage of having multiple solutions to promote fairness.

2.1. Multiple solutions

A key point to observe is that for a particular KEP instance, we can have multiple optimal solutions [20]. García-Soriano and Bonchi [21] discuss this multiplicity for general matching problems. These are all solutions that maximize the objective (cf. (CF)). At first, we will consider the objective of maximizing the total number of transplants performed, as it is usually the primary goal of KEPs to provide as many patients as possible with a new kidney [17]. Later, alternative criteria will be introduced in the form of a "hierarchical" structure of simpler objectives to maximize in a lexicographic way. A table of the various criteria that are used in the European KEPs can be found in [17].

We first note that using any of the mathematical formulations of the previous Chapter, we can have multiple exchange plans that correspond to the same set of patients receiving a kidney. By looking at Figure 0.1, it can be seen that an exchange plan involving the cycle (v_1,v_2,v_3) includes the same patients as the cycle (v_1,v_3,v_2). Under the assumption that these same patients do not have a preference for their choice of donor, it can be seen that all these exchanges are equivalent. This motivates the introduction of *KEP-equivalent* exchange plans.

Definition 2.1.1. *Two solutions (exchange plans) are called* ***KEP-equivalent*** *if the set of patients that receive a kidney is the same under each solution.*

Figure 0.1 shows that the exchange plan defined by the set $\{(v_1,v_2),(v_2,v_3),(v_3,v_1)\}$ of arcs contains the same patients as the exchange plan $\{(v_1,v_3),(v_3,v_2),(v_2,v_1)\}$. One can look at the number of (optimal) exchange plans that contains a particular patient. Some patients will be in a large number of those, while others can be in very few optimal exchange plans. If one always picks the same solution, say the one with the lowest lexicographic ordering (when looking at the solution vector), patients that are not in this exchange plan but who might be in another one will never be chosen. We then have a problem: is this process fair? It seems that it is not since some patients are prioritized over others in terms of their chance of receiving a compatible kidney. Surely, the viability of a KEP will lie in its capacity of maximizing the matching probability of every patient, if not, why participate in the first place?

The natural way of tackling this issue is to account for multiple solutions of a KEP. By having many candidate solutions to choose from, one increases the capability to add variance in the choice of exchange plan and correct for the bias introduced by the optimization method (the one finding a single solution). The reader can later see how having a set of solutions lends itself to using distributions over that set as a way to draw an exchange plan randomly. But now, to consider multiple solutions, one first needs a method to enumerate them. We propose two basic approaches to doing so based on integer programming (IP) and constraint programming (CP).

2.2. Enumerating optimal solutions

2.2.1. IP enumeration

Using the mathematical program (CF), we can enumerate further solutions by adding a solution cut constraint and an optimality constraint. The first constraint has the form:

$$\sum_{c\in\mathcal{C}|x_c^k=1}(1-x_c)+\sum_{c\in\mathcal{C}|x_c^k=0}x_c\geq 1 \tag{2.2.1}$$

where x^k is the k-th solution enumerated by the method. Essentially, the inequality forbids x^k from being computed again in the next iteration. The optimality constraint has the form:

$$\sum_{c\in\mathcal{C}}w_cx_c=OPT, \tag{2.2.2}$$

where OPT is the optimal value of the initial IP. One first solves the problem in its basic form to obtain the first solution x^1. Then a cut is added for that solution and the entire problem is resolved with the new cut. One obtains x^2 and the procedure is repeated until infeasibility is reached. This can be described by Algorithm 2.2.1. The main issue with this approach is that we are only cutting one solution at each iteration. Unfortunately, the enumeration of solutions for an IP problem is NP-hard [13].

Algorithm 2.2.1. Enumeration using solution cuts

```
procedure ENUMIP(G, OPT)                          ▷ G = (V,A), where V = P ∪ N
    sols ← [ ]                                    ▷ initialize to empty array
    while true do
        x ← SOLVEKEP¹ (G, OPT)                    ▷ returns null if no solution²
        addCut(x, SOLVEKEP)³                      ▷ add solution cut for x
        if x = null then
            break
        end if
        append(x, sols)                           ▷ append solution x to the list
    end while
    return sols
end procedure
```

2.2.2. CP enumeration

Another solution is to propose a mechanism based on constraint programming. Constraint programming only seeks to find a solution that satisfies a certain set of constraints. Many solvers also exist for this particular framework. Before delving into it, we will introduce a few basic concepts that are at the core of this approach. For the reader interested in an extensive exploration of constraint programming and its applications, the *Handbook of Constraint Programming* [22] can be a good introduction. Constraint programming takes a slightly different route to enumerating the solutions. We must first model the KEP instance as part of the CP formulation. The CP problem can be represented as a tuple (V, D, C), where V is the set of variables, D the domain of the variables, and C the constraints. Each constraint is defined over a subset of variables and determines the valid assignments. The CP problem is then solved through search and propagation. The search consists of "fixing" values to variables and propagation is done through the constraints that include those variables or a subset of them. A propagator for each constraint is used for that specific purpose. Using these ideas, we can represent the CP constraints mathematically. To begin with and for the sake of clarity, it is assumed that the limit to the size of cycles and chains is 3. In fact, this is the limit adopted by several KEPs due to practical reasons [14, 23]. We use X, an array of variables indexed by the vertices of V and the notation $X[v]$ to mean the successor of v in some path. Its domain is defined as $\{v\} \cup \{u : (v,u) \in A\}$. In other words, $X[v]$ represents the patient to whom v's donor gives their kidney. By $X[v] = v$, it is implied that the donor in v does not donate under the exchange plan. The different constraints in the model are:

$$\begin{aligned} & X[v] = X[w] \iff v = w \quad \forall v,w \in V \\ & (X[v] = v) \vee (X[X[v]] = v) \vee (X[X[X[v]]] = v) \quad \forall v \in V \\ & \sum_{v \in V} \mathbf{1}_{\{w : X[w] \neq w\}}(v) = OPT \end{aligned} \tag{CP^*}$$

The first constraint ensures that no vertex is chosen as the successor of two or more vertices (i.e. we get disjoint cycles and chains). Also notice that the second constraint is for cycles of maximal size 3, but we can extend the concept to any cycle size by adding more terms. Thus, for size K, we can add constraints $X[X[X[X[v]]]]$ up to $X \circ \underbrace{\ldots}_{K \text{ times}} \circ X[v]$ to the sequence of logical disjunctions. The last constraint of the model ensures that we are enumerating optimal solutions. We can always relax the optimality condition if we wish and this will be explored later on.

Thus, Algorithm 2.2.2 describes the CP enumeration. At every node of the search tree, the CP method selects the variable with the smallest number of values left in its domain. Two branches are then created. On the first branch, the solver fixes the value of the variable to the smallest possible, removes the other values and then triggers a call to the procedure *propagate*. On the second branch, the smallest value of variable's domain is instead removed and the solver calls *propagate*. The bitset $contains[v,u]$ is calculated for each arc (v,u): a component is equal to 1 if and only if the corresponding cycle contains the arc (v,u). Another binary mask is maintained in memory: *validCycles*. At every node of the search tree, it is computed. For each cycle, a value of 1 indicates that the cycle can still be selected as part of a solution without violating the constraints (CP^*). The procedure *updateCycle* uses the sets Δ_v of elements removed from the domains to determine which solutions are infeasible. It essentially eliminates the cycles that cannot be part of a solution after the Δ_v's have been updated by a call to *filterDomains*. The Δ_v's only keep the removed values from a previous call to *propagate* in the search tree (current node or upstream) The procedure *filterDomains* uses the domains of the variables and the graph arcs to determine which vertices can be ruled out of each variable's domain. Finally, the method *propagate* combines both previous procedures to find a solution. By employing the LP relaxation of the KEP formulation, it is able to take advantage of a key feature of integer programming; by computing a bound on the relaxation, the algorithm is able to have another indicator (apart from the directly verifying the valid cycles) as to whether or not an integer solution exists at the current node in the search tree. When solving the problem *relaxedKEP*, we set the variables x_c to the same values as the corresponding *validCycles* component.

Algorithm 2.2.2. Enumeration using CP

```
procedure updateCycle(G, OPT)                          ▷ G = (V,A), where V = P ∪ N
    for v ∈ V do
        mask ← 0                       ▷ binary mask with one component for each cycle c
        for u ∈ Δ_v \ {v} do     ▷ Δ_v is the set of removed values from the domain of X[v]
            mask ← mask | contains[v,u]                                  ▷ bitwise or^4
        end for
        mask ← ¬ mask                                                ▷ bitwise negation
        validCycles ← validCycles & mask                                  ▷ bitwise AND
    end for
end procedure
procedure filterDomains
    for v ∈ V do
        for u ∈ dom(X[v]) \ {v} do             ▷ dom(w) is the domain of the variable w
            if validCycles & contains[v,u] = 0 then
                dom(X[v]) ← dom(X[v]) \ {u}  ▷ remove u since it is never in valid cycles
            end if
        end for
    end for
end procedure
procedure propagate
    udpateCycles()
    if validCycles = 0 then
        return Backtrack      ▷ backtrack since there is no valid cycle under the current
assignment
    end if
    UB ← relaxedKEP()                                            ▷ LP relaxation of (CF)
    if UB < OPT then
        return Backtrack             ▷ the desired integer solution will never be found
    end if
    filterDomains()
end procedure
```

2.2.3. Comparison of the enumeration methods

In order to compare the IP and CP approaches for enumeration, we can look at how many instances they are able to solve; in other words, when can we fully enumerate all solutions with each method? We will use the following enumeration methods for our comparison:

- IP-lazy-cut
- CP-standard
- CP-specialized
- CP-greedy

IP-lazy-cut will refer to the IP enumeration of Algorithm 2.2.1. This is referred to as *lazy cut* because we generate the solution cuts using the lazy cut function of Gurobi. *CP-standard* corresponds to giving CP^* to the solver. *CP-specialized* will refer to Algorithm 2.2.2. We want to see if this propagator performs better than the general propagator. Finally, *CP-greedy* simply uses CP to find a greedy cover of the vertices.

Definition 2.2.3. *A set of solutions of a KEP is a* ***greedy cover*** *if each individual solution includes at least one vertex that is not present in the other solutions of that set.*

With this in mind, when trying to solve instances of various sizes (indexed by the number of vertices in each graph), we obtain Figure 2.1. The INESC TEC dataset was used [19]. By inspecting Figure 2.1, we conclude that CP methods seem to better take advantage

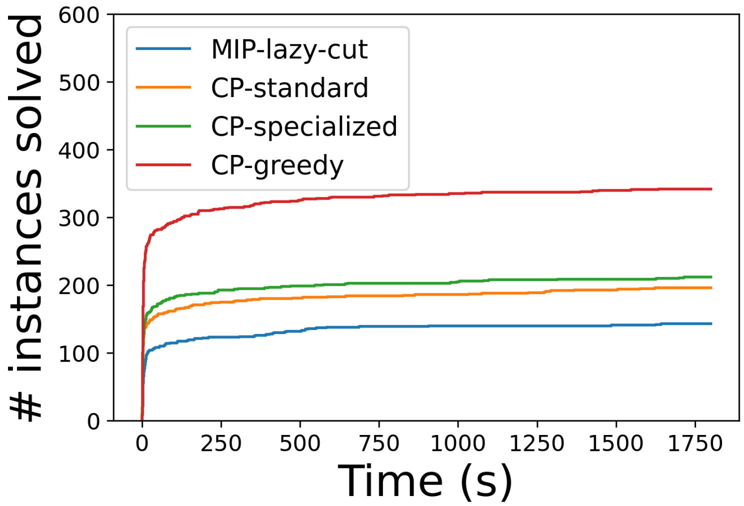


Fig. 2.1. Comparison of enumeration methods for optimal KEP exchange plans

of the available state-of-the-art solvers. The specialized propagator (*CP-specialized*) is an improvement over the default propagator. *CP-greedy* is highly efficient but since it uses a greedy cover for the enumeration, we anticipated it to be able to solve more instances because we are not looking for all possible solutions. We will later examine if having a greedy cover can achieve good results when tackling fairness. To better understand the complexity of this problem, it is interesting to see the actual number of solutions enumerated for the instances. These can be found in Table 2.1. The row of Table 2.1 labelled *Projected* corresponds to only enumerating solutions such that no two solutions are KEP-equivalent. The *Hierarchical* row corresponds to performing multiple optimizations over different objective functions in a sequential way. In this particular case, maximization of the number of transplants was performed, followed by maximization of the number of cycles, and then maximization of the number of back-arcs. These secondary criteria can be found in the sequential optimization procedure of some Kidney Exchange Programs [17].

$\lvert V \rvert$	20	30	40	50	60	70
All	256	8524	220181	352953	594217	788673
Projected	31	86	2522	6570	7001	10953
Hierarchical	76	495	54012	142137	62569	42478

Table 2.1. Average number of solutions per graph size

2.3. Solution selection policy

Once we have a set of optimal solutions, we still need to pick an exchange plan that will be realized. Now, a natural approach is to draw a solution from this set. This system of lottery is the most intuitive way to eliminate the bias and arbitrariness introduced by selecting the solution given by a solver. For example, the solutions returned by a solver might be biased against O blood-type patients since it is difficult to find a compatible donor. However, we still need to consider the types of distributions that make the selection "fair". If we draw solutions uniformly at random, we can still run in the issue of having multiple solutions corresponding to the same set of patients. Therefore, these KEP-equivalent solutions are prioritized in a sense over other sets of patients that do not exhibit such behaviour (i.e. they do not have multiple KEP-equivalent solutions including them). We can counter that effect by accounting for each patient's true probability of receiving a kidney when drawing exchange plans from the optimal solution set. When all these probabilities are at one's disposition, it is possible to set a policy in place that will increase the probability of the most disadvantaged individuals. No characteristic of the patients except these probabilities are taken into account. This concept will be referred to as *individual fairness* (IF) and described more thoroughly in the next Chapter. This will be contrasted with the concept of *group fairness* (or GF), where certain groups of individuals are given priority because of characteristics that normally negatively affect their chance of getting matched in an exchange plan (e.g. their PRA score or their blood type) [23, 24].

To better understand the balance between individual and group fairness, we can evaluate over instances how many of the patients are included by the distribution, that is patients with a nonzero probability of being selected as part of an exchange plan. We can measure how this value will vary as we relax the constraint enforcing that a maximal number of transplantations are to be performed. We do this measurement both for the absolute number of patients included and for the subgroup of patients that are labelled as hard-to-match using a *PRA* value threshold of 80% [23]. Because hard-to-match patients are more likely to be incompatible with many donors, they will tend not to be selected in exchange plans. They thus form a group of patients that are at a disadvantage when looking for a compatible kidney. A group fairness approach will then account for these patients' smaller probability of receiving a kidney by actively looking to improve this probability. Again, we use instances from the INESC TEC dataset and obtain Figure 2.2. Figure 2.2 shows that we do not

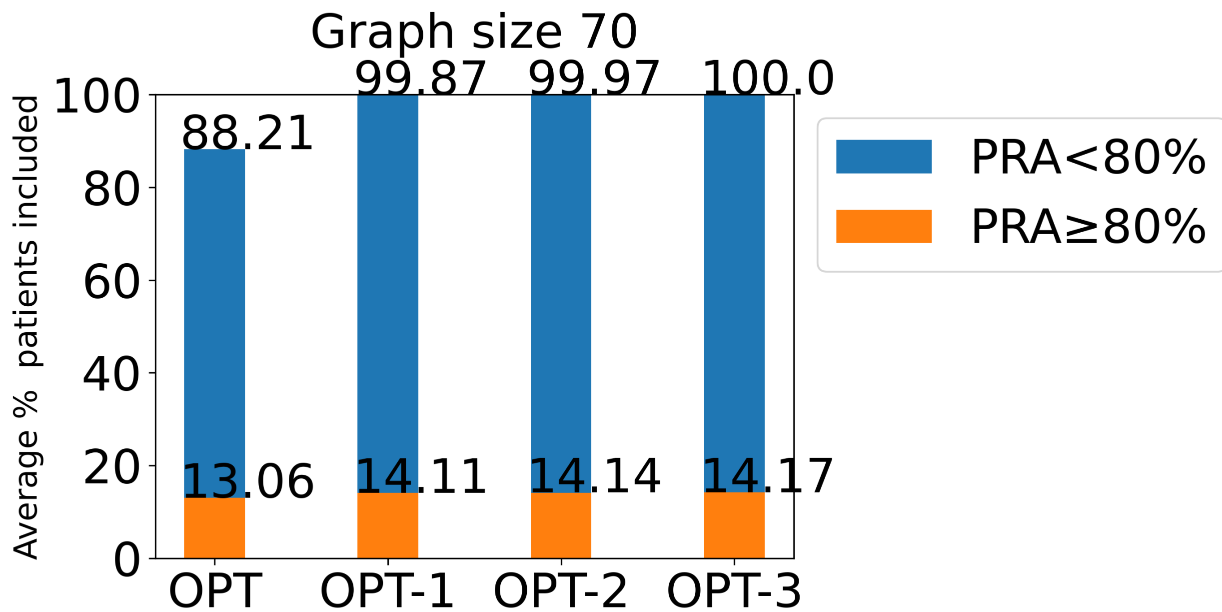


Fig. 2.2. Measuring the effect of relaxing the optimality constraint

have to sacrifice greatly the number of transplants to include more patients. In fact, we reach the maximum value possible quite easily. When relaxing the optimality constraint to $OPT - x$ (i.e. $\sum_{c \in \mathcal{C}} x_c \geq OPT - x$ during enumeration), it is possible to include the maximal number of patients with a value of $x = 3$. The same can be said of the hard-to-match patients. Therefore, we can account for the concept of group fairness even within the scheme of individual fairness and find a desirable balance between the two. Although the way this was done can seem ad hoc, we will later see that we can have a systematic process to balance these concepts.

Chapter 3

Fairness in KEPs

In the previous chapter, we showed the practical diversity of optimal solutions in a KEP. We will now demonstrate how to leverage this diversity and ensure fairness. In Section 3.1, distributions over the set of optimal solutions are considered: metrics over these sets of distributions will be optimized as part of a new concept called *individual fairness* (IF). Section 3.2 discusses various metrics that will be used to validate IF experimentally. In Section 3.3, the general mathematical form of the optimization process will be explicitly given and multiple experiments will be devised to evaluate the properties and performance of the individual fairness scheme. Section 3.4 will compare IF with GF in a bid to explore the pros and cons of each approach. The reader will then be introduced to the useful concept of column generation (Section 3.5), in an effort to scale the IF approach. Next, the reader will be able to see the connection of the various metrics to their philosophical roots in regards to fairness.

3.1. Individual fairness

Supposing we have a set of solutions (exchange plans) S to draw from, we define $\mathcal{P}(S)$ to be the space of probability measures (distributions) over the solutions of S. We define δ_s as being the probability of selecting solution $s \in S$ under distribution $\delta \in \mathcal{P}(S)$. We also define each patient v's probability of receiving a kidney as δ_v for all $v \in P' \subseteq P$, where $v \in P'$ if and only if v is in at least one solution of S (P is the set of patient-donor pairs). The reason for introducing P' is that it would be nonsensical to factor in pairs that cannot be in an optimal solution when computing a fairness score. Since they do not even affect the selection process, by giving them a weight in the decision process involving other pairs would be unfair in itself. We remark that the vertices v here refer to a patient-donor pair. Therefore, δ_v corresponds to the probability of receiving a kidney for the patient of pair v. We will refer to the *vertex set* of solution s as the set $\pi(s) = \{v \mid v \in c \wedge s_c = 1\}$. We have the following equation for δ_v:

$$\delta_v = \sum_{s \in S | v \in \pi(s)} \delta_s.$$

Given that we now have defined a probability associated to each vertex in the graph, we can proceed to define a measure of individual fairness for these vertices. This will come in the form of a metric $L : \mathcal{P}(S) \to \mathbb{R}$. We will seek to find the distribution $\delta \in \mathcal{P}(S)$ that minimizes the loss $L(\delta)$. Intuitively, this can often be thought of as minimizing the difference in the probabilities of each patient to obtain a kidney. In an ideal world, we can get as close as possible to equal probability. It is important to mention that the set S here refers to optimal solutions in terms of maximizing the number of transplants, but this set could be enlarged with other less efficient (again, with respect to maximizing the number of transplants) solutions. The tradeoff between the utilitarian and fairness approaches will be explored further down the road.

3.2. Various metrics

We previously introduced the concept of a metric over a distribution in $\mathcal{P}(S)$. Next, we will define some of these metrics that will prove useful in the rest of this thesis.

Definition 3.2.1. *The L_p loss is given by*

$$L_p(\delta) = \left(\sum_{v \in P'} (\delta_v - \bar{\delta})^p \right)^{\frac{1}{p}},$$

where $\bar{\delta} = \frac{1}{|P'|} \sum_{v \in P'} \delta_v$ is the average probability of receiving a kidney.

When using these metrics, we will seek to minimize their value, thus minimizing the discrepancy between the different pairs' probabilities of being selected as part of an exchange plan. We will make heavy use of the L_1 and L_2 losses in the sections that will follow. We can now introduce another natural metric.

Definition 3.2.2. *The minprob metric is given by*

$$\mathit{minprob}(\delta) = \min_{v \in P'} \delta_v.$$

This metric has been proposed in García-Soriano and Bonchi [21] in the context of fair matchings. Remark that when using this metric, we will seek to find the distribution that maximizes its value. In other words, *minprob* maximizes the probability of the patient with the least chance of receiving a kidney.

3.3. Evaluating the various metrics

It is interesting to see how the metrics introduced previously affect the distribution over exchange plans that will be selected. In order to see this effect, we optimize for the optimal distribution corresponding to each metric and then test it against the other metrics to see how it performs. Nevertheless, we will introduce the problem that we seek to optimize:

$$\text{optimize } L(\delta) \quad \text{subject to}$$
$$\delta \in \left\{ \mathbb{R}_+^{|\mathcal{S}|} \;\middle|\; \sum_s \delta_s = 1 \wedge S \text{ is the set of optimal solutions} \right\}$$

where L is our metric.

Proposition 3.3.1. *All the metrics described in Section 3.2 are either convex (L_1, L_2) or concave (minprob).*

Because all the resulting mathematical formulations are either linear of convex quadratic problems, we can use standard solvers like Gurobi or Mosek.

We look at the following distributions over solutions for instances created with the Saidman generator [19], namely the INESC TEC dataset:

- Optimize over L_1, L_2 and *minprob*;
- Greedy versions of the metrics L_1, L_2 and *minprob* (i.e. we find a greedy cover during the enumeration of solutions);
- Uniform distribution, where each $s \in S$ has an equal probability of being selected;
- A baseline, *First-best*, that attributes probability 1 to the first enumerated solution (i.e. working with only one exchange plan; e.g. solution to (CF) or (PIEF)).

The solution set was enumerated using the CP solver as it was the most efficient alternative (see Figure 2.1). From Figure 3.1, we can see that all of the distributions corresponding to the metrics, outperform the baseline. An interesting aspect to observe is that *Uniform* seems to perform well across all metrics. As the results are averaged over all the instances, however, this might not necessarily be the case for special worst-case scenarios. In Figure 3.2, the uniform distribution over all possible exchange plans implies that some vertices will have a chance of being selected close to 1, while for other vertices (those exclusively in the cycle), this probability will be close to 0. Meanwhile, using the L_1 norm will result in the cycle C_L being selected with probability $\delta_{C_L} = \frac{L}{\frac{5L}{2}-1}$, while the yellow and green paths are each selected with probability $\frac{1}{2} - \frac{\delta_{C_L}}{2}$ (see Appendix A). In effect, this means that all vertices will have a chance of being in an exchange plan closer to $\frac{1}{2}$. While the uniform distribution does well in general, one can observe that the greedy versions of the distributions are worse than their basic version. However, this was expected as we are trading optimality for computability (finding a cover is easy).

3.4. Comparison of IF and GF

After introducing IF, it becomes relevant to compare to GF. Since both methods take completely different approaches to tackle and define fairness, we perform three measurements over the INESC TEC instances:

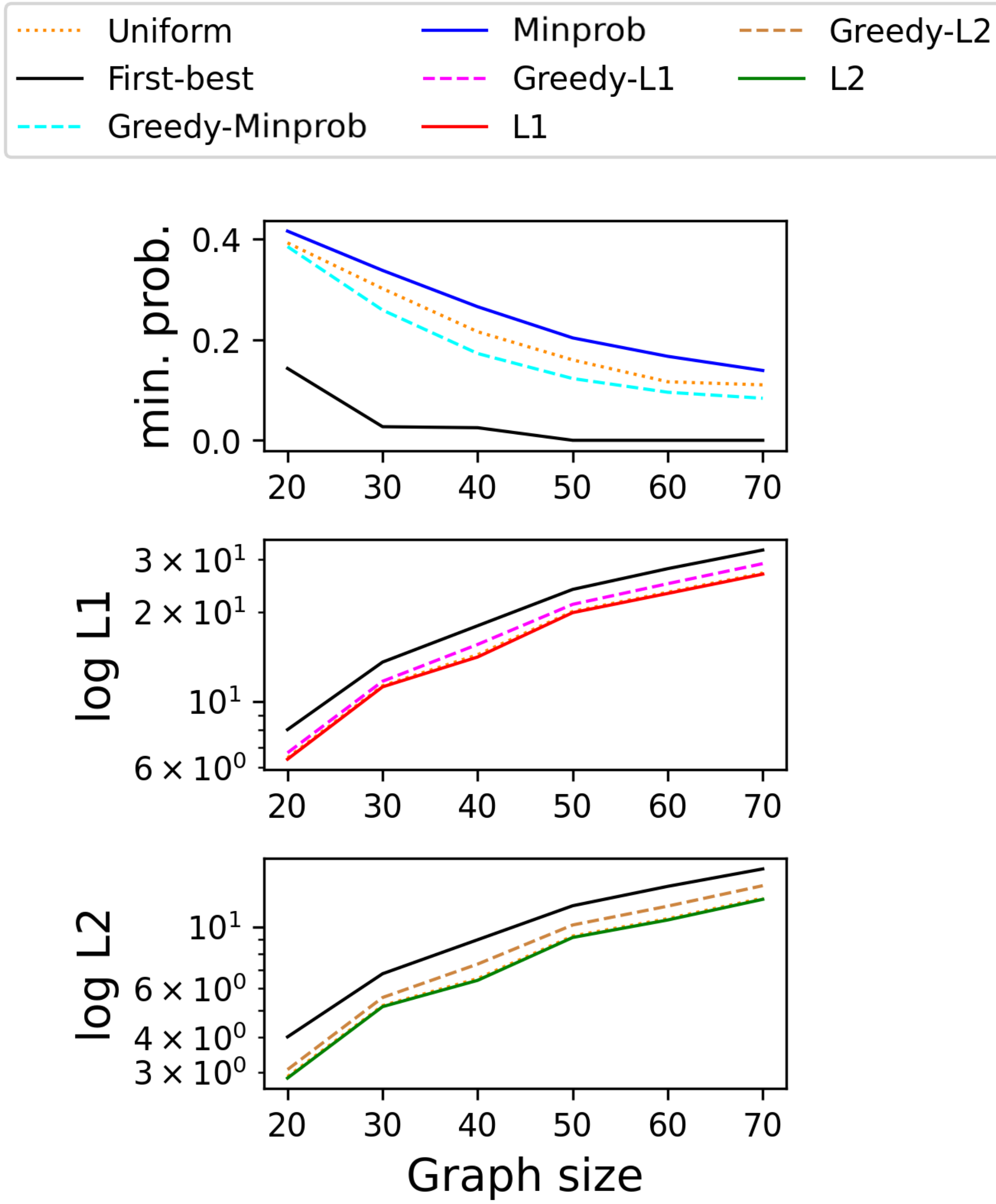


Fig. 3.1. Comparison of the optimal distributions corresponding to our metrics and first-best

(1) Group fairness measure using the α value. The α value is the number of hard-to-match patients (PRA $\geq 80\%$) that are included after selecting an exchange plan (or the expectation of this number). As a reference point, *First-best* is provided as a baseline to see how IF performs since GF will be the highest value (it maximizes the α value);

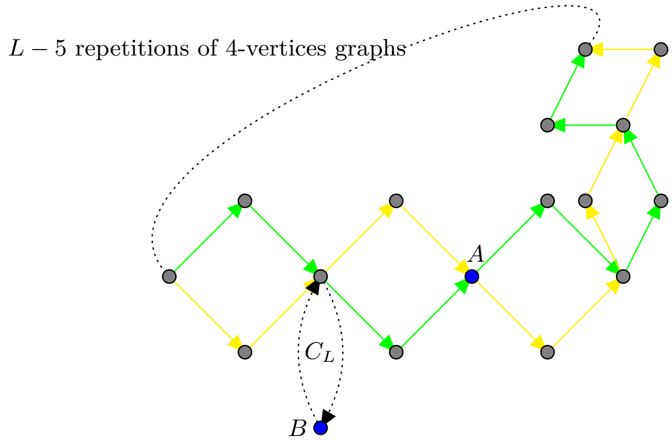


Fig. 3.2. Worst-case scenario for uniform distribution when compared to L_p-norm

(2) Optimality measure using the number of transplants;

(3) Individual fairness measure using the log L_2 value. *First-best* is provided as a baseline to see how GF performs since IF will be the lowest value (it minimizes that value).

The standard deviation was also computed for each mean value. The results can be found in Table 3.1. It can be observed that IF is comparable to *First-best* in terms of the GF measure

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 3.1. Comparing group fairness (GF) with individual fairness (IF) over the INESC TEC dataset

(i.e. α value in the table). It also has a higher number of transplants than GF.

3.5. Scaling methodology

Now that we have introduced the concept of distributions over a set of solutions S, we can discuss a more efficient way of enumerating solutions. Because we now have a metric over the set $\mathcal{P}(S)$, we will see that in certain instances, optimizing with respect to this metric lends itself to the method of column generation. For example, suppose that we are working with the L_1 metric. The problem of finding the distribution δ that minimizes $L_1(\delta)$ can be formulated as:

$$\begin{aligned}
\min \sum_{v \in P'} d_v & \\
\text{s.t.} \quad \sum_{s \in S} \delta_s = 1 & \\
y_v = \sum_{s \in S | v \in \pi(s)} \delta_s & \qquad \forall v \in P' \\
z = \frac{1}{|P'|} \sum_{v \in P'} y_v & \\
d_v \geq \sum_{s \in S | v \in \pi(s)} (\delta_s - z) & \quad \forall v \in P' \\
d_v \geq \sum_{s \in S | v \in \pi(s)} (z - \delta_s) & \quad \forall v \in P' \\
\delta_s \in [0,1]. &
\end{aligned} \tag{δ-LP}$$

This is a linear program, so one is able to use column generation [25]. Nevertheless, we need to broadly describe what is meant by column generation. We will do this for the L_1 metric as in the experiments of Section 3.3, the metric lead to a good improvement over *First-best*. Suppose that we have an LP given by the program (δ-LP). We observe that the constraints of (CF) form a polytope since they constitute a bounded polyhedron. Hence, finding extreme points of this polytope is sufficient since every solution will be a convex combination of these extreme points x^s.

$$\begin{aligned}
\min \sum_{v \in P'} d_v & \\
\text{s.t.} \quad \sum_{s \in S} \delta_s = 1 & \qquad (\mu_0) \\
y = \sum_{s \in S} (Ax^s)\delta_s & \qquad (\mu) \\
z = \frac{1}{|P'|} \sum_{v \in P'} y_v & \\
\delta_s \in \{0,1\} & \qquad \forall s \in S \\
y_v \in \{0,1\} & \qquad \forall v \in P',
\end{aligned} \tag{M}$$

where A is the $|P| \times |\mathcal{C}|$ matrix defined by

$$A = \begin{cases} 1 & \text{if } v \in c \\ 0 & \text{otherwise} \end{cases}$$

The restricted version of (M) is nearly identical: it uses a subset of the set of solutions S. Any subset will do, but experimentally, one usually starts with a single solution to (δ-LP). Using the previous result, one obtains the subproblem

$$\begin{aligned} \min -\mu A x - \mu_0 \\ Ax \leq 1 \\ \sum_{c \in \mathcal{C}} x_c \geq OPT \\ x_c \in \{0,1\}, \end{aligned} \tag{SP}$$

where the last constraint essentially ensures that only optimal solutions are enumerated (OPT being a constant found after solving the initial problem (CF)). If the objective of (SP) is greater or equal than 0, the algorithm terminates since there is no new solution s that will improve the objective of the restricted version of (M). Otherwise, the column Ax^s is added to the restricted version of (M). Each column that is generated will correspond to a solution s. We do not have to enumerate all the solutions to find the optimal distribution: simply the ones that will have non-zero probability (and maybe a few more that do have zero probability). The (full) solution vector δ will also tend to be sparse, which further strengthens the need for such an approach. As a remark, it can be said that this column generation method can be used with any KEP formulation, even (PIEF) [10].

Experimentally, we evaluated this column generation approach over the UNOS dataset. The goal was to determine if the approach could scale well to larger graphs than those found in the INESC TEC dataset. Again, the results were averaged over all instances of the UNOS dataset over each particular graph size. Figure 3.3 tells the same story as previously observed: the fair L_1 and L_2 optimizations of IF give a noticeable improvement over *First-best*. The trend thus seems to grow larger as the size of the graphs increases. Because column generation only makes use of relevant solutions to the final optimal distribution over S, the enumeration of solutions is limited to those. The sparsity of the L_1 metric is highlighted here as there is only a need to enumerate very few solutions out of the large pool of feasible solutions. In fact, we can see from Figure 3.4 that the column generation approach solves instances relatively quickly on average. It can be observed that for smaller graph sizes, the running time is significantly low (less than 100 seconds on average) and there is a cusp at the 256 mark. The fact that we were not able to complete enumerating solutions for such large graphs indicates that the column generation approach can indeed be very useful to scale IF

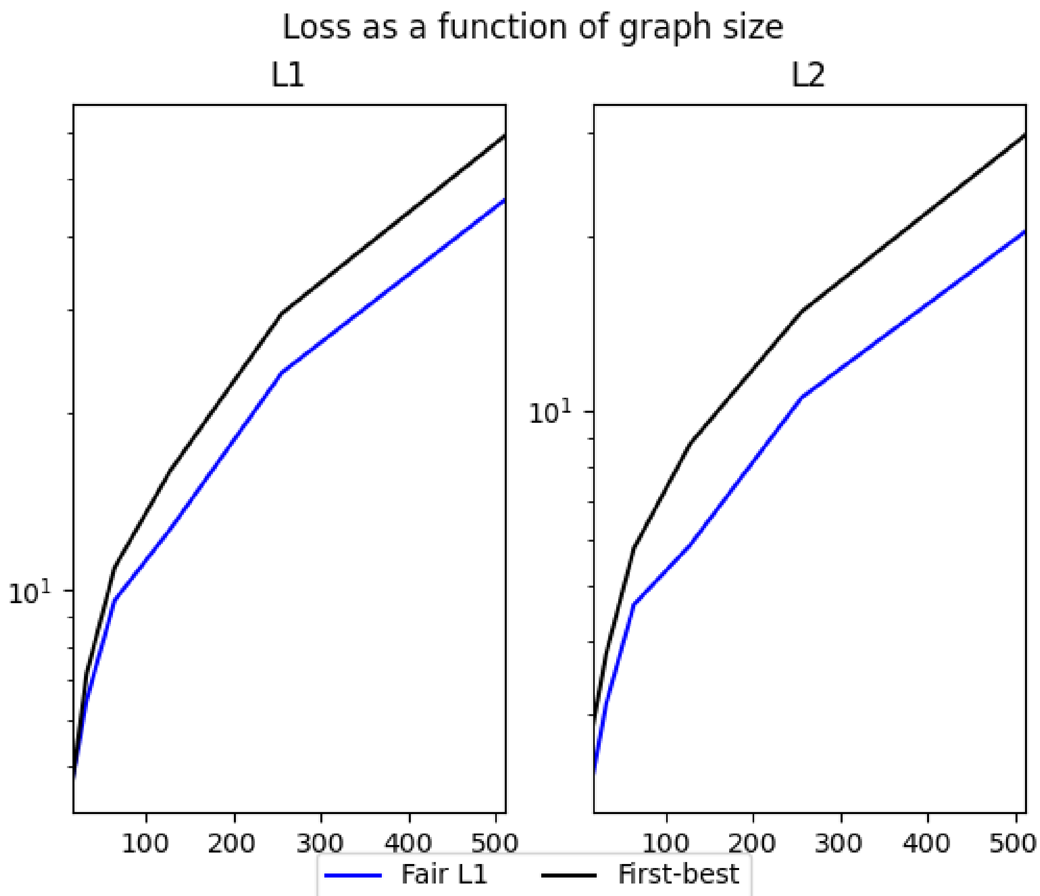


Fig. 3.3. Evaluating the L_1-optimal distribution over the L_1 and L_2 metrics against *First-best*

to large graphs. It is possible to improve on these results by using the (PIEF) instead of the (CF).

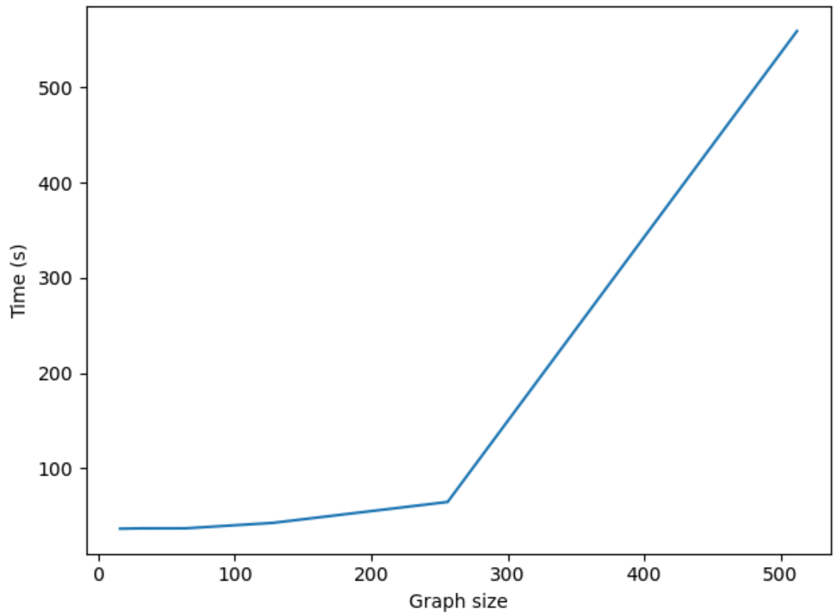


Fig. 3.4. Average time to solve (CF) using column generation as a function of graph sizes

Chapter 4

Price of fairness

In KEPs, we are faced with a limited amount of resources (the donors' kidneys) and conflicting preferences or desires (the individual agents). How do we distribute the kidneys? One obvious suggestion would be to look at a utilitarian mechanism where we maximize the expected number of successful transplants. However, it is easy to imagine that certain patients are going to be heavily favoured by such an approach. Just imagine a person that is highly compatible to others. We can think of a universal recipient blood type patient (i.e. AB-positive) with low PRA. If in addition, the patient is young and otherwise healthy, most donations will tend to be successful with that patient in terms of longevity of the graft. This will not be the case for highly sensitized patients like those with O-negative blood type. We can consider Figure 4.1. Patient A is in many solutions but not patient B. Therefore, de-

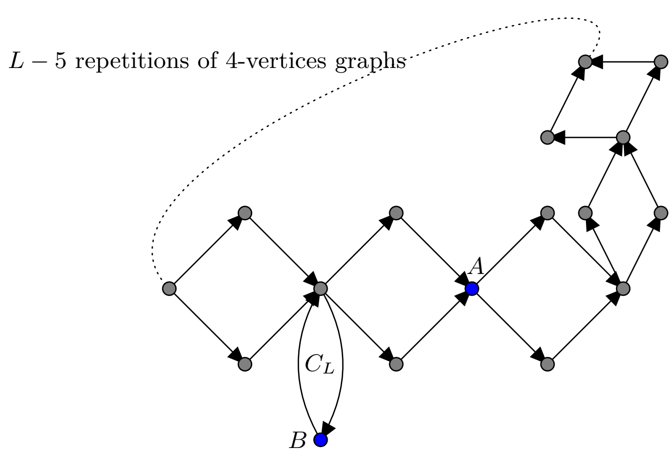


Fig. 4.1. Many maximal exchanges involving patient A, but few involving patient B

pending on the method used to obtain a maximal exchange plan, we might end up choosing patient A more often. This cannot be considered fair for patient B, especially when picking that patient does not sacrifice maximal utility. The method that we devised earlier takes care of this problem by balancing the probabilities of each patient. However, we were dealing with only optimal exchanges. What if a patient is never in an optimal exchange plan (in the utilitarian sense)? One obvious answer is to simply relax the notion of optimality when looking for a suitable exchange plan (see Figure 2.2). In other words, we sacrifice optimality to allow for a fairer scheme. But how do we select the allowed relaxation threshold? Is there an alternative that takes care of balancing utility and fairness? In fact, we can often view fairness and utility as two competing objectives. If we have a measure or metric that is associated to both objectives, we can examine the cost of prioritizing one over the other (and vice versa). In Bertsimas et al. [7], this is referred to as *price of fairness*.

Definition 4.0.1. *A utility set U given by a vector-valued function f is defined as*

$$U = \left\{ u \in \mathbb{R}_+^{|P|} \;\middle|\; \exists x \in X : u_j = f_j(x), j \in P \right\}$$

*where $X \in \mathbb{R}_+^m$ is the **resource set** (i.e. the constraints) and $f_j$is the utility function of patient j.*

Definition 4.0.2. *A fairness scheme $\mathcal{L} : 2^{\mathbb{R}_+^{|P|}} \to \mathbb{R}_+^{|P|}$ is defined by*

$$\mathcal{L}(U) \mapsto u$$

for some $u \in U$ for each utility set U.

Defintion 4.0.2 can be thought of as assigning a fair outcome to each set of attainable utilities. We introduce the two following definitions before formalizing the price of fairness:

Definition 4.0.3. *The system score with utility set U (or $SYSTEM(U)$) is given by*

$$SYSTEM(U) = \sup_{u \in U} e^T u,$$

where e is a vector of ones of the appropriate dimension.

Definition 4.0.4. *The fairness score of utility set U with loss $\mathcal{L}$ (or $FAIR(U, \mathcal{L})$) is given by*

$$FAIR(U, \mathcal{L}) = e^T \mathcal{L}(U)$$

The *Price of Fairness* is then defined as

Definition 4.0.5. *Price of Fairness*

$$POF(U, \mathcal{L}) = \frac{SYSTEM(U) - FAIR(U, \mathcal{L})}{SYSTEM(U)}$$

We now possess the vocabulary necessary to discuss the tradeoff between a fairness scheme and the utilitarian objective. In order to do so, we must examine various fairness schemes. The most well-known are due to Aristotle, Rawls [12], Nash and Shapley [8]. We will discuss their nature and how they apply to KEPs, as well as how these principles can be formally described mathematically.

4.1. Aristotle's equity principle

Aristotle's equity principle [7] is based on the idea of individuals having pre-existing rights to the resources. People should then receive what is rightfully theirs based on these claims. We can simply think of a person holding stocks that pay dividends. The share of dividends that are paid back to its stockholders will be distributed proportionally according to the share of the stocks that each person owns. When transposing this idea to KEPs, however, we immediately see an obstacle in the competing claims to access the limited amount of resources (i.e. kidneys). How should we prioritize a person over the other? Should an individual really have priority over another one and if so, when? What constitutes a stronger claim for a transplantation over another one? These are questions that naturally arise in this context. We can try to answer some of them by giving a ranking to the patients according to their need, or even the time they spent waiting in the program to obtain a kidney. Indeed, it intuitively makes sense to fill a patient's need for a compatible kidney over another if the first has waited multiple rounds to receive one. However, we cannot also ignore the potential urgency of this kidney transplantation for the second patient. Hospitals are confronted every day with this problem, ranking patients according to their need for medical treatment through a system called triage [26, 27]. The main objective being to save lives, the severity of the illness justifies prioritizing some patients over others. Secondly, this is balanced with the goal of maximizing total utility: if a doctor spends too much time treating a person whose chances of survival are very low at the cost of losing many other patients whose life could easily have been saved, then the doctors are faced with the crucial dilemma of treating patients according to their individual need versus maximizing the overall well-being of the patients.

To describe Aristotle's principle of fairness mathematically, we would need some sort of scoring mechanism to determine how to rank or order patients. Ideally, this scoring mechanism would be time-dependent: the longer a patient waits to obtain a kidney in the program, the higher this score becomes. Formally, we have a function $\rho(y_v, t)$ where y_v is a feature vector for patient $v \in P$ and $t \in \mathbb{R}$, with the following property:

$$\forall v \in P, \quad t_2 \geq t_1 \implies \rho(y_v, t_2) \geq \rho(y_v, t_1) \tag{4.1.1}$$

One can observe that through the dependence on $v \in P$, the function ρ allows for each patient's characteristics to influence their priority score. One can now maximize the total score over all patients in a selected KEP exchange. Mathematically adapting (CF), this is given by

$$\begin{aligned} \max_{x,z} \quad & \sum_{v \in P} \rho(y_v, t) z_v \\ \text{s.t} \quad & z_v \leq \sum_{c \in \mathcal{C}: v \in c} x_c \leq 1 && \forall v \in P \\ & z_v \in \{0,1\} && \forall v \in P \\ & x_c \in \{0,1\} && \forall c \in \mathcal{C} \end{aligned}$$

Similarly, one can rate the patient's health status and give priority to patients whose health is declining the most and hence, would highly benefit from a transplant. This takes care of the resource claim argument: the higher a patient's need for a kidney, the higher their claim on it. With this in mind, the goal is to maximize the total gain in the patients' health status. In this particular context, we have another function $\rho'(y_v, t)$ that is real-valued but is not required to have the defining property 4.1.1 of ρ. We now solve the following problem:

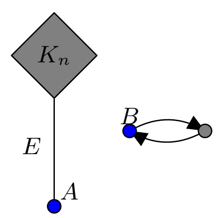
$$\begin{aligned} \max_{x,z} \quad & \sum_{v \in P} |\rho'(y_v, t_2) - \rho'(y_v, t_1)| z_v \\ \text{s.t} \quad & z_v \leq \sum_{c \in \mathcal{C}: v \in c} x_c \leq 1 && \forall v \in P \\ & z_v \in \{0,1\} && \forall v \in P \\ & x_c \in \{0,1\} && \forall c \in \mathcal{C}, \end{aligned}$$

where t_2 is the current time and t_1 precedes it. With this approach, we still seek to maximize the total score but we still give precedence to patients whose health is increasingly at risk if no transplantation is performed. At the same time, we are not giving complete priority to individuals that are in poor health but whose chance of survival is close to 0. This is in accordance with doctors' behaviour, under limited amounts of resources. Nevertheless, we are faced with a very tough ethical and moral dilemma: how can one select the patients to operate on and will this choice be unbiased? When dealing with life or death situations, the lines become blurred between what is optimal and what is socially acceptable. Maybe other approaches are better suited than Aristotle's fairness principle at finding the right distribution of resources. No matter how confident one might be in their ability to properly determine people's claims on the resource, the very individuals that are excluded under this principle will certainly disagree with it. It then becomes a question of how we can achieve some form of consensus where even the excluded individuals would agree with the underlying principles. This is what John Rawls and the *veil of ignorance* attempt to achieve.

4.2. Rawls and the Veil of Ignorance

In Rawls [12], the author introduces the concept of the *veil of ignorance*. With this veil, every individual is not made aware of their situation, biases, view on the world. The only thing that remains is knowledge about the world and their rational mind. Rawls argues that under this veil of ignorance, individuals can negotiate the terms of a social contract that seems inherently fair.

One way of approaching such a decision is to consider that individuals will then maximize the utility of the least well-off person. Since there would be no way for them to know whether or not they would be that person under the veil of ignorance, it would seem to be the rational solution to adopt. In mathematical terms, this would correspond to maximizing the *minprob* metric that we introduced previously in section 3.2. Doing so guarantees some form of robustness to our solution selection mechanism. A natural question to consider is then: "what is the price that we have to pay with respect to utility?" The Rawlsian approach does not care about that cost because it is the only rational strategy to adopt under the veil of ignorance. However, for every policy maker interested in maximizing utility as much as possible, the *minprob* optimization might prove too costly in some instances. We can see this in Figure 4.2. Vertex A is found in $(n+1)!$ feasible solutions, while vertex B is only found



$$E := \{a \mid a = (A, v) \vee a = (v, A) \quad \forall v \in K_n\}$$

Fig. 4.2. Worst-case scenario for Rawlsian fairness. K_n is a clique of size n

in 1 feasible solution. Thus, under an optimal distribution for this graph (with the *minprob* metric), we would see the vertices in the large cycle (except for A) have a probability of $\frac{1}{2}$ of being selected and vertex B would also have a probability of $\frac{1}{2}$ of being selected. Under this optimal that optimal distribution, we get $\frac{n+3}{2}$ expected transplants. However, if we were to sacrifice patient B's probability of receiving a transplant, we would get a maximal number of transplants of $n+1$. In that case, we can sacrifice a significant number of transplants when the value of n is large. Indeed,

$$\begin{aligned} POF =& \frac{n+1-\frac{n+3}{2}}{n+1} \\ =& \frac{n-1}{n+1} \\ \lim_{n\to\infty} & \frac{n-1}{n+1} = 1, \end{aligned}$$

and in that case the POF is close to 1 for large values of n. We therefore need to consider whether this is something that is acceptable, and whether we can justify "sacrificing" the chance of one patient to the advantage of others.

Using a similar approach to fairness in KEPs, another rational action to consider is to prioritize patients that have a greater need for a transplant. Here, the need would be defined by how much could be gained in terms of one's health when receiving a transplant. This seems to go in the same direction as the general approach to triage in hospitals: patients that are in a critical state but that can be saved by being treated usually are prioritized over others. If a patient is unlikely to survive even after treatment, that person will not be prioritized over someone that has a better chance of recovery. Inherently, this does not take into account the characteristics of the person, just their potential health gain. This is the key difference from the previous section: while Aristotle's principle would allow for characteristics of individuals to guide the prioritization of patients, Rawlsian fairness forbids this. The veil of ignorance would act as an ideological barrier to such an approach. On the contrary, it would seem reasonable and in everyone's best interest to agree with the premise that the resources should go to individuals with the greatest need.

We would therefore need to evaluate the potential health benefit to a person. This can be achieved through a score of the person's health. For each patient, we could have a score $\alpha \in [0,1]$ that corresponds to the probability of death from kidney failure. If a person receives a kidney and the transplant is successful, then their α value would now be equal to 1. Each person's score depends on whether or not they receive a kidney.

4.3. Nash standard of comparison

John Nash, in his paper on cooperative bargaining games [11], introduces his concept of fairness. For a two-player game, it measures the relative change in a player's utility when receiving a small amount of resources. If that amount is greater than the relative loss of utility of the other player, then this transfer of resources is justified. This fairness scheme satisfies the four important axioms of *Pareto optimality*, *symmetry*, *affine invariance*, and *independence of irrelevant alternatives*. It can be generalized to more than two players.

Definition 4.3.1. *Under **proportional fairness**, an allocation $\mathcal{L}(U)$ is fair if*

$$\sum_{j \in P} \frac{u_j - \mathcal{L}(U)_j}{\mathcal{L}(U)_j} \leq 0,$$

for any $u \in U$, where P is the set of players.

In simpler terms, for an allocation $\mathcal{L}(U)$ to be fair, any other allocation decreases in proportional utility (with respect to proportional fairness) when aggregated over all players. One consequence of proportional fairness is that a loss of utility to one player is acceptable when that player already has accumulated a lot of resources that would suit another player whose utility is much smaller. Intuitively, this corresponds vaguely to the redistribution of wealth by taxing high income and redistributing it among the lower end of the spectrum. This income analogy will be useful to describe Nash's principle by fixing utility to a tangible concept for the reader. While this might seem unfair or unreasonable to do, it can be justified using a utilitarian argument. It can be argued that individual's utility functions are not linear as a function of their income but rather concave. Thus, higher-income individuals have a smaller marginal utility gain for a fixed amount of x \$ that they earn when compared to lower income individuals. As a sidenote, it can be observed that this is common way of resolving the St. Petersburg paradox [28, 29].

Since utility functions that are linear in terms of the money earned lead to the paradox, it would seem rather undesirable to use such functions. This fact can thus be seen as a starting point or basis to justify the use of concave utility functions. In KEPs, individuals that have a high survival chance or are otherwise in good health can be seen as the rough equivalent to higher-income individuals: they will benefit less by obtaining a kidney than a person dangerously at risk of dying if a transplant is not performed immediately. This does not mean they should not be operated on, simply that the marginal gain in utility is less and thus it would be entirely justifiable under Nash's principle to favour those that would obtain a higher marginal utility gain. Fairness, according to Nash, can be justified simply as a matter of marginal utility maximization.

When we are looking to apply this fairness scheme to KEPs, the major issue that arises is definitely related to the utility functions of patients. We could take the simplest form, which is indeed the one used in the basic formulation of a KEP, and define the utility of a patient as 1 if they get a compatible kidney and an arbitrarily small $\epsilon > 0$ otherwise. We use ϵ simply to take care of the fact that in this section, we assume the utilities are always strictly positive. However, we can see that under this choice of utility function, whenever we obtain a proportionally fair allocation, it corresponds to an allocation that maximizes total utility. This signifies that the POF is equal to 0, but we are left with the common issues related to the utilitarian approach, which will be discussed later in Section 4.7. We could therefore try to have utility functions that do not exhibit such symmetry over players (i.e. being all equivalent). Having a score for the quality of the kidney that a patient receives would help break this symmetry by providing more granularity to the utilities. This also seems more realistic because it will probably hardly be the case that two different patients receive two kidneys of the exact same quality relative to their needs. If it is the case, it is probably because the analysis of the quality of the organ is not fine-grained enough. Nevertheless, considering this symmetric case as valid certainly makes for interesting research questions. By inspection, we might be able to abstract and generalize Nash's concept of fairness for all cases (even symmetric ones) with the same goal of always maximizing the marginal utility under the allocation of resources.

4.4. Shapley values

Another natural approach would be to use Shapley values [8], which have been applied previously in KEPs [30]. In the context of a cooperative game theory, Shapley values provide a measure of agents' contribution to a coalition. A coalition is simply a subset of players that decide to cooperate between themselves. For an n-player game and a set function $\nu : 2^N \to \mathbb{R}$ where $N = \{1,\dots,n\}$, the value $\nu(S)$ of a coalition $S \subseteq N$ represents the total utility achieved by the coalition S. Shapley's function is the unique (n-)vector-valued function ϕ that satisfies the following three axioms [8]:

(i) **Efficiency**: $\sum_{i\in S} \phi_i(\nu) = \nu(S)$ for any carrier S of ν. A carrier of ν is a coalition T such that $\nu(T \cap S) = \nu(S)$.

(ii) **Symmetry**: $\phi_{\pi(i)}(\pi\nu) = \pi(\nu)$ for any permutation π of the set N. The game $\pi\nu$ is taken to be the game u defined by $u(\{\pi(i_1),\dots,\pi(i_s)\}) = \nu(S)$ for all $S = \{i_1,\dots,i_s\} \subseteq N$.

(iii) **Linearity**: $\phi_i(u + \nu) = \phi_i(u) + \phi_i(\nu)$ for all $i \in N$ and any games u and ν.

Definition 4.4.1. *The unique function ϕ that satisfying axioms* (i) *to* (iii) *is defined by*

$$\phi_i(\nu) = \sum_{T\subseteq N | i\in T} \frac{(|T|-1)!(n-|T|)!}{n!} \left(\nu(T) - \nu(T \setminus \{i\})\right)$$

The logical interpretation behind the definition of Shapley values is associated with the marginal contribution of each player under a random arrival process to the coalition. Thus, if the arrival of each player to the coalition is uniformly random, one is able to evaluate the marginal contribution of player i for each possible case. Supposing that there is a coalition S such that $i \notin S$, player i contributes $\nu(S \cup \{i\}) - \nu(S)$ to the newly formed coalition. Averaging over all possible orderings, one gets the expected marginal contribution of each player to each possible coalition including that player. When transposing this to KEPs, one can think of a player as a patient-donor pair and the coalition as the set of pairs that choose to enter the KEP. If a pair has a high Shapley value, then their inclusion in a KEP exchange plan contributes to a marked increase in the number of transplants that can be performed. Using this idea, one can seek to fairly reward patient-donor pairs for their contribution to the KEP pool. This can take the form of the addition of a second weighted objective to (CF):

$$\begin{aligned} \max \sum_{c\in\mathcal{C}} w_c x_c + \lambda \sum_{i\in P} \sum_{c\in\mathcal{C}|i\in c} \phi_i x_c & \\ \text{s.t.} \quad \sum_{c\in\mathcal{C}|v\in c} x_c \le 1 & \qquad \forall v \in V \\ x_c \in \{0,1\} & \qquad \forall c \in \mathcal{C}, \end{aligned}$$

where $\lambda > 0$ is a weight that is chosen beforehand for the KEP. This method intuitively combines the objective of transplant maximization, as well as maximizing the Shapley values of the selected individuals. The latter objective seeks to reward the pairs that contribute the most to the overall utility of the system. This method, however, runs into a few hurdles. First, there is the issue of selecting the appropriate weight λ. Is there an obvious candidate? Does it have to be selected experimentally? And then, there is the issue related to the Shapley values themselves: they are hard to compute. Indeed, the sum involved in their definition is over all subsets S of P that contain a particular player i. Hence, there are $2^{|P|-1}$ candidates subsets to consider for each player. This is why applications of Shapley values to KEPs have mostly been concerned with specific cases where the number of players is small, such as in multi-country KEPs [30] (the players are the countries).

4.5. Discussion of fairness schemes

By discussing these various fairness schemes, it is possible to see which ones would be more easily adaptable to KEPs. While both Rawls and Nash's approaches can be useful, they possess flaws. Rawlsian fairness could potentially lead to a massive POF, which would render its application undesirable (see Figure 4.2). Nash's fairness principle, on the other hand, has some very desirable properties (see Section 4.3) but its utilitarian approach fundamentally still implies that some patients could suffer greatly under that principle. Only by considering distributions over solutions is it possible to give a chance to patients not included in utility-maximizing solutions. This is exactly what motivates IF. Then, it might be interesting to ask how one can add this added fairness component without paying too high a price in terms of utility (i.e. having a low POF). In Chapter 5, we will introduce an interesting concept to tackle fairness in KEPs: the *Nash Social Welfare Program*. It provides the ability to balance multiple objectives and thus, it can be applied to fair procedures in KEPs to restrict their POF. It will become useful to compare this method to the various fairness schemes discussed in this section.

4.6. Comparison between fair procedure and fair outcome

In Bolton et al. [31], the authors compare two fairness schemes that arise naturally in the context of games: fair procedures and fair outcomes. Three types of simple games are played between pairs of individuals. These games are referred to as Sequential Battle of the Sexes (BOS), Ultimatum Game (UG) and Sequential Battle of the Sexes with Fair Procedure (BOSFP).

[1] Figure taken from Bolton et al. [31].
[2] Idem.
[3] Idem.

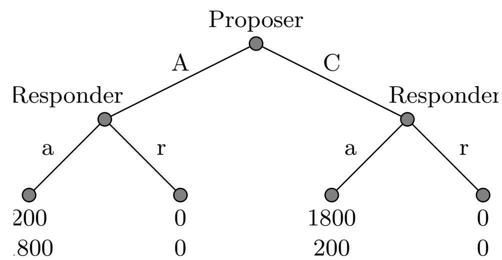


Fig. 4.3. Sequential Battle-of-the Sexes Game (BOS)[1]

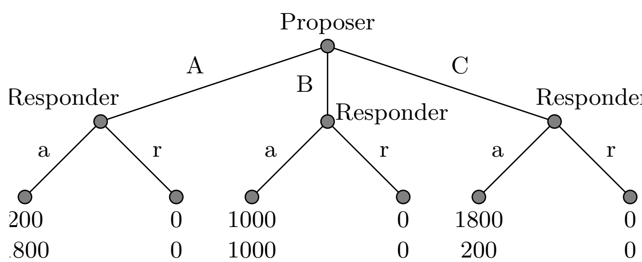


Fig. 4.4. Ultimatum Game (UG)[2]

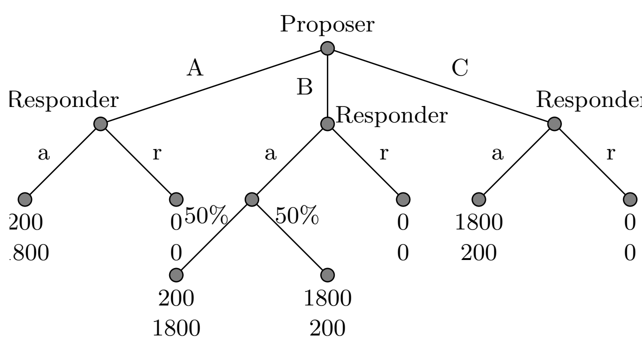


Fig. 4.5. Sequential Battle-of-the Sexes Game with Fair Procedure (BOSFP)[3]

In BOS (see Figure 4.3), player 1 is asked to make 3 offers to player 2: option A benefits player 2 and option C is beneficial to player 1. In each case, player 2 can refuse and both get no payoff. Naturally, player 1 will be tempted to offer choice C to player 2. The authors experimentally show that player 2 refuses offer C in 6% of the time.

In UG (see Figure 4.4), the game setting is similar to BOS, but there is another offer B available: both players equally benefit if this offer is accepted by player 2. The authors show that in this setting, the rejection of offer C (i.e. beneficial to player 1) is higher (40%) since there is a “fair” alternative where both players get equal payoffs. It can be assumed that offer C is now seen as unnecessarily selfish and therefore player 2 prefers to refuse it. For BOS, since that option is not available, player 2 will accept more readily option C, probably because there is no scenario of a fair outcome. The authors conclude that an unfair offer is more tolerable to the subjects of the study if no fair outcome is available.

The last game, BOSFP (see Figure 4.5), is similar to UG, but the offer B is rather a lottery, where players get the payoffs of offer A and B with equal probabilities. Again the rejection rate for offer C (i.e. beneficial to player 1) is similar to UG.

The authors conclude that a fair procedure can thus be thought of as substitute to a fair outcome. It becomes harder to tolerate an unfair offer when either a fair procedure or a fair outcome is available. A fair procedure can better be described as a procedure or a set of rules such that when followed, every player (or most of them) are satisfied with the “fairness” property of the game. A fair outcome rather focuses on whether the players are satisfied with the outcome or result of the game. The discussion can thus be framed in terms of the dichotomy between a fair procedure and a fair outcome. The former can generate situations where the payoff of each individual (when framed in the language of games) is not equal. However, in expectation these payoffs will be the same, which makes it seem acceptable to individuals. This is what IF attempts to achieve. Not every exchange plan will satisfy every patient, but at least, they will have a somewhat comparable chance of receiving a kidney. The optimization process makes this probability as evenly distributed as possible when using L_p metrics. On the other hand, a fair outcome approach will rather focus on generating a set of payoffs that will be uniformly distributed. In KEPs, this would take the form of an exchange plan that is acceptable to all the participating parties because it satisfies desirable characteristics and is thus a fair alternative (or middle ground). If we think of it in terms of the utility of agents, both approaches try to balance the utilities of the agents; the distinction being that a fair procedure deals with the expected utility.

In the context of the games played in Bolton et al. [31], the “risk” of getting a lower payoff under a fair procedure is counterbalanced by the chance of receiving a higher payoff. We can imagine that this means the agents are totally risk neutral, since this procedure is as acceptable as the fair outcome approach. It is worth mentioning that in Bolton et al. [31], the expected payoffs for both games (UG and BOFSP) were set to be the same. However, things can get a little bit murky when we transpose this discussion in the context of KEPs. We are ultimately dealing with the selection of patients for an exchange plan. It must be stressed that this is often a question of life or death. Therefore, does it seem plausible that patients would be risk neutral under a fair procedure? Even if it is not the case, how do we deal with the issue of limited resources, here being compatible kidneys for each patient? We will have to prioritize certain patients, while leaving out others, even if this is done through a lottery. On the other hand, if we completely reject the notion of using a fair procedure, and simply focus on a fair outcome, we might sacrifice greatly the total utility of the system. Just imagine a scenario where no one gets a kidney: it is surely not unfair in the sense that no one is prioritized over the other, but then it does not make a lot of sense to simply refuse to perform any transplant at all. Finding a good balance between the utility of a solution and some measure of fairness of that solution will be explored later in Chapter 5.

4.7. Limitations of the utilitarian approach

In addition to the discussion surrounding fair procedures and outcomes, another important detail to explore concerns the objective function that is optimized in a KEP. Before we define a metric that must be optimized, we must first consider the drawbacks of the utilitarian approach. Let us consider the following thought experiment: person A, who is very sick, can swap donors with person B to receive a kidney and is expected to recover fully. However, person B can instead enter in an N-way exchange with $N-1$ other pairs; the total utility would be N instead of 2. Suppose that person B and the other $N-1$ pairs are relatively healthy: their condition is in the early stages and they are otherwise very healthy individuals. Would it make sense to sacrifice person A so that the rest of the group benefits? Probably not.

Each individual has a utility function that is dependent on their personal characteristics. Whether we can guess a “true” utility function when disregarding everyone’s biases and particular circumstance is an interesting question. If this was possible, we could have an unbiased method to evaluate the priority of patients when assigning exchange plans. This would fall under Aristotle’s fairness principle as discussed in Section 4.1. Nevertheless, we can still implement a system where patients with a higher priority are indeed prioritized at fair value. By using this system of prioritization, we can now combine both the utilitarian approach and the aforementioned prioritization to balance utility with fairness. To do so, Chapter 5 will introduce an important concept in multi-objective optimization (MOOP): the *Nash Social Welfare Program* (or NSWP).

Chapter 5

The Nash Social Welfare Program (NSWP)

In Section 5.1, we discuss the terminology associated with MOOPs and how we can balance fairness and utility. Section 5.2 discusses the Social Welfare Program and its various flaws. Next, the Nash Social Welfare Program is introduced in Section 5.3 to address the weaknesses of the SWP. Some properties of the NSWP are also discussed and we apply the NSWP to KEPs, combining it with the concept of IF. In Section 5.4, we extend the formulation of Section 5.3 to scale to larger graphs. Finally, Section 5.5 presents multiple experiments that were realized in order to evaluate the effectiveness of the NSWP in balancing IF and utility.

5.1. Competing objectives

The main issue that arises when discussing fairness in the context of KEPs is the opposition of the two main objectives that we aim to optimize: maximization of the total efficiency of an exchange plan and maximization of the benefit for the least advantaged patients. It is easy to find instances where these two objectives are competing against one another. There is generally no ideal solution capable of maximizing both of these objectives simultaneously.

Hence, we enter in the field of multi-objective optimization (MOOP), where we search for Pareto efficient solutions. The set of all Pareto efficient solutions is called the Pareto frontier, Pareto front or Pareto set. We introduce these concepts in the following definitions.

Definition 5.1.1. *Consider a vector-valued objective function $f : X \to \mathbb{R}^k$. A solution vector x (Pareto) dominates another solution y (for a maximization problem) if*

(1) $f_i(x) \geq f_i(y) \quad \forall i \in \{1,\ldots,k\}$;

(2) $\exists j \in \{1,\ldots,k\}$ s.t. $f_j(x) > f_j(y)$.

We denote this dominance as $x \succ y$.

Definition 5.1.2. *A Pareto optimal solution is a solution x that is not dominated by any other solution y. In other words, $\nexists y$ such that $y \succ x$.*

Definition 5.1.3. *The set of Pareto optimal solutions is called the Pareto frontier (or Pareto front, Pareto set). It can also be written as*

$$P^* := \{y : \nexists z \ s.t. \ z \succ y\}.$$

With these useful definitions, we can introduce the concept of ideal vector. Each component of this vector corresponds to the best value that can be attained by the different components f_i's of the objective function f.

Definition 5.1.4. *The ideal vector z^{ideal} is defined as*

$$z_i^{ideal} = \sup_{x \in P^*} \{f_i(x)\} \quad \forall i \in \{1, \dots, k\}.$$

Definition 5.1.5. *The nadir vector z^{nadir} is defined as*

$$z_i^{nadir} = \inf_{x \in P^*} \{f_i(x)\} \quad \forall i \in \{1, \dots, k\}.$$

Now, we possess the language to discuss solutions methods to MOOPs.

5.2. The Social Welfare Program (SWP)

As mentioned in the previous section, if we have multiple objectives f_i that we aim to maximize subject to a set of constraints, we are faced with the problem of computing the Pareto frontier. A common workaround is to attribute weights to each objective and then maximize the single (aggregate) objective. A solution is then guaranteed to be Pareto optimal. While this is a reasonable idea, we will see that it has important pitfalls associated with it. To this end, we first introduce the Social Welfare Program formally:

Definition 5.2.1. *The Social Welfare Program is any optimization problem of the form*

$$\max \sum_{i=1}^{k} w_i f_i(x)$$
$$s.t. \quad x \in X$$

where X is the feasible region (can be defined by a set of constraints) $\{f_1, \dots, f_k\}$ is the set of objectives to be maximized. The $w_i > 0$ are the positive weights associated with each objective.

We will later see a value of 1 for each w_i being used in the SWP for KEPs. There are inherent weaknesses that come with this approach. As discussed in [9];

- **Weakness 1**: Computing a Pareto solution is often hard to do or simply intractable.
- **Weakness 2**: In some cases, there is no decision-maker or it is not obvious how to select a (Pareto-optimal) solution from the Pareto frontier.
- **Weakness 3**: There can be multiple Pareto-optimal solutions that cannot be obtained by the SWP. These points are called *unsupported* Pareto-optimal points/solutions.

When applying this framework to KEPs, the two objectives that arise naturally are:

(1) maximizing the number of transplants;

(2) optimizing a fairness criterion to minimize biases in the selection of an exchange plan and their negative impact on some patients.

Computing a Pareto solution is therefore not hard (though still NP-hard for IPs): we can simply find a solution that satisfies either objective to optimality and then, we can add a constraint to enforce this optimality while optimizing for the second objective. Thus, we shall not be too concerned with weakness 1 in this case. It is important to note, however, that if we have a fairness criterion, we can use SWP to find a Pareto-optimal solution, since it simply reduces to finding an optimal solution to the SWP.

With respect to weakness 2, we have a major issue. It is indeed the crux of the problem. While we do have a decision-maker that will validate the selected exchange plan and start the process of realizing the transplantations, it is not immediately obvious how to select a Pareto-optimal solution. Given two Pareto-optimal solutions x_1 and x_2, how do we determine the "best" one? We know that none of them dominates the other, so we are faced with a non-trivial problem. Usually, the intuitive idea is to try to balance the objectives in a way that satisfies all of them as close as possible to optimality. While we might sacrifice true optimality for one or both, closeness usually seems good enough. This is what the Nash Social Welfare Program (NSWP) will attempt to solve, as we will see later.

Finally, weakness 3 is another inherent problem of the SWP. Some solutions from the Pareto frontier might never be selected by a method based on weighting the objectives. This can be best illustrated by looking at a Figure 5.1. In this figure, the Pareto frontier is given

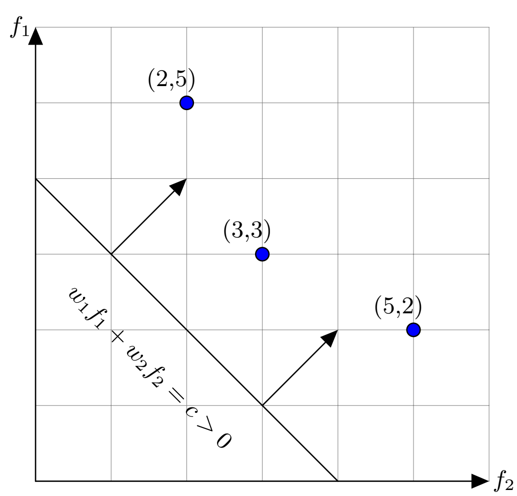


Fig. 5.1. Illustrating Weakness 3 of the SWP. The points in blue form the Pareto frontier[1].

[1]Inspired by figure in [9].

by the three points that are labelled with their Cartesian coordinates. Given any weights w_i, only the two extreme points, namely (2,5) and (5,2) will be obtained. The third point (3,3) cannot be achieved with this method. This can be observed by looking at the line that is drawn in the figure. Its slope is the same as the vector (w_1,w_2). The arrows indicate the direction to increase the objective value of the SWP. Even if the angle of the slope changes slightly, the SWP will still choose one of (2,5) or (5,2) as the best value from the Pareto front.

5.3. The Nash Social Welfare Program

In [9], the authors discuss the concept of *Nash Social Welfare Program* (NSWP). This is mathematically represented as:

$$\begin{aligned} &\max \prod_{i=1}^{k} (f_i(x) - d_i)^{w_i} \\ &\text{s.t. } x \in X \\ &\quad f_i(x) \geq d_i \qquad \forall i = 1, \ldots, k, \end{aligned}$$

where vector d is a reference point in $\mathbb{R}^k$.

Concretely, it is simpler to explain what the NSWP does if one restricts themselves to two objectives and unitary weights w_i. The NSWP attempts to maximize the total rectangular area defined by the two corners given by the reference point d and the solution picked from the Pareto front (see Figure 5.2). We can now apply the NSWP to KEPs. Note that the

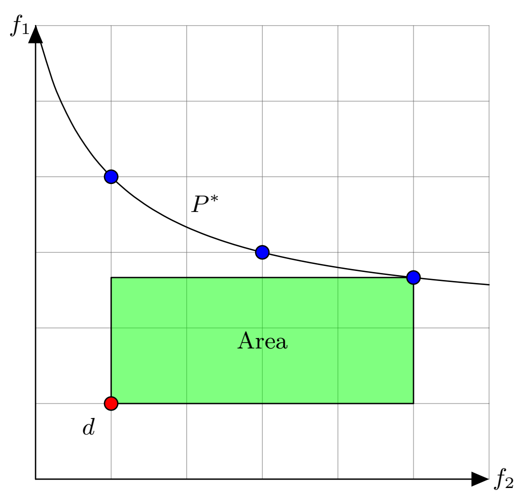


Fig. 5.2. Pareto front and the NSWP[2]

powers w_i can all be set to 1, to ensure that all objectives have the same order of precedence.

[2]Inspired by figure in [9].

The ultimate goal of applying NSWP to KEPs is to be able to select an exchange that will be both efficient in terms of the number of transplants, while also taking into account a fairness component. There might be multiple optimal solutions that can fit this description and this is why we again apply distributions over these exchanges. Therefore, the NSWP needs to be modified slightly.

The reference point d can be chosen to be the nadir point. We get the following program:

$$\begin{aligned} \max \quad & \mathbb{E}_p\left[(f_1(x) - d_1) \times (f_2(x) - d_2)\right] \\ \text{s.t. } & p \in \left\{ \delta : X \to [0{,}1] \,\middle|\, \sum_{x \in X} \delta(x) = 1 \right\}. \end{aligned}$$

Finally, this can be rewritten as:

$$\begin{aligned} \max \quad & \sum_{s \in \mathcal{S}} \delta_s (f_1(s) - d_1) \times (f_2(s) - d_2) \\ \text{s.t. } & \sum_{s \in \mathcal{S}} \delta_s = 1 \\ & \delta_s \geq 0 \quad \forall s \in S, \end{aligned} \tag{P}$$

where S is the set of solutions that maximize the objective $(f_1(x) - d_1) \times (f_2(x) - d_2)$.

Remark 5.3.1. *Note that because the expectation is a linear function, the set S is sufficient to find the optimal value of $\mathbb{E}_p\left[(f_1(s) - d_1) \times (f_2(s) - d_2)\right]$.*

This way, any feasible δ is optimal to P and hence, given S, one needs to decide the distribution to use. The challenge is thus to obtain S. It is not surprising that the cardinality of S can be very large. Therefore, we must enumerate elements of S and draw from the returned solutions according to some distribution. Alternatively, we can derive solutions directly from samples (without full enumeration), which approximate some distribution. For the latter approach, various ways to do this that have been proposed. In [32], the authors achieve near-uniform sampling of combinatorial spaces using $\oplus$ (XOR) constraints.

The NSWP exhibits certain interesting properties. It returns a Pareto-optimal solution and is both global-power-scale-free and local-benefit-scale-free [9]:

Proposition 5.3.2. *[9] The NSWP is global-power-scale-free: the NSWP with powers w_i is equivalent to the one with powers/weights αw_i for some $\alpha > 0$.*

Proposition 5.3.3. *[9] The NSWP is local-benefit-scale-free, i.e., if the objective is replaced with*

$$\max \prod_{i=1}^{k} (\alpha_i f_i(x) - \alpha_i d_i)^{w_i},$$

there is an equivalent problem to the original, up to a multiplication constant in the objective that is equal to

$$\prod_{i=1}^{k} \alpha_i^{w_i}.$$

Proposition 5.3.3 is interesting because it tells us that no matter what the scale of the objective functions is, we obtain an equivalent problem. Thus, we do not have to care about rescaling them or even comparing their scales to make sure they are comparable. The only relevant factor is the instantaneous change in one objective over the others. For example, we can suppose that the objective is a linear function in terms of the variable x, while the second objective is quadratic. This will not be equivalent to a problem where the second objective is linear too. Note that we can always play with the weights w_i to change this. The weights can be thought of as the "bargaining" power of each objective. We could have instances where each objective is the utility function of an agent and thus, each has a given bargaining power. Also, we could have two optimization criteria to be balanced like the utilitarian and fairness objectives.

The authors of [9] proceed to analyze the effectiveness of the NSWP when dealing with the issues of the SWP. For weakness 2, the NSWP does well since it balances the multiple objectives. Indeed, since we are maximizing the rectangular area defined by the two corners given the nadir point and the solution point, we will tend to pick solutions that do well with respect to each objective. For weakness 3, the NSWP provides a great workaround since it can return unsupported Pareto-optimal solutions. Therefore, NSWP is not disregarding such solutions of the Pareto frontier. It is important to note that we still might have multiple optimal solutions for the NSWP. Moreover, it does not tell us how to select from these equivalent solutions (equivalent in terms of their objective value). Since they are equivalent, we should be indifferent among them. Additionally, since we can allow for one or more of the objectives to take care of the fairness component of the problem, we can reassure the decision-maker by guaranteeing a certain level of fairness in the returned solution. But there is the lingering question of individual fairness in the context of the NSWP. Is it still an issue? How do we circumvent it? Can we integrate individual fairness inside this method?

To address the question of individual fairness, we can build on the method we described previously in this text. In other words, using the enumerated set of solutions for a particular instance, we balance the two objectives using the NSWP. Under this setting, the functions f_1,f_2 take a distribution δ over solutions to the KEP instance as an argument. We can think of f_1 as the utilitarian objective: the expected number of transplants under the distribution δ. The objective f_2 can simply be an individual fairness metric like L_1 or L_2-norm. In this way, the problem can be represented mathematically in the following manner:

$$\begin{aligned} \max \quad & \left(\sum_{s\in\mathcal{S}}\delta_s\sum_{c\in\mathcal{C}}w_c s_c - d_1\right)\times\left(-\sum_{v\in P}\left|\sum_{s\in\mathcal{S}|v\in s}\delta_s - \frac{1}{|P|}\sum_{v\in P}\sum_{s\in\mathcal{S}|v\in s}\delta_s\right| - d_2\right) \\ \text{s.t.} \quad & \sum_{s\in\mathcal{S}}\delta_s = 1 \\ & \delta_s \geq 0 \quad s\in\mathcal{S}, \end{aligned} \tag{P$_{IF}$}$$

where the first multiplicative term in the objective is f_1 and the second is f_2, representing the L_1-norm. $\mathcal{S}$ is the set of feasible exchange plans. The changes that must be performed when using L_2-norm are immediate. The optimal distribution under this setting will be expected to perform well under the utilitarian and fairness criteria. The key aspect to highlight here is that we were previously dealing with exchange plans that sought to maximize both the total utility and the balance of the distribution of the utilities among the patients involved. The latter approach rather seeks to find a distribution and this is exactly why we are able to apply our concept of individual fairness to it.

In order to tackle (P_{IF}), we are again forced to enumerate solutions. This is not a trivial task, especially when the objectives are complex functions that might not even be concave (maximization) or convex (minimization). Fortunately, when dealing with functions that can be described using Linear Programming, we are able to lift the tools from column generation. The various ways to such a more efficient enumeration will be described in the next section.

5.4. Algorithmic enhancement

Remark that the set of $\mathcal{S}$ in (P_{IF}) is potentially much larger than the set of optimal solutions to f_1 used in Chapter 3. We can use column generation to find the optimal distribution. First, we define $P' \subseteq P$ to be the set of pairs that are at least in one solution of the KEP (i.e. one solution of (PIEF)). By inspection of the Program (P_{IF}), we can write it in the following way:

$$
\begin{aligned}
\max\ & y_1 \times y_2 \\
\text{s.t.}\quad & \sum_{s\in\mathcal{S}} \delta_s = 1 \\
& y_1 = \sum_{s\in\mathcal{S}} \sum_{v\in\pi(s)} \delta_s - d_1 \\
& y_2 = -\sum_{v\in P'} z_v - d_2 \\
& |P'| z = \sum_{s\in\mathcal{S}} \sum_{v\in\pi(s)} \delta_s \\
& z_v \geq \sum_{s\in\mathcal{S}|v\in\pi(s)} \delta_s - z \quad \forall v \in P' \\
& z_v \geq z - \sum_{s\in\mathcal{S}|v\in\pi(s)} \delta_s \quad \forall v \in P' \\
& \delta_s \geq 0 \quad \forall s \in \mathcal{S},
\end{aligned}
\tag{P_{IF}}
$$

where $\pi(s)$ is the set of patients that receive a transplant in solution s. We can use the following results to modify the Problem (P_{IF}) and make the objective linear:

Lemma 5.4.1. *For any $y_1^*, y_2^*, y_1, y_2 \geq 0$, it holds that*

$$y_1^* \times y_2^* \geq y_1 \times y_2 \iff \sqrt{2y_1^* \times y_2^*} \geq \sqrt{2y_1 \times y_2}$$

Lemma 5.4.2. *The feasible region of the problem*

$$\begin{aligned} &\max r \\ s.t.\quad &0 \le r \le \sqrt{2y_1 \times y_2} \\ &y_1, y_2 \ge 0 \end{aligned}$$

is a rotated second-order cone $(y_1,y_2,r) \in \mathcal{Q}_r^3 = \{\alpha \in \mathbb{R}^3 \,|\, 2\alpha_1\alpha_2 \ge \alpha_3^2, \alpha_1 \ge 0, \alpha_2 \ge 0\}$.

Lemma 5.4.3. *In an optimal solution,*

$$t_v = \left|\sum_{s \in \mathcal{S}} \delta_s - z\right| \iff (t_v, z) \in \mathcal{Q}^2 = \left\{\alpha \in \mathbb{R}^2 \,\middle|\, \alpha_1 \ge |\alpha_2|\right\}.$$

The interested reader can find out more about $\mathcal{Q}_r^3$, $\mathcal{Q}^2$ and second-order conic programming in the article by Lobo et al. [33]. By having a linear objective, it can be observed that the optimum is an extreme point of the feasible region. From these lemmas, we obtain the following Master Problem, which is equivalent to (P_{IF}):

$$\begin{aligned}
&\min -r \\
\text{s.t.}\quad &(\pi_0) && \sum_{s \in \mathcal{S}} \delta_s = 1 \\
&(\pi_1) && y_1 = \sum_{s \in \mathcal{S}} \sum_{v \in \pi(s)} \delta_s - d_1 \\
&(\pi_2) && y_2 = -T - d_2 \\
&(\pi_3) && |P'|z = \sum_{s \in \mathcal{S}} \sum_{v \in \pi(s)} \delta_s \\
&(\beta_v) && z_v = \sum_{s \in \mathcal{S} | v \in \pi(s)} \delta_s - z \quad \forall v \in P' \\
&(\lambda_s) && \delta_s \ge 0 \quad \forall s \in \mathcal{S} \\
&(u \in \mathcal{Q}_r^3) && (y_1, y_2, r) \in \mathcal{Q}_r^3 \\
&(w_v \in \mathcal{Q}^2) && (t_v, z_v) \in \mathcal{Q}^2 \quad \forall v \in P' \\
&(\eta) && \sum_{v \in P'} t_v = T.
\end{aligned} \tag{MP}$$

Note that only the objective changed to $\sqrt{2y_1 \times y_2}$ which means that the optimal value of MP does not coincide with the optimal value of P. However, their optimal solutions coincide since $\sqrt{2y_1 y_2}$ is an increasing function in the domain where y_1 and y_2 are defined. Indeed, the reformulation of (MP) is the application of the second-order cone transformation for geometric mean constraints given in Ben-Tal and Nemirovski [34].

From this, we can deduce the dual of (MP):

$$
\begin{aligned}
& \max -\pi_0 + \pi_1 d_1 + \pi_2 d_2 \\
\text{s.t.} \quad & \begin{pmatrix} \pi_1 \\ \pi_2 \\ -1 \end{pmatrix} - u = 0 \\
& \begin{pmatrix} \eta \\ \beta_v \end{pmatrix} - w_v = 0 \quad \forall v \in P' \\
& \pi_2 - \eta = 0 \\
& \sum_{v \in P'} \beta_v + |P'|\pi_3 = 0 \\
(\dagger_s) \quad & \pi_0 - \sum_{v \in \pi(s)} (\pi_1 + \pi_3 + \beta_v) - \lambda_s = 0 \quad \forall s \in \mathcal{S} \\
& u \in \mathcal{Q}_r^3 \\
& w_v \in \mathcal{Q}^2 \quad \forall v \in P' \\
& \pi \in \mathbb{R}^3, \beta \in \mathbb{R}^{|P'|}, \eta \in \mathbb{R}, \lambda \geq 0.
\end{aligned}
\tag{DM}
$$

Since the only constraint on λ is $\lambda_s \geq 0$, we note that the constraints ($\dagger$) can instead be written as

$$(\dagger_s) \quad \pi_0 - \sum_{v \in \pi(s)} (\pi_1 + \pi_3 + \beta_v) \geq 0.$$

Finally, we get the subproblem:

$$
\begin{aligned}
& \min_s \pi_0^* - \sum_{v \in \pi(s)} (\pi_1^* + \pi_3^* + \beta_v^*) & \qquad \text{(SP)} \\
\text{s.t.} \quad & \sum_{c \in \mathcal{C} | v \in c} s_c \leq 1 & \forall v \in V \\
& s_c \in \{0,1\} & \forall c \in \mathcal{C}.
\end{aligned}
$$

The set $\mathcal{S}$ used in (MP) is defined in (SP) here by the constraints of the cycle formulation. We remark that many alternatives to define $\mathcal{S}$ can be used instead such as PIEF. We start by solving (MP) for a subset of $\mathcal{S}$. Then, we take the dual variables of the restricted (MP) and solve (SP). If the objective of the subproblem is negative, then we add the column corresponding to s. Otherwise, we already have an optimal subset of columns and we do not need more. The key idea here is that under an optimal solution $(\pi^*, \beta^*, \eta^*, \lambda^*)$ for the relaxed dual, if we can find a deterministic exchange $s \in \mathcal{S}$ that violates the constraint $(\dagger_s)$, then the restricted dual solution is not feasible in the (complete) dual program. We then need to add that row (corresp., column in (MP)). If all the constraints ($\dagger$) are satisfied, then the relaxed dual solution is in fact a solution to (DM). By strong duality, the optimal value will be the same as for (MP) and we will have our optimal subset of columns and rows. Note that the subproblem (SP) is an IP and (MP) and (DM) are SOCPs. This allows us to use conventional solvers to effectively solve them.

5.5. Experimental results

In the first set of experiments, we start by analyzing the relationship between the clinical definition of hard-to-match patients and its mathematical counterpart. Explicitly, we evaluate the correlation between hard-to-match patients and proxies to the number of solutions containing them. This allows us to question group fairness and motivate individual fairness.

The second set of experiments is concerned with evaluating the effectiveness of the NSWP in terms of its running time over large graphs, as well as its efficiency with respect to the two objectives that are simultaneously optimized. The efficiency is measured in terms of the POF. This allows us to experimentally validate the use of this method and also provide new research questions regarding its potential flaws, as well as solutions to them.

5.5.1. Hard-to-match patients

In our application of NSWP to KEPs, we can have various candidates for both the utilitarian and fairness criteria. To begin with, we considered maximizing the number of highly sensitized patients, defined by a PRA percentage above 80%. Since these patients should be harder to match when drawing a donor at random, we would expect these patients to be in fewer cycles and chains. Thus, we can think of the number of cycles and chains as a proxy for the number of exchange plans containing a patient. From the UNOS dataset used in Dickerson et al. [10], we computed the number of highly sensitized vertices that were contained in every possible cycle. Looking at all instances for which every solution containing the maximum number of hard-to-match patients could be enumerated, we obtain the histogram depicted in Figure 5.3. From Figure 5.3, we can directly conclude that many vertices are in few cycles or no cycles, while very few are in many cycles. However, the same shape of distribution can be found when looking at vertices with a PRA below 80%. It thus raises the question of whether this proxy is a good one. The results seem to indicate that it is not the case. In fact, we should instead focus our attention on the actual number of exchange plans containing each vertex because this is what captures the true advantage that a patient might have compared to another. The fact that we are using the structural properties of the graph instead of relying on distributional assumptions simply based on blood type and PRA will give us a better notion of compatibility between donors and patients under a random population. This way, we can know which patients will be hard-to-match to a donor and prioritize those under our exchange plan. Of course, we are completely disregarding the sequential notion of KEPs over time: indeed, it might be advantageous to opt for a different

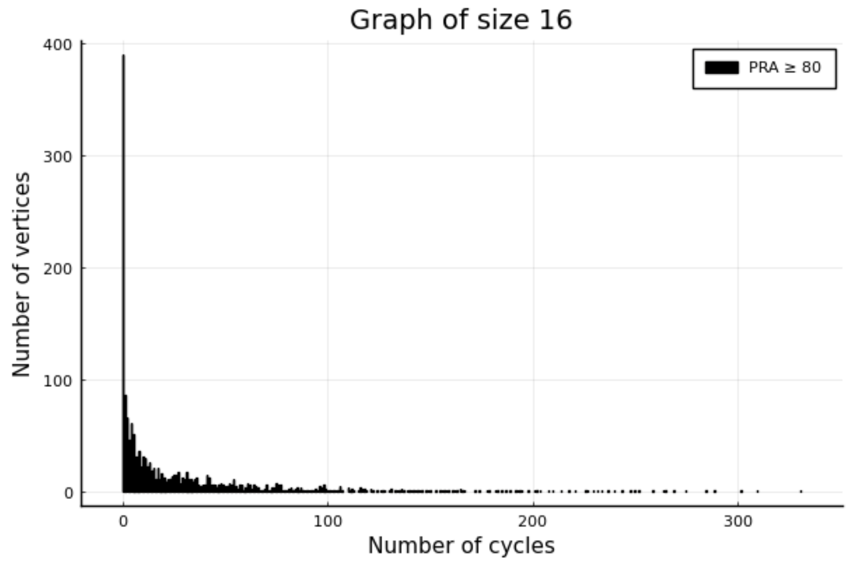


Fig. 5.3. How often does a vertex appear in cycles?

strategy than the one maximizing the number of hard-to-match patients given a particular graph instance at time t. This will be touched upon briefly in the following chapter.

Another interesting aspect to analyze is the correlation between the number of solutions containing a vertex and the number of cycles containing a vertex. Over the same instances that were solved, we obtain a correlation value of ~ 0.4. Therefore, we can conclude that this proxy is not a good one. Again, we should look at better proxies or find the true number of solutions containing each vertex (i.e. through enumeration). This also casts some doubt over the concept of group fairness introduced earlier in this text. The whole point of this approach is to give individuals that are marginalized in terms of their ability to obtain a compatible donor a better chance of getting one. Of course this comes at the expense of patients in a better position but altogether, the aim is to achieve some acceptable balance. From our results, it seems labelling patients as hard-to-match according to group notions such as PRA or number of cycles is not representative of the true capacity of the Program to find a compatible donor for these patients. Only by having a grasp on the number of solutions (i.e. exchange plans) containing each patient, are we able to truly determine how difficult it is for that patient to obtain a kidney. Whether we are relying on a fair procedure or a fair outcome, both should take this notion of individual fairness into account.

5.5.2. NSWP solution

We can also look at the efficiency of the NSWP method in terms of the utilitarian criterion. It would not be worthwhile considering such an optimization paradigm if we were not able to perform a large number of transplants. In order to do this, we can evaluate the efficiency of the NSWP as a percentage of the maximal number of transplants that can be performed when we completely disregard the fairness component of the optimization. The results can be observed in Table 5.1. We see that we do reasonably well in terms of the

$\lvert P \rvert$	16	32	64	128	256
POF	0.3399 ± 0.1422	0.3334 ± 0.0889	0.3640 ± 0.0597	0.3372 ± 0.0700	0.4173 ± 0.0173
POU	0.4252 ± 0.1537	0.5120 ± 0.0738	0.5620 ± 0.0565	0.5779 ± 0.0410	0.6304 ± 0.0149

Table 5.1. Efficiency of the NSWP compared to Utilitarian approach

number of transplants. However, we are still far from the optimal value since the POF varies between 0.3 and 0.4. The reason for this has nothing to do with the NSWP per se, but rather with the choice of metric and reference point (i.e. the nadir point). The particular nature of the IF metric signifies that when dealing with exchange plans of differing sizes (in the number of transplants), we must be careful in assigning a fairness value to such an exchange plan (see Section 6.2). There is also a considerable improvement in the f_2 objective. In the table, we list a measure that is related to POF, which we will call *price of utility* (or POU). It is exactly equivalent to the definition of POF but instead of evaluating the f_1 score of the fairness scheme, we evaluate the f_2 of the utility-maximizing scheme. Its values can range from 0 to 1 and a small value implies that one does not pay a high cost to fairness when maximizing utility (i.e. f_1). The purpose of these values is to show that maximizing utility results in noticeable degradation to the fairness score when compared with the NSWP. The improvement of the NSWP is even more significant as we increase the size of the graphs (smaller values are better).

Finally, we evaluate the efficiency of the column generation method by looking at the average time to find the optimal distribution over solutions of $\mathcal{S}$. These results are presented in Figure 5.4. We can see that the optimization terminates rather quickly on average. It is worth pointing out that scaling to very large instances will most likely encounter a bottleneck in the number of enumerated cycles and chains when using the cycle formulation. Thus, for these instances, (PIEF) [10] is the most obvious candidate formulation to use in (SP). This formulation has polynomial size, making the subproblem (SP) more compact and, one surmises, more easily solvable. By employing the column generation approach described in the previous section with the (PIEF) formulation, we were able to solve many large instances (graphs with 256 vertices) relatively quickly. Some instances, however, were harder to solve, and could not be terminated in the 1-hour time limit. This issue was encountered for a small subset of the KEP instances: even smaller sizes featured some of these difficult instances to solve. By extending the time limit to four hours for example, we were able to solve some of these instances. Some instances could take close to twenty-four hours to solve. The fact that there are special cases that are hard to solve is not surprising given that (SP) is NP-hard. From the sparsity of the L_1-norm, it is likely that most of the difficulty in solving

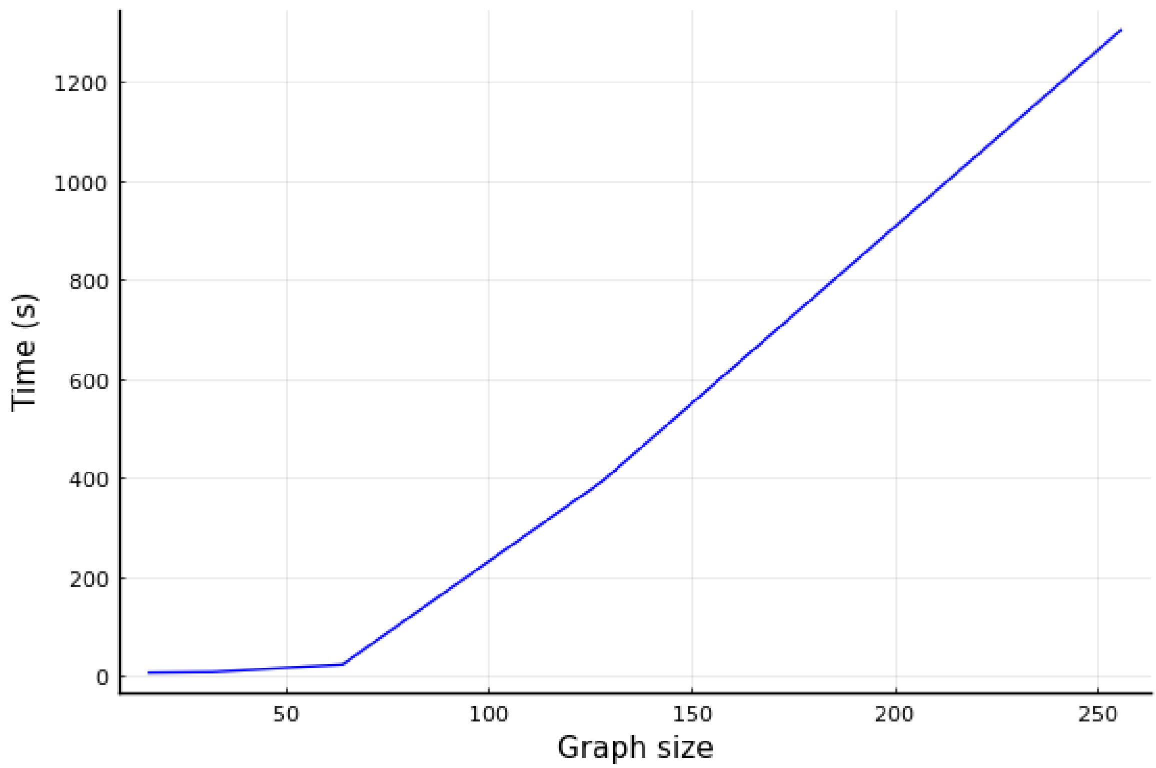


Fig. 5.4. Average time to solve (P_{IF}) using column generation as a function of graph sizes

the NSWP is related to the size of the mathematical formulation for (SP) and not to the enumeration itself as this difficulty was present even when simply solving (CF) or (PIEF). Further investigation needs to be done before concluding as such.

Chapter 6

Conclusion

Over the course of this thesis, we explored extensively the concept of fairness as it relates to KEPs. We first motivated the importance of KEPs in helping patients find a compatible donor. The use of KEPs in multiple countries, combined with ever-increasing pools of incompatible patient-donor pairs, implies that we must do everything we can to ensure a fair selection process when determining an exchange plan. We then proceed to describe KEPs mathematically and highlight the fact that many solutions often exist. Because of this, we motivated the need to enumerate multiple solutions. By having more flexibility in the choice of exchange plan, decision-makers are better equipped to combat biases against certain groups or individuals. Chapter 3 introduces the concept of individual fairness as an alternative to group fairness. By balancing the probabilities for each pair of being selected as part of an exchange plan, we ensure that no pair is unjustly treated and disregarded. We also suggest a column generation to ease the search of an optimal distribution for IF. Because not every solution is enumerated, we are able to solve many instances for which enumeration would be too lengthy or even infeasible. Chapter 4 discusses various fairness schemes and their benefits and costs. We provide mathematical descriptions of these schemes and recommend an alternative mechanism that can balance both fairness and utility. This leads to Chapter 5, which introduces the Nash Social Welfare Program as an improvement over the SWP. This framework allows us to optimize for two objectives at the same time and thus strike a balance between fairness and utility. The method is applied to KEPs and we extend the mathematical formulations to scale the method to larger graphs. We also provide an analysis of the effectiveness of the NSWP with respect to its two objectives.

6.1. Extended literature

As mentioned before in this thesis, the main goal of KEPs is to maximize the benefit of the patients. In an attempt to achieve this goal, the practiced trend assigns weights that allow to maximize the number of transplants while prioritizing certain groups of patients

(See for example Biró et al. [17] for a description of the disadvantaged groups receiving special weight in European KEPs). A group that has received particular attention is the set of highly-sensitized patients, who have a low probability of being compatible with a random kidney. In this context, Dickerson et al. [23] concentrate on the tradeoff of moving from maximizing the number of transplants (utilitarian objective function) towards maximizing the number of highly-sensitized patients receiving a kidney. However, McElfresh et al. [35] 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. [36] 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.

The works mentioned above consider static KEPs, as it was the case in this text. Anticipation of future kidney exchanges is taken into account in Dickerson and Sandholm [37] 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. Gao [38] also argues that instead of focusing solely on highly-sensitized patients, a dynamic KEP should take into account time-critical pairs, i.e. pairs whose patients are in critical condition.

In Klimentova et al. [39], Biro et al. [30], 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 [40], Ashlagi and Roth [41], Carvalho and Lodi [20] 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. [42, 43]), 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.

6.2. Future work

The algorithmic enhancement that was developed in Section 5.4 made use of the L_1-norm for the fairness objective. However, it can be observed that a limitation arises when using this fairness criterion. Farnadi et al. [6] use the L_1-norm to balance the patient probabilities (Chapters 2 and 5). Nonetheless, we do so by first restricting the set of solutions to be the ones maximizing the number of transplants. The goal of the NSWP, on the other hand, is to determine how and if it is possible to relax that optimality constraint in order to achieve a fairer distribution in terms of the L_1 score. But what happens under the empty exchange

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

plan, i.e. when no transplant is performed? Obviously, this solution is far from optimal in the number of transplants performed. However, its L_1 measure is 0, which is the best value obtainable. We can observe that depending on how we choose the reference point d, this can cause undesirable solutions. Let us consider the nadir point to be the reference point. Using the same function f_1 and f_2 as in Section 5.4, we get the following nadir point:

$$
\begin{aligned}
d_1 &= \inf_{\delta | \delta \in \arg\max f_2(\delta)} f_1(\delta) \\
&= 0 \\
d_2 &= \inf_{s | s \in \arg\max f_1(s)} f_2(s)^2 \\
&= -\sum_{v \in P' \cap \pi(s^*)} (1 - \frac{1}{|P'|}|\pi(s^*)|) - \sum_{v \in P' \setminus \pi(s^*)} \frac{1}{|P'|}|\pi(s^*)| \\
&= -|P' \cap \pi(s^*)| - \frac{(|P' \setminus \pi(s^*)| - |P' \cap \pi(s^*)|)|\pi(s^*)|}{|P'|}.
\end{aligned}
$$

Thus, a deterministic exchange plan that maximizes the number of transplants could be considered worse than a nearly empty exchange plan. For the sake of clarity, if we suppose that s is an exchange plan with $|\pi(s)| = \frac{|P'|}{2}$ and $K = 2$, we could take the nearly empty exchange plan s' to be any two vertices forming a pairwise exchange under s. It can be observed that s' has a L_1 score converging to 3:

$$\lim_{|P'| \to \infty} 2 \times \left(1 - \frac{2}{|P'|}\right) + (|P'| - 2) \times \frac{2}{|P'|} = \lim_{|P'| \to \infty} 3 - \frac{8}{|P'|} = 3.$$

This is much less than the score for s, which is equal to $\frac{|P'|}{2}$. Therefore, by using the nadir point as defined above, the exchange plan s' scores better than s: the NSWP value for s is $\frac{|P'|}{2} \times (d_2 - d_2) = 0$, while it is strictly greater than 0 for s'. This is a problem since under no circumstance, does it not make sense to consider s' to be better than an exchange plan containing it (s in the discussion). The issue can be remedied by simply changing the reference d_2 to be a higher value such as the theoretically highest value of L_1 possible for any graph of the same size (i.e. when half of the vertices are in an exchange plan). Nevertheless, it might not be immediately obvious if this is a good idea or not. The fact that we would have to try the optimization process and then adjust the reference point (or indeed the powers w_i) subsequently seems undesirable.

The main objective of a good IF criterion is to penalize distributions that do not involve as many pairs as possible and that favour some pairs disproportionately over others. Thus, from the above discussion, we propose a new definition of an individual fairness criterion. It is worth to point out that this was not an issue when considering only optimal solutions in terms of the number of transplants (in Section 3.1). However, because we now seek to

[2] Abuse of notation for f_2.

include solutions that might not be optimal under f_1, we must adjust the definition of the IF metric. To resolve the issue described in the previous paragraph, we can simply compute the metric on distributions over exchange plans that cannot be extended. In other words, we do not consider an exchange plan s if there exists another exchange plan s' such that $\pi(s) \subsetneq \pi(s')$. This way, the only suboptimal exchange plans (with respect to f_1) considered are those that feature at least one new pair. This idea is somewhat related to the greedy cover enumeration of Section 2.2. In turn, we must adapt (SP) in the column generation approach. The new formulation is now a bilevel optimization program.

$$\min_{s,s'} \pi_0^* - \sum_{v \in \pi(s)} (\pi_1^* + \pi_3^* + \beta_v^*) \tag{SP'}$$

$$\text{s.t.} \quad f_1(s') \leq f_1(s)$$

$$s' \in \arg\max \left\{ f_1(t) : \sum_{c \in \mathcal{C} | v \in c} s_c \leq \sum_{c \in \mathcal{C} | v \in c} t_c \leq 1 \quad \forall v \in V \wedge t_c \in \{0,1\} \quad \forall c \in \mathcal{C} \right\}$$

$$\sum_{c \in \mathcal{C} | v \in c} s_c \leq 1 \qquad \forall v \in V$$

$$s_c \in \{0,1\} \qquad \forall c \in \mathcal{C}$$

In other words, the subproblem still minimizes the same objective, but now there must be a certificate that no other exchange plan s' is such that $\pi(s) \subseteq \pi(s')$, i.e. no other exchange plan strictly contains the set of pairs included in s.

One interesting future research direction is the evaluation of patient utility function. Specifically, we are interested in learning their utility functions using machine learning. This can be useful for multiple reasons. In this thesis, it was assumed for the most part that all patients are equally satisfied with being part of an exchange plan. This was modeled using an indicator function: 0 utility is derived when the patient is not selected and a utility score of 1 otherwise. In reality, it is expected that different patients have different preferences. Many factors can influence the level of satisfaction of a patient in a KEP. For instance, the quality of the graft and its level of compatibility with the patient certainly influence the latter's satisfaction [44]. There is also the time spent waiting in the KEP pool and how it affects the deterioration of the patient's health. An immediate difficulty with learning the patient utilities is to control for the bias that each individual possesses. In other words, the utilities should be invariant to the choice of patient when the parameters (i.e. characteristics of the patient, graft, time spent in the KEP pool, etc.) are unchanged. This is an important property to strive for as we must be able to understand the tradeoff between the utilities of different patients, since they would be hard to compare otherwise.

Another area of investigation is related to uniform sampling of a set of optimal solution in a mathematical programming problem. This work proposes simple enumeration methodologies through IP and CP to find a distribution over the set of solutions that minimizes some fairness loss. An interesting research direction consists in uniformly (or near-uniformly) sampling the set of optimal solutions of a mathematical program. This is what Gomes et al. [32] introduce. However, the method still involves an exponential number of constraints to be able to sample near-uniformly the set of optimal solutions. Being able to sample it efficiently could potentially lead to useful fairness applications whenever there are limited resources that must be distributed amongst a population. Diversity of solutions can also leave decision-makers in better positions if other constraints arise.

In the same vein as the previous idea, building neural networks that can learn distributions over discrete solutions without having to fully enumerate the set of optimal solutions would facilitate the adoption of IF methods to KEPs. This would scale better to large graphs and could translate to a wider adoption in large KEP pools, ultimately helping more patients to find a compatible donor. Also, applications such as large barter exchange systems could prove to be interesting from an economic point of view.

Finally, the work presented in this thesis only focuses on static KEP pools: time is not taken into account when optimizing an objective. It is of interest to explore the behaviour of KEP in a dynamic setting. In fact, a dynamic KEP enables more flexibility in terms of fairness since it can be optimized over multiple time steps. As the horizon is infinite, some approaches of reinforcement learning can be useful in understanding and modeling the problem. In fact, it might be necessary to focus on a more general description than simply KEPs since the general problem of optimization over an infinite horizon with constraints is not currently well understood.

References

[1] CDC. Chronic kidney disease basics, Feb 2020. URL `https://www.cdc.gov/kidneydisease/basics.html`.

[2] K Park, JH Lee, KH Huh, SI Kim, and YS Kim. Exchange living-donor kidney transplantation: diminution of donor organ shortage. *Transplantation proceedings*, 36(10): 2949–295, 2004.

[3] 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.

[4] 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.

[5] 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.

[6] Golnoosh Farnadi, William St-Arnaud, Behrouz Babaki, and Margarida Carvalho. Individual fairness in kidney exchange programs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11496–11505, May 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17369`.

[7] Dimitris Bertsimas, Vivek Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59:17–31, 02 2011. doi: 10.1287/opre.1100.0865.

[8] G. Owen. *Game Theory*, volume Fourth edition of *0*. Emerald Group Publishing Limited, 2013. ISBN 9781781905081. URL `https://books.google.fr/books?id=yeVbAAAAQBAJ`.

[9] Hadi Charkhgarda, Kimia Keshanianb, Rasul Esmaeilbeigic, and Parisa Charkhgardc. The magic of nash social welfare in optimization: Do not sum, just multiply! *Optimization Online*, 2020.

[10] John P. Dickerson, David F. Manlove, Benjamin Plaut, Tuomas Sandholm, and James Trimble. Position-indexed formulations for kidney exchange. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, page 25–42, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450339360. doi:

10.1145/2940716.2940759. URL https://doi.org/10.1145/2940716.2940759.

[11] John F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1907266.

[12] John Rawls. *A theory of justice*. Oxford University Press, 1973.

[13] David J. Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 295–304, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-653-0. doi: 10.1145/1250910.1250954. URL http://doi.acm.org/10.1145/1250910.1250954.

[14] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review*, 97(3):828–851, June 2007. doi: 10.1257/aer.97.3.828. URL https://www.aeaweb.org/articles?id=10.1257/aer.97.3.828.

[15] Miguel Constantino, Xenia Klimentova, Ana Viana, and Abdur Rais. New insights on integer-programming models for the kidney exchange problem. *European Journal of Operational Research*, 231(1):57–68, 2013. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.05.025. URL https://www.sciencedirect.com/science/article/pii/S0377221713004244.

[16] Ross Anderson, Itai Ashlagi, David Gamarnik, and Alvin E. Roth. Finding long chains in kidney exchange using the traveling salesman problem. *Proceedings of the National Academy of Sciences*, 112(3):663–668, 2015. ISSN 0027-8424. doi: 10.1073/pnas.1421853112. URL https://www.pnas.org/content/112/3/663.

[17] Péter Biró, Joris van de Klundert, David Manlove, William Pettersson, Tommy Andersson, Lisa Burnapp, Pavel Chromy, Pablo Delgado, Piotr Dworczak, Bernadette Haase, Aline Hemke, Rachel Johnson, Xenia Klimentova, Dirk Kuypers, Alessandro Nanni Costa, Bart Smeulders, Frits Spieksma, María O. Valentín, and Ana Viana. Modelling and optimisation in european kidney exchange programmes. *European Journal of Operational Research*, 2019. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2019.09.006. URL http://www.sciencedirect.com/science/article/pii/S0377221719307441.

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

[19] Susan Saidman, Alvin Roth, Tayfun Sönmez, Utku Unver, and Francis Delmonico. Increasing the opportunity of live kidney donation by matching for two- and three-way exchanges. *Transplantation*, 81:773–82, 04 2006. doi: 10.1097/01.tp.0000195775.77081.25.

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

[21] David García-Soriano and Francesco Bonchi. Fair-by-design algorithms: matching problems and beyond. *CoRR*, abs/1802.02562, 2018. URL `http://arxiv.org/abs/1802.02562`.

[22] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*. Elsevier, 2006. ISBN 978-0-444-52726-4. URL `http://www.sciencedirect.com/science/bookseries/15746526/2`.

[23] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1013–1020. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[24] M. B. Zafar, Isabel Valera, M. Gomez-Rodriguez, and K. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2017.

[25] Laurence Wolsey. *Column (and Row) Generation Algorithms*, chapter 11, pages 213–233. John Wiley & Sons, Ltd, 2020. ISBN 9781119606475. doi: https://doi.org/10.1002/9781119606475.ch11. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119606475.ch11`.

[26] Ken Kipnis. Triage and ethics, Jan 2002. URL `https://journalofethics.ama-assn.org/article/triage-and-ethics/2002-01`.

[27] Alexander F. C. Hulsbergen, Marleen M. Eijkholt, Naci Balak, Jannick Brennum, Ciarán Bolger, Anna-Margarete Bohrer, Zeev Feldman, Daniel Holsgrove, Neil Kitchen, Tiit I. Mathiesen, Wouter A. Moojen, Nicolás Samprón, Martin Sames, Ulrika Sandvik, Magnus Tisell, and Marike L. D. Broekman. Ethical triage during the COVID-19 pandemic: a toolkit for neurosurgical resource allocation. *Acta Neurochirurgica*, 162(7): 1485–1490, July 2020. ISSN 0001-6268, 0942-0940. doi: 10.1007/s00701-020-04375-w. URL `https://link.springer.com/10.1007/s00701-020-04375-w`.

[28] M.T.L.B. An introduction to probability theory and its applications, vol. i. by william feller. [pp. xii 419. new york: John wiley and sons inc.; london: Chapman and hall ltd. 1950. price 48s.]. *Journal of the Staple Inn Actuarial Students' Society*, 10(4):316–318, 1951. doi: 10.1017/S0020269X00004679.

[29] Daniel Bernoulli. Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36, 1954. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/1909829`.

[30] Peter Biro, Márton Gyetvai, Xenia Klimentova, Joao Pedro Pedroso, William Pettersson, and Ana Viana. Compensation scheme with shapley value for multi-country kidney exchange programmes. In *Proceedings of the 34th International ECMS Conference on*

Modelling and SimulationECMS 2020, pages 129–136, 06 2020. doi: 10.7148/2020-0129.

[31] Gary E Bolton, Jordi Brandts, and Axel Ockenfels. Fair procedures: Evidence from games involving lotteries*. *The Economic Journal*, 115(506):1054–1076, 2005. doi: https://doi.org/10.1111/j.1468-0297.2005.01032.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0297.2005.01032.x.

[32] Carla P Gomes, Ashish Sabharwal, and Bart Selman. Near-uniform sampling of combinatorial spaces using xor constraints. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007. URL https://proceedings.neurips.cc/paper/2006/file/4110a1994471c595f7583ef1b74ba4cb-Paper.pdf.

[33] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1): 193–228, 1998. ISSN 0024-3795. doi: https://doi.org/10.1016/S0024-3795(98)10032-0. URL https://www.sciencedirect.com/science/article/pii/S0024379598100320. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.

[34] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2001. ISBN 9780898718829. URL https://books.google.ca/books?id=CENjbXz2SDQC.

[35] Duncan C. McElfresh, Hoda Bidkhori, and John P. Dickerson. Scalable robust kidney exchange. *CoRR*, abs/1811.03532, 2018. URL http://arxiv.org/abs/1811.03532.

[36] 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.

[37] John Paul Dickerson and Tuomas Sandholm. 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, 2014.

[38] Irena Gao. Fair matching in dynamic kidney exchange. *CoRR*, abs/1912.10563, 2019. URL http://arxiv.org/abs/1912.10563.

[39] Xenia Klimentova, Ana Viana, Jo ao Pedro Pedroso, and Nicolau Santos. Fairness models for multi-agent kidney exchange programmes. *Omega*, 2020. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2020.102333. URL https://www.sciencedirect.com/science/article/pii/S0305048320306873.

[40] Tayfun Sönmez and M. Utku Ünver. Market design for kidney exchange. In *The Handbook of Market Design*. Oxford University Press, 2013.

[41] Itai Ashlagi and Alvin E. Roth. Free riding and participation in large scale, multi-hospital kidney exchange. *Theoretical Economics*, 9(3):817–863, 2014. ISSN 1555-7561.

doi: 10.3982/TE1357.

[42] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151 – 188, 2005. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2005.04.004. URL http://www.sciencedirect.com/science/article/pii/S0022053105001055.

[43] Jian Li, Yicheng Liu, Lingxiao Huang, and Pingzhong Tang. 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, page 445–452, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450327381.

[44] Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. Deep learning for patient-specific kidney graft survival analysis. *CoRR*, abs/1705.10245, 2017. URL http://arxiv.org/abs/1705.10245.

Appendix A

Optimal L_1 distribution for Figure 3.2

First, we denote the set of vertices of the yellow and green paths by P_1 and P_2, respectively. We begin with the followin lemma:

Lemma A.0.1. *The mean vertex probability $\bar{\delta}$ is a function of the graph G and is independent of the choice of distribution δ when S only contains solutions that maximize the number of transplants.*

Proof. We write the mean vertex probability:

$$\begin{aligned}
\bar{\delta} &= \frac{1}{|P'|} \sum_{v \in P'} \sum_{s \in S | v \in \pi(s)} \delta_s \\
&= \frac{1}{|P'|} \sum_{s \in S} \sum_{v \in P' | v \in \pi(s)} \delta_s \\
&= \frac{1}{|P'|} \sum_{s \in S} |\pi(s)| \delta_s \\
&= \frac{OPT}{|P'|} \sum_{s \in S} \delta_s \\
&= \frac{OPT}{|P'|}.
\end{aligned}$$

□

Using this result, we suppose that we have the following distribution δ for Figure 3.2:

- The cycle C_L is selected with a probability of $\bar{\delta}$. We will abuse notation and denote the probability associated with selecting C_L as δ_{C_L};
- The yellow and green paths are each selected with a probability of $\frac{1}{2} - \bar{\delta}$; we will abuse notation and denote this as δ_{P_i}.

We now prove the main result.

Proposition A.0.2. *The distribution δ introduced above for is an optimum for the L_1-norm of Figure 3.2.*

Proof. We begin with our initial distribution δ with mean vertex probabillity $\bar{\delta}$. We deviate from it with an arbitrary small vector ϵ (same dimension as δ). We write out the L_1-norm explicitly:

$$\begin{aligned}
L_1(\delta+\epsilon) &= \sum_{v\in P_1\Delta P_2}\left[\bar{\delta}-\left(\delta_{P_i}-\sum_{s|v\in\pi(s)}\epsilon_s\right)\right]+\sum_{v\in P_1\cap P_2\setminus C_L}\left[2\delta_{P_i}+\sum_{s|v\in\pi(s)}\epsilon_s-\bar{\delta}\right]\\
&+\sum_{v\in C_L\cap P_1\cap P_2}[1+\sum_{s|v\in\pi(s)}\epsilon_s-\bar{\delta}]+\sum_{v\in C_L\setminus(P_1\cup P_2)}\left|\bar{\delta}-\left(\delta_{C_L}+\epsilon_{C_L}\right)\right|\\
&= L_1(\delta)-\sum_{v\in P_1\Delta P_2}\sum_{s|v\in\pi(s)}\epsilon_s+\sum_{v\in P_1\cap P_2}\sum_{s|v\in\pi(s)}\epsilon_s+\sum_{v\in C_L\setminus(P_1\cup P_2)}|\epsilon_{C_L}|\\
&= L_1(\delta)-\sum_{s}\sum_{v\in P_1\Delta P_2|v\in\pi(s)}\epsilon_s+\sum_{s}\sum_{v\in P_1\cap P_2|v\in\pi(s)}\epsilon_s+\sum_{v\in C_L\setminus(P_1\cup P_2)}|\epsilon_{C_L}|\\
&= L_1(\delta)-\sum_{s\neq C_L}|P_1\Delta P_2|\epsilon_s+\sum_{s\neq C_L}|P_1\cap P_2|\epsilon_s+\sum_{v\in C_L\setminus(P_1\cup P_2)}|\epsilon_{C_L}|+\epsilon_{C_L}\\
&= L_1(\delta)+\sum_{v\in C_L\setminus(P_1\cup P_2)}|\epsilon_{C_L}|+\epsilon_{C_L}\\
&\geq L_1(\delta).
\end{aligned}$$

We get the last equality since $|P_1\Delta P_2| = |P_1\cap P_2| = \frac{L}{2}$. Because ϵ is arbitrarily small, we conclude that δ is locally optimal. Because L_1 is convex, this implies that δ is a global optimum. $\square$