Université de Montréal

Weak Core Solution for the Non-Transferable Utility Kidney Exchange Game

par

Raphaël Collette

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

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

Août 2023

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Weak Core Solution for the Non-Transferable Utility Kidney Exchange Game

présenté par

Raphaël Collette

a été évalué par un jury composé des personnes suivantes :

Emma Frejinger
(présidente-rapporteure)

Margarida Carvalho
(directrice de recherche)

Jean-Yves Potvin
(membre du jury)

Résumé

Plusieurs pays possèdent des programmes de don croisé de rein (PDCR). Le but de ces programmes est d'aider les patients ayant un donneur incompatible à obtenir une greffe, en échangeant les donneurs incompatibles entre les patients. Pour pouvoir obtenir des bassins de paires incompatibles de plus grande taille, il est possible d'élargir les PDCR pour y inclure plusieurs pays ou hôpitaux. Par contre, on doit s'attendre à ce que ces derniers agissent de façon stratégique pour maximiser le nombre de leurs patients obtenant une greffe. Avec ce cadre, on peut définir le problème de don croisé de rein à plusieurs agents.

Dans ce mémoire, nous modélisons ce problème comme un jeu coopératif à utilité non-transférable et nous présentons le *noyau faible* comme solution à ce jeu. Nous étudions empiriquement notre solution sur des exemples basés sur des données réelles et montrons qu'elle est atteignable en pratique. Nous comparons aussi le noyau faible à une autre solution présente dans la littérature: les couplages résistants aux rejets.

Mots Clés : Don croisé de rein, Noyau faible, Théorie des jeux coopératifs, Couplage maximum, Programmation en nombres entiers.

Abstract

In various countries, kidney paired donation programs (KPDs) are implemented. These programs aim to help patients with an incompatible donor to obtain a transplant by swapping the donors between the patients. In order to increase the size of the pool of incompatible patient-donor pairs and potentially enhance patient benefits, KPDs can be extended to include multiple countries or hospitals. However, unlike existing nationwide KPDs, strategic behaviour from these entities (agents) is to be expected. This gives rise to the multi-agent kidney exchange problem.

In this work, we model for the first time this problem as a non-transferable utility game. We also propose and argue in favour of the use of the weak core as a solution concept for the game. Using integer programming tools, we empirically study our solution concept on instances from the literature, which are derived from real-world data, and show that it is attainable in practice. We also compare the weak core to another recently presented solution concept from the literature, the rejection-proof matching.

Keywords: Kidney exchange, Weak core, Cooperative game theory, Maximum matching, Integer programming.

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

PDCR	Programme de don croisé de rein
KPD	Kidney paired donation program
NDD	Non-directed donor
IP	Integer program
2^N	Power set of N
N-KEP	N-player kidney exchange problem

Remerciements

Un énorme merci à Margarida qui m'a accompagné tout au long de ma maîtrise. Elle m'a embarqué dans un projet passionnant et a été une excellente mentore. J'attendais toujours avec impatience nos rencontres, car je savais qu'elle allait pouvoir me guider dans la bonne direction lorsqu'un problème me posait de la difficulté, ou m'encourager à continuer quand j'étais sur la bonne voie. Margarida a été une directrice sympathique, accommodante et à l'écoute. J'ai énormément appris avec elle et je n'aurais pas pu travailler avec une meilleure personne pendant ces deux années!

Je remercie également mes amis et ma famille. J'ai passé de beaux moments avec vous pendant ma maîtrise et je me compte chanceux d'être aussi bien entouré.

Introduction

Problem context

Since the publication in 1944 of the groundbreaking book *Theory of Games and Economic Behaviour* [1] by Von Neumann and Morgenstern, the field of game theory has grown to become an important discipline combining the fields of computer science, applied mathematics and economy. Game theory has found many applications in everyday life. Examples of this include urban traffic control, school choices by students and auctions for online advertisement spots. Another topic where the tools of game theory have proven themselves useful is the topic of multi-agent kidney exchanges.

Mutli-agent kidney exchanges arise from the fact that it is possible for a patient suffering from kidney failure to receive a kidney transplant from a live donor possessing two healthy kidneys. Usually, a living donor will be a close relative of the patient. In order for the transplant to occur, however, the patient-donor pair must meet multiple clinical criteria. Kidney exchanges can help overcome the problems caused by patient-donor incompatibility. The idea is to swap the donors of two incompatible pairs. For this to occur, the donor of the first pair needs to be compatible with the patient of the second pair, and vice versa. Nowadays, many countries possess national kidney exchange programs. We have a multi-agent kidney exchange when multiple organizations – which can be hospitals, transplant centres or countries – combine their pools of incompatible pairs to try to obtain more kidney transplants. In such a context, we can expect the agents to act strategically in order to obtain more transplants for the patients under their care. Hence, even if we try to maximize the global welfare, the agent's behaviour might prevent it.

It is here that game theory can step in to help us not only to model the interactions of the agents in such context, but also to devise suitable mechanisms to distribute the transplants among the agents. Many properties can be requested from the mechanisms. Fairness, individual rationality, strategyproofness and stability over deviations from coalitions are all examples that have been studied in the literature. While it is impossible to devise the perfect mechanism, satisfying all desirable properties, we can study and compare them to provide decision makers with the most complete information possible.

Contributions

Our first contribution is to model the multi-agent kidney exchange problem as a cooperative non-transferable utility game. As our second contribution, we propose the *weak core* as a suitable solution concept for our game. The weak core is a less strict version of the *core*, a widely used solution for cooperative games. As our last contribution, we present computational experiments showing that the weak core is attainable in practice. Moreover, we theoretically and empirically compare the weak core to the rejection-proof matching, another recent solution concept from the literature. Our computational results show that it is possible to obtain kidney exchange solutions meeting both the weak core and rejection-proof criteria. For all our computational experiments, we use two cutting plane algorithms devised by us.

Organization of the thesis

In Chapter 1, we revise the game theoretical literature on kidney exchanges. We start by providing a formalization of the kidney exchange problem and we give an overview of its different integer programming formulations. Afterwards, we survey approaches on multi-agent kidney exchanges from the cooperative and non-cooperative game theory perspective. Chapter 2 contains the research paper and finally, Chapter 3 concludes the thesis and discusses further work.

Chapter 1

Literature review

In this chapter, we start by introducing kidney exchange programs in Section 1.1. We then provide an integer programming formulation for the kidney exchange problem in Section 1.2 and explain multi-agents kidney exchanges in Section 1.3. Finally, we overview important approaches for the multi-agents kidney exchange problem from the view point of cooperative game theory in Section 1.4 and non-cooperative game theory in Section 1.5.

1.1. Kidney exchange programs

Patients suffering from kidney failure can have their life quality greatly improved by receiving a kidney transplant [2]. While the transplant can come from a deceased donor, it can also come from a live donor possessing two healthy kidneys. Living donors must, however, meet multiple clinical criteria for them to be compatible with the patient. Hence, even if a patient finds a willing donor, the transplant is not guaranteed to take place. In an attempt to circumvent this problem, Rapaport [3] introduced the concept of a live donor kidney exchange. The idea involves utilizing two incompatible patient-donor pairs. If the donor in the first pair is compatible with the patient in the second pair, and vice versa, a kidney exchange can occur. In this exchange, each pair's donor gives their kidney to the other pair's patient, creating a cycle of length 2. This foundational idea underlies the seminal work in the definition of kidney exchange systems, also known as kidney paired donation programs (KPDs) [4, 5, 6]. In fact, pairwise exchanges can even be generalized to more than two pairs by forming exchange cycles of length L. Moreover, non-directed donors (NDD) can be included. A non-directed donor is a donor who has no patient assigned to them. They can therefore be used in chains of donations where the NDD gives to the patient of an incompatible pair, whose donor gives to another pair and so on.

Nowadays, many countries such as Canada [7] or the Netherlands [8] possess national kidney exchange programs, where incompatible pairs and NDDs can enroll. These systems

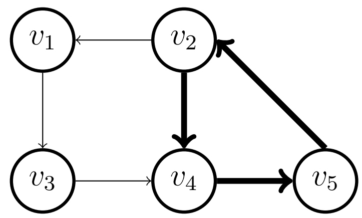


Figure 1.1. A compatibility graph for a kidney exchange instance. The arcs in bold represent a possible exchange plan.

aim to form exchange plans among their participants with the goal of maximizing the benefits for the registered patients.

1.2. Integer programming formulation of the kidney exchange program

Since the goal of most kidney exchange programs is to maximize the registered patients' welfare, it is natural to turn to the tools of mathematical optimization to analyze these programs. Following the work of Roth et al. [9], we can model the pool of incompatible pairs of the kidney exchange program as a directed compatibility graph $G = (V, A)$. The set of vertices V represents incompatible patient-donor pairs and an arc $(i,j) \in A$ if an only if the donor of vertex i is compatible with the patient of vertex j. If the program includes NDDs, they can be included in the set of vertices, and arcs can be defined analogously, except that no arc can end in a vertex representing an NDD. Figure 1.1 provides an example of a compatibility graph with an exchange plan. The goal of KPDs is to maximize a pre-defined objective through the selection of an exchange plan, i.e., a set of disjoint cycles and chains (bearing in mind that each donor can contribute only one kidney). The length of the cycles is usually restricted to an upper-bound, as the transplants within it need to take place simultaneously. This synchronization attempts to avoid the cycle to break because of a donor withdraw. Therefore, due to logistic reasons for the simultaneous performance of transplantations, the length of the cycles is restricted. See for example Biró et al. [10] for a description of exchange constraints of KPDs in Europe as well as the objectives being optimized.

As a special case, we note that if a KPD is restricted to exchanges of length 2 with no NDD, the compatibility graph becomes a non-directed graph $G = (V,E)$, where an edge $(i,j) \in E$ is placed between vertices i and j if and only if the donor of i is compatible with the patient of j and vice-versa (that is, $(i,j) \in A$ and $(j,i) \in A$). An exchange plan is thus a matching of G, i.e., a subset of E consisting of non-adjacent edges. In this case, an exchange plan yielding a maximum number of transplants corresponds to a maximum matching of the graph. It is possible to assign weights to the edges or arcs of the graph to attribute a value to the exchanges. In the case where we just want to obtain the maximum number of transplants, the weights are simply set to 1.

Given a directed compatibility graph $G = (V,A)$, the problem of finding an exchange plan of maximum weight can be formulated as an integer program (IP). The first two IP formulations were proposed by Abraham et al. [11] and Roth et al. [12]. Both model exchanges with size at most L and do not include NDDs in their formulation. Following Abraham et al. [11], one is the edge formulation:

$$\max_{a} \quad \sum_{a \in A} w_a a \tag{1.2.1a}$$

$$\text{s.t.} \quad \sum_{a_{out}=(v_i,v_j)} a_{out} - \sum_{a_{in}=(v_j,v_i)} a_{in} = 0 \qquad \forall v_i \in V \tag{1.2.1b}$$

$$\sum_{a_{out}=(v_i,v_j)} a_{out} \leq 1 \qquad \forall v_i \in V \tag{1.2.1c}$$

$$\sum_{k=1}^{L} a_{p_k} \leq L - 1 \qquad \forall p \in \mathcal{P} \tag{1.2.1d}$$

$$a \in \{0,1\} \qquad \forall a \in A. \tag{1.2.1e}$$

Here the a's are binary variables indicating whether the arc they represent was selected in the exchange, while the w_a's represent the weights of those arcs. Constraints (1.2.1b) account for the fact that a donor in v_i is selected to participate in an exchange if and only if their associated patient receives a kidney, while Constraints (1.2.1c) account for the fact that a donor in vertex v_i can only donate to a single patient. Constraints (1.2.1d) limit the possible exchanges to cycle of length at most L ($\mathcal{P}$ represents the set of paths of length L).

The second formulation is the cycle formulation. We let $C(L)$ be the set of cycles of length at most L, and for $c \in C(L)$, we let w_c be the sum of the weights of the arcs in c. This results in the following IP:

$$\max_{c} \quad \sum_{c \in C(L)} w_c c \tag{1.2.2a}$$

$$\text{s.t.} \quad \sum_{c \in C(L) | v_i \in c} c \leq 1 \qquad \forall v_i \in V \tag{1.2.2b}$$

$$c \in \{0,1\} \qquad \forall c \in C(L). \tag{1.2.2c}$$

In both formulations, the number of variables is exponential in the instance size. As an improvement, Constantino et al. [13] provide two new formulations that are compact i.e. their number of variables and constraints is bounded by a polynomial in the number of pairs in the instance. In Mincu et al. [14], the authors list properties that might be implemented in a kidney exchange program and provide their formulation as constraints of an integer program. Dickerson et al. [15] introduce three IP formulations. Two of them are compact and one of the compact has a tight linear program relaxation. Riascos-Álvarez et al. [16] and Omer et al.

[17] also provide IP formulations and branch-and-price methodologies to solve the problems effectively in practice. Dickerson et al. [15], Riascos-Álvarez et al. [16] and Omer et al. [17] all account for NDDs in their models.

1.3. Multi-agents kidney exchange

In order to increase the size of the incompatible pairs pools and, potentially, improve the patients' welfare, multiple countries, hospitals or transplant centres can combine their kidney exchange pools. In the U.S., there are three cross-hospital programs, the NKR, the APD, and the UNOS, which involve different participating hospitals. International kidney exchanges already occur in Europe, e.g., an exchange between Czech Republic and Austria [18] and an exchange involving Italy, Spain and Portugal [19]. Indeed, cross-border programs exist such as the South Alliance for Transplants, which includes France, Italy, Spain, Portugal and Switzerland; and the Scandiatransplant, which includes Denmark, Finland, Iceland, Norway, Sweden and Estonia. More recently, the APD in the U.S. and the CNT (the organization managing transplantation activities in Italy) have signed an agreement to allow kidney exchanges between their respective countries [20].

The existence of the above programs make it interesting to study multi-agent kidney exchange programs (the agents being countries or hospital). In this context, we can no longer only aim to just maximize the overall patients' welfare (or more concretely, the primary KPD objective in single-agent systems, the number of transplants), as strategic behaviour can be expected from the agents. Countries and hospitals participating in such programs will indeed try to maximize the welfare of their own patients. We can use the tools of game theory to help us model the interactions between the agents as well as try to devise mechanisms which maximize social welfare, while taking the strategic behaviour of the agents into account. The game theoretical approach can mainly be separated into two categories: the cooperative one and the non-cooperative one.

1.4. Cooperative game theory

In the cooperative game theory context, the general assumption is that a set of agents participating in the kidney exchange program will merge their pairs in a large single pool. An independent agent will then choose a set of exchanges to be performed involving a subset of the pairs. The goal of such a program is thus to ensure that the decisions of the independent agent follow a certain set of mathematical properties, ensuring that the process accounts for aspects such as fairness or individual rationality, while also maximizing the utility of the agents (here the utility of the agents can mean the number of transplants they receive or, more generally, the sum of the weight of the transplants they receive). In simple words, in the cooperative case, it is assumed that agents will collaborate as long as they deem the program beneficial accordingly with their individual utility.

Klimentova et al. [21] propose integer programming models permitting exchanges of length at most L and including non-directed donors. These IPs model the maximization of the number of transplants in a pool created by joining the incompatible pairs and NDDs of multiple agents, while also ensuring that the participation of each agent is individually rational. A kidney exchange program is individually rational when all the participating agents are guaranteed to be better off by participating in the program. Moreover, they implement two different second level objectives which share the goal of fairly distributing the surplus of transplants generated by the cooperation of the agents. The first is based on each agent's potential to augment their number of transplants, while the second is based on the benefit that each agent contributes to the common pool of incompatible pairs. In both cases, the fairness is achieved through multiple rounds of exchanges via a credit system, since it might not be possible for a single round to achieve a fair outcome. Biró et al. [22] expand the work of Klimentova et al. [21] by comparing the fairness based on the benefit of each agent (the benefit value) to another compensation scheme using the Shapley value. The Shapley value, first presented in [23], is a common solution concept in cooperative game theory. It distributes the total value (here, the number of transplants) obtained by the cooperation of the agents while respecting four desirable properties. These properties are efficiency, symmetry, additivity and the null player property. Through computational experiments, the authors found that both the Shapley and the Benefit value yielded similar numbers of transplants, but the Shapley value gave results that tended to have smaller temporary unfairness.

Biró et al. [24] model the international kidney exchange problem by allowing exchanges of size two and by attributing a numerical value to an exchange. They do this through the novel concept of a *generalized matching game*. In this game, there is a weighted undirected graph with every player owning a subset of its vertices. Each coalition S of players has a value corresponding to the maximum weighted matching it can form on its associated subgraph, and the goal is to find a weighted matching of the graph that is in the *core* (this is a matching where all players collaborate and no player has incentive to deviate from the *grand* coalition). Since the core of an instance of this game might be empty, the authors propose a credit system to compensate over multiple rounds the countries which matched fewer vertices than their target allocation. Benedek et al. [25] also use generalized matching games while allowing two-way exchanges, but without adding weights to the exchange. They investigate six different solution concepts from the cooperative game theory literature as target allocations, while also using a credit system when the target allocation is unattainable. They select their solution by choosing a matching that lexicographically minimizes the difference between the players' utilities and their target allocation. The authors perform a computational study with up to 15 players to show that lexicographically minimizing the matchings yields more balanced solutions, in the sense that the utility of the players after many matching runs is closer to their target utility.

1.5. Non-cooperative game theory

In the non-cooperative game theory context, we suppose that the cooperation of the agents is not enforced. When they participate in a kidney exchange program, they might decide to match a subset of their incompatible pairs beforehand, to not register a subset of their pairs or to withdraw from the program altogether. The goal is usually to predict how the agents will behave in a given context, and perhaps to look for a set of rules (mechanism) leading to more desirable actions from them.

A desirable property that a kidney exchange program should possess is individual rationality, as it garantees that all the agents want to participate in the program. Another desirable property is strategyproofness. A mechanism is strategyproof when it is weakly dominant for all the agents to reveal their private information. In the context of kidney exchanges, strategyproofness would imply that the agents gain no benefit by hiding some of their pairs to the program.

Ashlagi and Roth [26] observe that the matching algorithms in the U.S. do not make it individually rational for hospitals to register all their pairs in the U.S. UNOS program, resulting in a loss of transplants. They show that when allowing exchanges of size greater than 2, no single-round mechanism can be strategyproof and individually rational, while still producing a maximum number of transplants. They however propose a mechanism that would incentivize hospitals to register their easy to match patients in the program instead of only registering their hard-to-match patients. Ashlagi et al. [27] allow exchanges of size 2 and construct a randomized mechanism that is strategyproof and provides a 2-approximation. This means that in the worst case, the cardinality of the matching outputted by the mechanism will at least be half of the cardinality of a maximum matching on the compatibility graph. Hajaj et al. [28] propose a mechanism working over multiple rounds that attains the maximum number of transplants, while being strategyproof. To make their mechanism strategyproof, they use a credit system where the agents receive credits when they reveal more imcompatible pairs than they are expected to have.

One of the most central concepts in game theory is the concept of Nash equilibrium [29]. It provides a solution concept for a competitive game. In such a game, we have a set of agents $N = \{1, \ldots, n\}$ and each agent i possesses a set of strategies S_i. We let $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ with $s_k \in S_k$ denote the strategies chosen by all the players except i. A player's payoff $u_i(s_{-i}, s_i)$ is a function of their strategy as well as the other players' one. Way say that a set of strategies $s_1^\star, \ldots, s_n^\star$ is a *Nash equilibrium* if for all players i, $u_i(s_{-i}^\star, s_i^\star) \geq u_i(s_{-i}^\star, s_i) \quad \forall s_i \in S_i$. This means that a set of strategies is a Nash equilibrium if, given the other players' strategy profile, no player has an incentive to unilaterally change their strategy. In his famous paper, John Nash proved that all finite games possess a Nash equilibrium, provided that we allow the use of mixed strategies. That is, we allow a player to put probabilistic weights on a subset of their possible strategies instead of only choosing one of them. When we have a Nash equilibrium where all the players choose to play a single strategy, we call it a *pure strategy Nash equilibrium*.

Carvalho et al. [30] analyze the international kidney exchange problem through the lens of the Nash equilibrium. They consider pairwise exchanges and only the two players case. They devise a game where each player owns a set of incompatible patient-donor pairs and first select an internal matching on their pairs, as their strategy. An independent agent then selects a maximum matching on all the remaining unmatched pairs. The authors show that this game always possesses a pure strategy Nash equilibrium yielding a matching of maximum cardinality. In Carvalho and Lodi [31], the authors extend their previous result to the general case of an n-player game. Blom et al. [32] model the kidney exchange game assuming that the players can do what they call rejection strategies, as opposed to the withholding strategies presented by Carvalho et al. [30]. The game consists of a mechanism proposing a set of feasible exchanges, the proposed solution, on the set of all the players' incompatible pairs. These exchanges are allowed to be of any given size. The players then individually select which of the proposed exchanges involving at least one of their pairs they accept. They can also choose to make new exchanges involving their pairs only. These decisions are the players' rejection strategies. Afterwards, the only exchanges from the initial matching that are carried are the exchanges for which every participating player agrees. The new exchanges involving only a single player are also carried. The mechanism is deemed *rejection-proof* if it is a weakly dominant strategy for each player to accept the proposed matching. Blom et al. [32] propose two rejection-proof mechanisms. Through computational experiments, they show that rejection strategies provide better outcomes for the players than withholding strategies and the use of a rejection-proof mechanism comes with relatively small losses when compared to a mechanism choosing an optimal solution.

Chapter 2

Weak Core Solution for the Non-Transferable Utility Kidney Exchange Game

by

Raphaël Collette[1], and Margarida Carvalho[1]

([1]) CIRRELT and Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, H3T 1J4, Canada

This article will be submitted in a journal after minor modifications.

Raphaël Collette was involved in all stages of the research, including the proposal of the approach to tackle the problem, the conduct of a literature review, the formulation of the model, the demonstration of the theoretical results, the implementation of the code, the analysis of the results and the writing of the article.

Abstract. In various countries, kidney paired-donation programs (KPDs) are implemented. These programs aim to help patients with an incompatible donor to obtain a transplant by swapping the donors between the patients. In order to increase the size of the pool of incompatible patient-donor pairs and potentially enhance patient benefits, KPDs can be extended to include multiple countries or hospitals. However, unlike existing nationwide KPDs, strategic behaviour from these entities (agents) is to be expected. This gives rise to the multi-agent kidney exchange problem.

In this work, we model for the first time this problem as a non-transferable utility game. We also propose and argue in favour of the use of the weak core as a solution concept for the game. Using integer programming tools, we empirically test our solution concept on instances from the literature, which are derived from real-world data, and show that it is attainable in practice. We also compare the weak core to another recently presented solution concept from the literature, the rejection-proof matching.
Keywords: Kidney exchange, Weak core, Cooperative game theory, Maximum matching, Integer programming.

2.1. Introduction

Patients suffering from kidney failure can have their life quality greatly improved by receiving a kidney transplant [2]. The transplant can come from a deceased donor or a living donor possessing two healthy kidneys. For the transplant to occur, the patient-donor pair needs to satisfy clinical criteria, such as blood type compatibility. Hence, even if a patient finds a living donor willing to help them, they might still be unable to obtain a transplant. Rapaport [3] introduced the concept of a live donor kidney exchange to help mitigate the problem of finding a compatible donor. The idea consists of finding two incompatible pairs and swapping their donors, provided that the donor of the first pair is compatible with the patient of the second pair and vice versa. This forms a 2-way exchange, and the idea can be naturally generalized to L-way exchanges, i.e., a sequence of L swaps. Moreover, it is possible to include donors with no patients associated to them. These non-directed donors (NDDs) can be part of chains of donations: an NDD donates to a patient within an incompatible pair, subsequently, the donor from this pair can donate to another pair, and so forth.

The previous ideas gave rise to the definition of kidney paired donation programs (KPDs) [4, 5, 6]. The goal of KPDs is to maximize the welfare of the patients registered in the program. A typical metric of welfare is the number of transplants, or a weighted sum of the number of transplants if certain transplants are deemed to have more value than others. Nowadays, national KPDs are implemented in many countries such as Canada [7] or the Netherlands [8]. To increase the size of the pool of incompatible pairs, hospitals, transplant centres or countries can combine their incompatible pairs into a joint pool. Examples of this include three cross-hospitals programs in the U.S. (the NKR, the ADP and the UNOS) and multiple occurrences of international kidney exchanges. For example, exchanges happened between Czech Republic and Austria [18]; and between Italy, Spain and Portugal [19]. Recently, two organizations, one from Italy and the other from the U.S., have signed an agreement to allow kidney exchanges between these two countries [20].

In light of this, it becomes of interest to study multi-agent KPDs. In a single agent KPD, the goal is to maximize the patients' welfare, usually by maximizing the number of transplants. In multi-agent KPDs, this can no longer be the only goal as strategic behaviour is to be expected from the agents, compromising potential gains of a joint pool of patients and donors. For instance, in order to increase the amount of patients from an agent receiving a kidney, the agent can withhold some of their easy to match pairs from the program or withdraw from it altogether if the agent judges that it no longer benefits them. Here, the tools of cooperative and non-cooperative game theory can help us not only in modeling, but also in devising solution concepts for multi-agent KPDs. Next, we provide an overview of the different approaches that were used in the literature to tackle the multi-agent KPD problem, and then, we summarize our contributions.

2.1.1. Literature review

Cooperative game theory. In the cooperative game theory context, the general assumption is that a set of agents participating in the KPD will merge their pairs in a big single pool. An independent agent will then choose a set of exchanges to be performed, often called *exchange plan*. The mechanism (procedure) through which the independent agent selects an exchange plan must satisfy certain desirable mathematical properties, usually relying on the contribution of each agent to the global pool. The output of a mechanism satisfying certain properties is referred to as solution concept. Hence, different solution concepts can be used to attain various properties.

Klimentova et al. [21] propose integer programming formulations as the mechanism for the selection of an exchange plan. These formulations use a hierarchy of objectives where at the second level, the adequate distribution of the surplus of transplants generated by the joint pool of the agents is optimized. Biró et al. [22] use the Shapley value [23] as a solution concept. Biró et al. [24] introduce the *generalized matching game* as a suitable framework to model multi-agent kidney exchanges. They attribute a numerical value to the possible exchanges and use the core as a solution concept. Benedek et al. [25] also use the generalized matching game framework, but do not attribute values to the exchanges. They investigate six solution concepts from the cooperative game theory literature. While doing so, they try to reach an exchange plan that lexicographically minimizes the difference between the agent's payoff and a target allocation set by the solution concept.

It is important to point out that all the solution concepts discussed above are often not attainable, in the sense that we might not be able to reach their requirements for a given kidney exchange instance. As a result, all the previous papers run their mechanisms over multiple rounds and implement a credit system to compensate the agents who were less favoured in a given round.

Non-cooperative game theory. In the non-cooperative game theory context, we suppose that the cooperation of the agents is not enforced. When they participate in a KPD, they might decide to match a subset of their incompatible pairs beforehand, to not register a subset of their pairs or to withdraw from the program altogether. The goal is usually to anticipate how the agents will behave in a given context, and perhaps to look for a set of rules (mechanism) leading to more desirable actions from them. Desirable properties most notably include individual rationality and strategyproofness. Individual rationality ensures that all the agents are better off when participating in a KPD, and strategyproofness makes sure that they have no incentive to hide some of their pairs from the program.

Ashlagi and Roth [26] show that no single-round mechanism can be both strategyproof and individually rational while still selecting an exchange plan with the maximum number of transplants. Ashlagi et al. [27] design a strategyproof randomized mechanism that provides an exchange plan whose cardinality is at least half of the cardinality of a maximum exchange plan. Hajaj et al. [28] devise a strategyproof mechanism that is able to attain the maximum number of transplants. They use a credit system so that participating in the mechanism over multiple rounds becomes strategyproof for the agents. Carvalho et al. [30] analyze the multi-agent kidney exchange problem as a competitive 2-player game, restricted to 2-way exchanges. They show that this game always possesses a Nash equilibrium, which achieves the maximum number of transplants. Carvalho and Lodi [31] expand their work to the case of an n-player game. Finally, Blom et al. [32] also study the context of a competitive game. However, they assume that the players use rejection strategies, as opposed to the withholding strategies described in [30, 31]. Their work is described in greater details in Section 2.4.

2.1.2. Our contribution and paper overview

In this paper, we model the multi-agent kidney exchange problem (N-KEP) as a cooperative game. As in [27, 25, 30, 31], we consider 2-way exchanges with no NDDs. This enables us to work with matchings on undirected graphs, simplifying the problem from both mathematical and computational perspectives, while still examining a case of practical interest, as certain countries only consider 2-way exchanges or give them priority [10]. In our game the goal of the mechanism (or independent agent) is to output an exchange plan maximizing the number of transplants. As our first contribution, contrary to other papers, we frame the game as a non-transferable utility game. This encompasses the fact that kidney transplants are a non-transferable resource. Moreover, it is generally agreed that monetary payments should not be made to obtain a kidney. This makes it harder to distribute the utility generated by a group of agents cooperating. For example, no side payments can be made to compensate an agent whose set of patients receives less transplants. Following this, as our second contribution, we introduce the *weak core* as a solution concept for our game. As indicated in its name, the weak core is a weaker version of the core, which is a widely used solution concept for cooperative games. The relaxation of the core requirements provide us with a solution concept that is more easily attainable in practice, therefore reducing the need of introducing a credit system. All the above concepts of the weak core and of a non-transferable utility game are described in Section 2.2 with the necessary background. In Section 2.3, we provide an integer programming formulation of the problem of finding a matching (exchange plan) in the weak core. We also devise a cutting plane method to solve the problem. In Section 2.4, we give an overview of the rejection-proof kidney exchange mechanisms proposed by Blom et al. [32], and investigate some theoretical resemblances it possesses with the weak core. Finally, in Section 2.5, as our last contribution, we present the results of the computational experiments from applying the weak core to instances from the literature based on real-world data. Our results show that the weak core is an attainable solution concept in practice, as we were not able to find an instance with an empty weak core among all the tested ones. Moreover, our experiments also show that in practice, we are always able to find a solution for the N-KEP that is not only in the weak core, but is also rejection-proof.

2.2. Problem formalization

In Section 2.2.1, we introduce the essentials of cooperative games that we need for the rest of the paper. Then, in Section 2.2.2, we present the multi-agent kidney exchange problem from the perspective of cooperative game theory. Section 2.2.3 concludes with the definition of *weak core*, which is our proposed solution concept to the N-KEP.

2.2.1. Characteristic function games

A *characteristic function game* is a pair (N, v), where $N = \{1, \dots, n\}$ is a set of players and $v : 2^N \to \mathbb{R}$ is a characteristic function that assigns a payoff to each coalition $S \subseteq N$. The outcome of such a game is a *payoff vector* $x = (x_1, \dots, x_n)$, allocating a payoff $x_k \geq 0$ to each player $k \in N$ and respecting $\sum_{k \in N} x_k \leq v(N)$. Such a vector x is also called an imputation. The value obtained by a coalition S under a payoff vector x, is defined to be $x(S) = \sum_{k \in S} x_k$. In many games, such as the N-KEP, the characteristic function is monotone, i.e., $v(S) \leq v(S')$ for all $S \subseteq S' \subseteq N$, and hence, the goal is to find a payoff vector allocating the value $v(N)$. In other words, the maximum payoff lies in the *grand coalition* N, which implies that the focus is on determining a payoff vector that incentivizes all players to cooperate. A widely used solution concept to answer this problem is the *core*, first introduced by Gillies [33].

Definition 1. The core of a characteristic function game (N, v) is the set of payoff vectors x such that $x(S) \geq v(S)\ \forall S \subseteq N$.

A payoff vector in the core ensures that no coalition wishes to deviate from the grand coalition, since it cannot obtain a better payoff by doing so. However, the core of a given game may be empty, as we shall see later. It is therefore interesting to investigate whether a given game has a non-empty core and, if so, how to compute it.

2.2.2. The multi-agent kidney exchange problem

The N-KEP is defined as follows: First, there is a set of countries[1], each possessing a pool of incompatible donor-patient pairs. Each country can carry out internal 2-way exchanges between its pairs, and international 2-way exchanges can occur as well. A 2-way exchange can occur (i.e., it is feasible) when the donor of one pair is compatible with the patient of another pair, and vice versa. The goal is to find an exchange plan that only contains feasible 2-way exchanges which are disjoint (i.e. a pair does not participate in two separate exchanges). In doing so, we also aim to maximize the number of transplants, while also ensuring that the agents will not deviate from the proposed plan. Each agent wants to maximize their own number of transplants and they will deviate if they can improve their situation. We suppose that when an agent or a coalition deviates, it tries to find a set of exchanges only among the pairs belonging to the deviating agents.

We can model the above problem as a characteristic function game. To this end, we start by mathematically abstracting the game into a graph using the setup by Roth et al. [9]. Concretely, the compatibility between all the pairs is represented as an undirected graph $G = (V,E)$, called the *compatibility graph*, where each vertex corresponds to an incompatible donor-patient pair and $(i,j) \in E$ if and only if the donor of i is compatible with the patient of j and vice-versa. The participating countries form the set $N = \{1, \ldots, n\}$ of players. We define $G^k = (V^k, E^k) \subseteq G$ to be the subgraph induced by the vertices of player k, $G^S = (V^S, E^S)$ to be the subgraph induced by the vertices of coalition S, $E^I \subseteq E$ to be the set of possible international exchanges and we let $E^I_k \subseteq E^I$ be the set of international edges adjacent with a vertex of V^k. Finally, we also set $V(e)$ to be the set of vertices at the ends of edge e. A feasible set of exchanges is given by a *matching* $M \subseteq E$. A matching of a graph is a subset of the edges such that no two edges are incident to the same vertex, hence, in our setup, this corresponds to a set of disjoint and feasible 2-way exchanges. We define $\mathcal{C}(S)$ to be the set of all possible matchings of maximum cardinality (i.e. maximum matchings) attainable by coalition S. We restrict ourselves to maximum matchings since for every non-maximum

[1]These agents could also be hospitals or transplant centres.

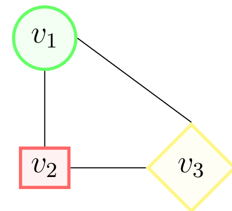


Figure 2.1. A kidney exchange instance with empty core

matching of a coalition on their graph G^S, there is a maximum matching such that each agent in S is as well off. We say that such a maximum matching dominates the previous matching. Finally, we conclude the description of the game by defining the characteristic function $v : 2^N \to \mathbb{R}$ as $v(S) = 2 \cdot |M^S|$ for some $M^S \in \mathcal{C}(S)$. This function is clearly monotonic.

In this context, after players have registered their pairs in a centralized platform, the latter determines a maximum matching M of G, which corresponds to a payoff vector (imputation) where each player's payoff is equal to the number of its pairs in the matching. In other words, for player k, the payoff $x_k(M)$ received from matching M is $x_k(M) = 2 \cdot |M \cap E^k| + |M \cap E^I_k|$. A matching M is in the core if $2 \cdot |M \cap E^S| = \sum_{k \in S} x_k(M) \geq v(S)\ \forall S \subseteq N$, i.e., no coalition of players has incentive to leave the grand coalition. Therefore, the determination of a matching in the core would be a matching rule ensuring the stability of the grand coalition. Nevertheless, core solutions do not always exist for the characteristic function formulation of the N-KEP. A simple example of an instance with empty core is illustrated by Figure 2.1. In this instance player 1 controls the circle, player 2 the square and player 3 the diamond. For each coalition S of size 2, we have $v(S) = 2$, but $v(N) = 2$ and thus it is impossible to find a matching in the core.

2.2.3. The weak core in a non-transferable utility game

Despite previously studied in other works [24, 34], the core as a solution concept for the N-KEP has a notable flaw: Besides being potentially empty, it fails to encompass the fact that kidney transplants are a resource that can not be redistributed. When we calculate the value of the characteristic function for a given coalition, we assume that the payoff generated can be freely redistributed among the members of the coalition. In other words, we suppose that we have a *transferable utility game* when we in fact have a *non-transferable utility game*. In some cases, the assumption of a transferable utility game may lead us to the incorrect conclusion that a coalition can profitably deviate from a given payoff vector, whereas some of its members would obtain a lower payoff by deviating. To see this, consider the example given by Figure 2.2. Player 1 owns the circles, Player 2 the squares and Player 3 the diamonds. A maximum matching is $\{(v_1,v_2),(v_4,v_5),(v_7,v_8)\}$ with payoff vector equal to $x = (3,0,3)$. Here, if we use the core as a solution concept we find that the coalition $\{1,2\}$ wants to deviate since

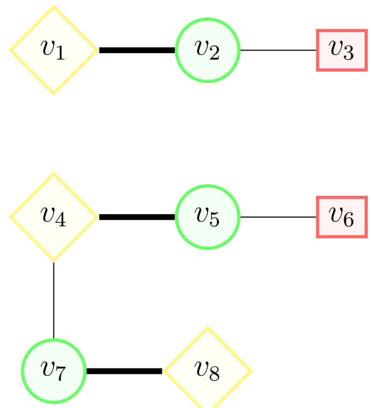


Figure 2.2. A false deviation

$v(\{1,2\}) = 4$ and $x(\{1,2\}) = 3$. This is clearly not rational for Player 1, since their payoff decreases by 1. These details can be encompassed by a different type of game.

Definition 2. A non-transferable utility game is a pair (N,u), where N is a set of players and $u \subseteq 2^N \times \mathbb{R}^{|N|}$ is a total characteristic relation that links every coalition to its feasible payoff vectors. [2]

We can now define the core for non-transferable utility games by using the notion of Pareto improvement (or objection). For a set of players S, an imputation x' is a Pareto improvement over another imputation x if every player in S is equally well off and at least one player is strictly better off.

Definition 3. The core of a non-transferable utility game (N,u) is the set of payoff vectors x such that for all coalitions S there is no $x' \in \mathbb{R}^{|N|}$ such that $(S, x') \in u$ and x' is a Pareto improvement for S.

We saw with Figure 2.1 that the core of the transferable utility kidney exchange game can be empty. Unfortunately the instance presented in Figure 2.1 also has an empty core when we consider it as a non-transferable utility game. In the hopes of augmenting the number of instances where the solution concept exists, we weaken the core conditions, obtaining a new solution concept.

Definition 4. The weak core of a non-transferable utility game (N,u) is the set of payoff vectors x such that there is no $(S, x') \in u$ where x' is a strong Pareto improvement over x with respect to S. A strong Pareto improvement $(S,x') \in u$ over x is a Pareto improvement in which every player in S with payoff x' is strictly better off than in x.[3]

[2]We note that we have applied a simplified version of the definition of non-transferable utility game that fits our context. We refer the interested reader to [35] for more details on non-transferable utility games.

[3]We remark here that in some work such as [35], what we define as the weak core for a non-transferable utility game is referred to as the core. Here we choose to use the term weak core to differentiate our solution concept from others in the kidney exchange literature

We can now apply this definition to the N-KEP. First, let $u^k_{M^S,S}$ be the payoff obtained by player k when deviating with coalition S under the matching M^S of G^S, i.e., $u^k_{M^S,S} = 2 \cdot |M^S \cap E^k| + |M^S \cap E^I_k|$. Our valuation relation is given by $(S,x) \in u \iff \exists M^S \in \mathcal{C}(S)$ such that $x_k = 0$ if $k \notin S$ and $x_k = u^k_{M^S,S}$ if $k \in S$. A matching M is in the weak core if no coalition can find a matching among its vertices providing a strong Pareto improvement.

Since every matching in the core is also in the weak core, the latter provides a larger solution space to work with. However, this comes with the downside that weak core solutions are not as resistant to coalition deviations as core solutions. We argue that in the context of the N-KEP, this drawback is not very important. Indeed, for a coalition to deviate from a payoff vector in the weak core proposed by a centralized platform, quite some work would be required: the coalition members would need to establish communication channels (to obtain information about the pools of the participant players), and to compute and agree on a new matching for their subgraph. Furthermore, new arrangements would have to be made to perform the transplants, since the coalition would no longer benefit from the organizational structure provided by a central platform, from which it deviates from the decision. Finally, as this is, after all, a game about helping as many patients as possible, altruism by the players towards the overall welfare is to be expected in practice. Taking all these factors into account, it seems implausible that a player would agree to the extra work of forming a coalition in order to deviate to a situation leading to the same number of transplants. Of course, a game-theoretical solution concept is only as powerful as the empirical evidence demonstrating that the game outcome coincides with it, so the stability of this solution should be tested experimentally in future work.

2.3. Computing the weak core

In this section, we provide an integer programming (IP) formulation for the problem of finding a matching in the weak core. We then devise an algorithm (a cutting plane method) that exploits graph structures found in our problem to quickly solve the IP.

2.3.1. The two players case

Before attacking the general N-player problem, we first tackle the much simpler two-player case. For this, we use a non-cooperative setup for the 2-KEP developed by Carvalho et al. [30], which we will call the withholding game. In this setting, players 1 and 2 first act individually by selecting internal matchings, M^1 and M^2, on their graphs G^1 and G^2. We call M^1 and M^2 the players' strategies. Afterwards, an independent agent selects a maximum matching M^I among all the remaining edges. The matchings M^1, M^2 and M^I form a *Nash equilibrium* if neither player can choose a different matching, while the other keeps the same, and obtain a better payoff. The Nash equilibrium is a widely used solution concept for non-cooperative games [29].

In this non-cooperative setup, we notice that a player k can always select a maximum matching M^k as their strategy, and therefore any Nash equilibrium must lead to a payoff of at least $2|M^k|$. Hence, any Nash equilibrium is in the weak core since the only possible deviating coalitions are $\{1\}$ and $\{2\}$, where players can achieve the payoff of the maximum matching of their individual graphs.

Carvalho et al. [30] proved that we can always find a Nash equilibrium in polynomial time in the withholding game. We can thus also always find a matching in the weak core in polynomial time by computing a Nash equilibrium with the algorithm devised by Carvalho et al. We thus conclude:

Lemma 1. The 2-KEP has a non-empty weak core. Moreover, a matching in the weak core can be computed in polynomial time.

2.3.2. The general case

The problem of finding a matching in the weak core for the N-KEP can be expressed as the following mathematical programming problem:

$$\max_{y} \quad \sum_{e \in E} y_e \tag{2.3.1a}$$

$$\text{s.t.} \quad \left(\bigvee_{k \in S} u^k_{M^S,S} \leq \sum_{e \in E} w^k_e y_e \right) \quad \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S) \tag{2.3.1b}$$

$$\sum_{e \in E} a^j_e y_e \leq 1 \quad \forall j \in V \tag{2.3.1c}$$

$$y_e \in \{0{,}1\} \quad \forall e \in E. \tag{2.3.1d}$$

In Problem (2.3.1), the y_e are binary variables with value 1 when $e \in E$ is part of the selected matching and 0 otherwise; the parameters w^k_e correspond to the number of transplants obtained by player k when edge e is matched and the parameters a^j_e take value 1 when edge e is incident to vertex j and 0 otherwise. Constraints (2.3.1b) ensures that no coalition can strictly improve the payoffs of all its members, while Constraint (2.3.1c) enforces y to be a matching.

In order to input Problem (2.3.1) into an off-the-shelf solver, we transform it into an IP problem. In particular, we will reformulate Constraints (2.3.1b). A standard procedure for handling disjunctive constraints is to introduce the binary variables $d^k_{M^S,S}$ to indicate whether Constraint (2.3.1b) is respected for player k under coalition S and the internal matching M^S, i.e., $d^k_{M^S,S} = 1$ if and only if player k does not improve their payoff by deviating with coalition S under matching M^S. Mathematically, we aim to model:

$$d^k_{M^S,S} = 1 \iff u^k_{M^S,S} \leq \textstyle\sum_{e \in E} w^k_e y_e \quad \forall k \in N,\ \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S).$$

To enforce this logical biconditional (equivalence), we can add the following two constraints:

$$(\Longrightarrow) \sum_{e\in E} w_e^k y_e + \underline{B}^k d_{M^S,S}^k \geq \underline{B}^k + u_{M^S,S}^k \qquad \forall k \in N,\ \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S) \quad (2.3.2a)$$

$$(\Longleftarrow) \sum_{e\in E} w_e^k y_e - (\bar{B}^k + 1) d_{M^S,S}^p \leq u_{M^S,S}^p - 1 \quad \forall k \in N,\ \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S), \quad (2.3.2b)$$

where $\bar{B}^k$ and $\underline{B}^k$ are, respectively, upper and lower bounds to $\sum_{e\in E} w_e^k x_e - u_{M^S,S}^k$. We can define valid bounds by setting $B = |V^k| + 1$ and $b = -B$. Combining all these elements, we obtain the following IP:

Weak core problem (2.3.3)

$$\max_{y,d} \quad \sum_{e\in E} y_e \qquad (2.3.3a)$$

$$\text{s.t.} \quad \text{Constraints (2.3.1c), (2.3.1d),(2.3.2a),(2.3.2b)}$$

$$\sum_{k\in S} d_{M^S,S}^k \geq 1 \qquad \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S) \qquad (2.3.3b)$$

$$d_{M^S,S}^k \in \{0,1\} \qquad \forall k \in N, \forall S \subseteq N,\ \forall M^S \in \mathcal{C}(S), \quad (2.3.3c)$$

with Constraints (2.3.3b) guaranteeing that, for each coalition and associated matching, at least one of its members has no incentive to deviate

Problem (2.3.3) has a number of constraints and decision variables that is exponential in the number of players. In particular, in order to write the mathematical formulation, for each coalition, we need to generate all their possible maximum matchings in order to calculate the values needed for Constraints (2.3.2a) and (2.3.2b). Doing so is a $\#P$- complete problem [36], even if a maximum matching of a graph $G = (V,E)$ can be found in run time $O(\sqrt{|V|}|E|)$ by the Micali and Vazirani algorithm [37]. Notice however that this only applies to coalitions of size greater than or equal to two, since only one maximum matching is needed for a single player. Due to the potentially large number of constraints and the time needed to generate them, solving Problem (2.3.3) can be computationally expensive. In the next section, we will show how we can exploit certain structures in the compatibility graphs to reduce the number of constraints we need to find the optimal solution to Problem (2.3.3).

2.3.3. Cutting plane approach

Before describing our approach to solve Problem (2.3.3), we provide some necessary background on matchings. Given a graph $G = (V,E)$ and a matching M of G, an M-*alternating path* is a path with edges alternating between M and $E \setminus M$. An M-*augmenting path* is an M-alternating path starting and ending in an M-unmatched vertex. The following theorem by Berge presents an important fact linking augmenting paths and maximum matchings:

Theorem 2 (Berge [38]). A matching is maximum if and only if it has no augmenting path.

Berge's theorem provides a simple algorithm to obtain a maximum matching. Given a non-maximum matching M, find an M-augmenting path p and produce a new matching, $M \oplus p$, where $\oplus$ represents the symmetric set difference. Repeat this process until no more augmenting paths exist. It is worth noting that the vertices which are matched in a step of the algorithm remain matched in the final maximum matching; in particular, the vertices in the matching M are also matched in $M \oplus p$.

Inspired by the optimality condition of Theorem 2 involving augmenting paths and similar ideas in [31, 30] to model players' incentives, we introduce a new type of alternating paths useful in our context.

Definition 5. Let $G = (V,E)$ be a compatibility graph for a kidney exchange instance, M be a matching of G and $S \subseteq N$ be a coalition. An S-*deviation path* from M is an alternating path starting in an M-matched vertex not belonging to S, then passing only through edges in E^S and ending in an M-unmatched vertex belonging to S.

Figure 2.3 provides an example of a deviation path.

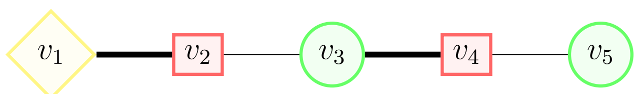


Figure 2.3. An S-deviation path from $M = \{(v_1,v_2),(v_3,v_4)\}$ where coalition S consists of the square and circle vertices

Deviation paths are an important structure for coalition deviation as shown by the following theorem:

Theorem 3. Let M be a maximum matching of $G = (V,E)$ and let $S \subseteq N$ be a coalition. Coalition S wishes to deviate from M only if there exists $|S|$ disjoint S-deviation paths from M.

Proof. Suppose S can deviate to the maximum matching M^S of G^S and let $M' = M^S \cup (M \cap E^{N \setminus S})$. We say that a vertex is newly matched in M^S when it is matched in M^S but not in M, and that it is previously matched in M when it is matched in M but not in M'. First, notice that since M is a maximum matching, for each newly matched vertex j in M^S, there must be an M-alternating path p starting at j and ending at another vertex w previously matched in M. Moreover, we have $(M \oplus p) \cap p = M^S \cap p$.

When S deviates, every edge $e \in M^S \setminus M$ must have endpoints in V^S. Hence, each vertex in p belongs to V^S, except for w which can either be in V^S or $V^{N \setminus S}$. If $w \in V^{N \setminus S}$, then p is an S-deviation path from M, and the number of matched vertices in V^S along p increases by 1. If not, p is not a deviation path and the number of matched vertices in V^S along p is the same in M' and M.

Since S has incentive to deviate, each player $k \in S$ must improve their payoff by at least 1. This means that the number of M'-matched vertices in V^S must augment by at least $|S|$.

Hence, there must be at least $|S|$ newly matched vertices that are the endpoint of deviating paths in P, given that only deviating paths can increase the coalition payoff. □

Theorem 3 gives a necessary condition for a coalition to deviate. This condition is, however, not sufficient. For instance, consider the example of Figure 2.4 where there are two deviation paths associated with the coalition formed by the square and the circle players: (v_1, v_2, v_3) and (v_4, v_5, v_6);. The square player, however, has no incentive to deviate with the circle player as their payoff would decrease by 1.

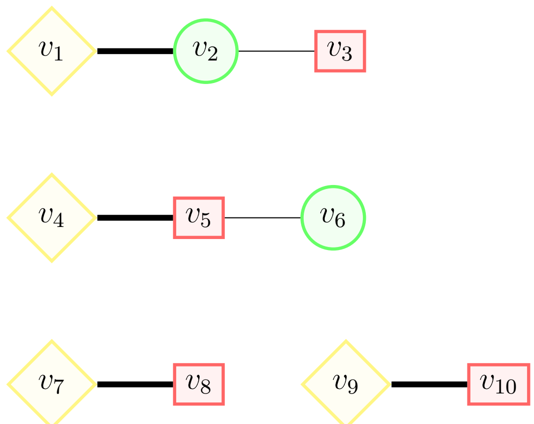


Figure 2.4. An insufficient condition for the coalition formed by the square and circle vertices to deviate under the matching $M = \{(v_1,v_2), (v_4,v_5), (v_7,v_8), (v_9,v_{10})\}$

Theorem 3 still gives us valuable tools to speed the solving of Problem (2.3.3). Since a coalition can only deviate if it has a certain number of disjoint deviation paths, we can start by calculating a maximum matching on G without using the coalitions constraints, i.e., solve a relaxation of Problem (2.3.3):

Main Program (2.3.4)

$$\max_{y} \quad \sum_{e \in E} y_e \tag{2.3.4}$$
$$\text{s.t.} \quad \text{Constraints (2.3.1c), (2.3.1d).}$$

We can then search for a coalition S with enough deviation paths and add all its constraints to Problem (2.3.4). This demands finding all maximum matchings for coalition S, i.e., the optimal solutions of

Coalition S Problem (2.3.5)

$$\max_{y} \quad \sum_{e \in E} y_e \tag{2.3.5a}$$
$$\text{s.t.} \quad \sum_{e \in E^S} a_e^j y_e \leq 1 \quad \forall j \in V^S \tag{2.3.5b}$$
$$y_e \in \{0,1\} \qquad \forall e \in E^S. \tag{2.3.5c}$$

which can be obtained by repeatedly solving it and adding no-good cuts.

We repeat this process until either all the coalitions constraints have been added or until we can no longer find a coalition with enough deviation paths in the last obtained matching. This is formalized in Algorithm 1. Since in this procedure we add sequentially constraints, our approach is a cutting plane method.[4]

Algorithm 1 Cutting plane method: Computation of a solution in the weak core

Input: A compatibility graph $G = (V,E)$ and set of players N
Output: A solution $y^\star$ in the weak core or a certificate of its non existence

$y^\star \leftarrow$ an optimal solution for **Main Program** (2.3.4) ▷ Binary vector encoding a matching in G
$\mathcal{P} \leftarrow 2^N$ ▷ Power set
while $\exists S \in \mathcal{P}$ s.t. there are $|S|$ S-deviation paths from $y^\star$ **do**
  $z \leftarrow$ an optimal solution for **Coalition S Problem** (2.3.5)
  $O \leftarrow$ the objective value associated to z in **Coalition S Problem** (2.3.5)
  while Coalition S Problem (2.3.5) has an optimal solution with objective value equal to O **do**
    With respect to z, add no-good cut from **Coalition S Problem** (2.3.5)
    $z \leftarrow$ an optimal solution for **Coalition S Problem** (2.3.5)
  end while
  $\mathcal{C}(S) \leftarrow$ all optimal solutions of **Coalition S Problem** (2.3.5)
  for $M^S \in \mathcal{C}(S)$ **do**
    for $k \in S$ **do**
      $\bar{B}^k \leftarrow |V^k|+1$, $\underline{B}^k \leftarrow -\bar{B}^k$
      With respect to coalition S and player k, add Constraints (2.3.2a), (2.3.2b) and (2.3.3c) to **Main Program** (2.3.4)
    end for
    With respect to coalition S, add Constraint (2.3.3b) to **Main Program** (2.3.4)
  end for
  $\mathcal{P} \leftarrow \mathcal{P} \setminus S$
  $y^\star \leftarrow$ an optimal solution of **Main Program** (2.3.4)
end while
Return $y^\star$

Theorem 4. Algorithm 1 outputs a solution in the weak core, or provides a certificate that there is no such solution if the weak core is empty.

PROOF. Let $G = (V,E)$ be a compatibility graph and let N be a set of players.

Suppose that the weak core of this instance is non-empty. In this case Problem (2.3.3) has a non-empty feasible region. Since all the possible cuts made in Algorithm 1 correspond to constraints of Problem (2.3.3), the main program in the algorithm always has a non-empty feasible region. Hence Algorithm 1 outputs a matching $y^\star$ as a solution of **Main**

[4]We remark that our approach can be seen as a simultaneous constraint and column generation method since the progressively added constraints have associated variables $d^k_{M^S,S}$. However, as these are auxiliary variables, we use the terminology cutting plane.

Program (2.3.4) in this case. Let $T \subset 2^N$ be the set of coalitions whose constraints have been added to **Main Program** (2.3.4), and let $T' = 2^N \setminus (N \cup T)$. Now take a coalition $S \in 2^N$. If $S \in T$, then the constraints associated with S guarantees that S does not have the incentive to deviate under $y^\star$. If $S \in T'$, then our algorithm guarantees that the number of S-deviating paths in $y^\star$ is less than $|S|$. By Theorem 3, this again shows that S has no incentive to deviate under $y^\star$. Therefore, no coalition has incentive to deviate from $y^\star$ and this solution is in the weak core.

Now suppose that the weak core is empty. Then, by Theorem 3, given any maximum matching $y^\star$ obtained in an iteration of the algorithm, it is possible to find a coalition S such that the number of S-deviation paths in $x^\star$ is at least $|S|$. Hence, as long as **Main Program** (2.3.4) has a non-empty feasible region, Algorithm 1 provides a cut associated with a coalition whose constraints are not yet added to **Main Program** (2.3.4). This process ends either when the feasible region of **Main Program** (2.3.4) is empty, or when all the possible constraints have been added. In the latter case, the feasible region of **Main Program** (2.3.4) is also empty, since it becomes the feasible region of Problem (2.3.3). We therefore have a certificate of the non-existence of a solution in the weak core. □

2.4. Relation with rejection-proof mechanisms

In this section, we review another solution concept for the N-KEP: the rejection-proof mechanism. These mechanism work by proposing an exchange plan such that no player has an incentive to refuse it. We then theoretically study how it compares to our weak core concept by analyzing when it is possible to obtain a matching that is both rejection-proof and in the weak core.

2.4.1. Rejection-proof kidney exchange mechanism

A rejection-proof kidney exchange mechanism is a solution concept designed by Blom et al. [32]. Although Blom et al. [32] present their ideas for a context where exchanges of arbitrary size are allowed, we present it from a 2-way exchange perspective, with the aim to ease the comparison of this solution with the weak core.

In their setting, Blom et al. [32] consider a mechanism that begins by proposing a matching on the compatibility graph, and then each player decides on a rejection strategy. A player's rejection strategy consists of deciding which proposed exchanges involving their vertices they accept. Formally, it is defined as follows: Let $G = (V,E)$ be a compatibility graph and let N be a set of players. First, a mechanism f selects a maximum matching $f(G)$. Then, every player $k \in N$ selects a matching M^k on G such that $\forall e \in M^k,\ V(e) \cap V^k \neq \emptyset$ and $M^k \cap E^I \subseteq f(G) \cap E^I$. This matching is called the player's rejection strategy. Lastly, a final matching M is found where M consists of the exchanges accepted by the players, formally, $\forall e \in M$ it holds that $\forall k \in N, \Big(V(e) \cap V^k \neq \emptyset \implies e \in M^k\Big)$.

When selecting their rejection strategies, each player k aims to optimize their payoff, i.e., player k solves the following IP:

$$\text{RKEP}(f(G),k)$$

$$\max_{y} \quad \sum_{e \in E} w_e^k y_e \tag{2.4.1a}$$

$$\text{s.t.} \quad \text{Constraints (2.3.1c), (2.3.1d)}$$

$$y_e = 0 \quad \forall e \in E^I \setminus M. \tag{2.4.1b}$$

We say that a player k accepts a proposed matching $f(G)$ if their rejection strategy is $M^k = \{e \in f(G) \mid V(e) \cap V^k \neq \emptyset\}$ and rejects it otherwise.

Definition 6. A mechanism f is *rejection-proof* if for every compatibility graph G and set of players N, it holds that for any player $k \in N$, $\text{RKEP}(f(G),k) \leq x_k(f(G))$.

We can informally say that a matching M is rejection-proof if $\text{RKEP}(M,k) \leq x_k(M)\ \forall k \in N$ (i.e., without specifying the mechanism f).

Now that we have described the game by Blom et al. [32], we are ready to present our approach for computing a rejection-proof matching. Contrary to the general approach in [32], we explore the 2-way exchanges structure. This allows us to devise a simple and effective method, as well as to easily clarify the relation with other solutions, namely, Nash equilibrium in the withholding game and weak core. The next, we define the fundamental ingredient used by our approach:

Definition 7. Let $G = (V,E)$ be a compatibility graph for a kidney exchange instance, N be a set of players, and M be a maximum matching on G. An *M-rejection path* for player $k \in N$ is an alternating path in M starting in a matched vertex belonging to a player $k' \neq k$, then passing only through vertices belonging to player k and ending in an unmatched vertex belonging to player k.

Figure 2.5 provides an illustration of a rejection path. When searching for a rejection-proof solution, rejection paths act in a way similar to deviation paths when searching for a solution in the weak core. In fact, they are even more powerful than deviation paths, since they provide a necessary and sufficient condition for a matching M to be rejection-proof, as shown by the following lemma.

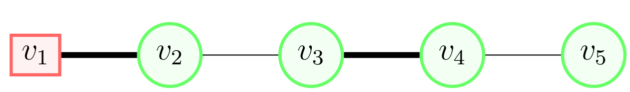


Figure 2.5. An M-rejection path for the circle player where $M = \{(v_1,v_2),(v_3,v_4)\}$

Lemma 5. Let G be a compatibility graph and let N be a set of players. A maximum matching M on G is rejection-proof if and only if for every player $k \in M$, there is no M-rejection path for that player.

Proof. Let M be a maximum matching on G.

$\Longleftarrow$: Suppose that there is a player $k \in N$ with an M-rejection path p. Then this player can select $M^k = (M \oplus p) \cap (E^k \cup E^I_k)$ as a rejection strategy and obtain a better payoff than the one they would get by accepting the proposed solution. Hence, M is not rejection-proof.

$\Longrightarrow$: Suppose M is not rejection-proof. Then, there is a player $k \in N$ with RKEP($f(G)$,k)$> x_k(M)$. Let G^I_k be the subgraph of G induced by keeping only the edges in $E^k \cup E^I_k$. Notice that restricting M to G^I_k still yields a maximum matching on G^I_k. Now, since player k rejects M, there is a vertex $v \in V^k$ unmatched in M, but matched in M^k (player k's rejection strategy). Since M is a maximum matching, there is an M-alternating path p starting in v and ending in a vertex v', matched in M and unmatched in M^k. Moreover, any international edge in M^k must also be in M. Hence, all vertices along p are in V^k except for v', and thus p is a rejection path for player k. □

We note here that rejection paths can be regarded as a special case of the alternating paths used in Carvalho and Lodi [31] to provide a necessary and sufficient condition for a matching to be a Nash equilibrium in the withholding game. Hence, every Nash equilibrium of the game defined by Carvalho et Lodi is also a rejection-proof matching. Moreover, since it is always possible to find a Nash equilibrium of maximum cardinality for any given N-KEP instance, it is also always possible to find a rejection-proof matching of maximum cardinality.

Lemma 5 gives us the tool we need to design an algorithm similar to Algorithm 1 for the computation of a rejection-proof matching. To this end, we first define an integer program allowing us to find a rejection-proof matching:

Rejection-proof problem (2.4.2)

$$\max_{y} \quad \sum_{e \in E} y_e \tag{2.4.2a}$$

$$\text{s.t.} \quad \text{Constraints (2.3.1c), (2.3.1d)}$$

$$\sum_{e \in p | y_e = 1} (1 - y_e) + \sum_{e \in p | y_e = 0} y_e \geq 1 \quad \forall p \in \mathcal{RP}_G, \tag{2.4.2b}$$

where $\mathcal{RP}_G$ is the set of all possible rejection paths on G. Constraints (2.4.2b) amount to require the matching to have no rejection path. Now that we have our IP, potentially with an exponential number of Constraints (2.4.2b), we can again use a cutting plane method to compute a rejection-proof matching.

Algorithm 2 Cutting plane method: Computation of a rejection-proof matching

Input: A compatibility graph $G = (V,E)$ and set of players N

Output: A rejection-proof matching $y^\star$

$y^\star \leftarrow$ an optimal solution for **Main Program** (2.3.4) $\triangleright$ Binary vector encoding a matching in G
while $\exists p \in \mathcal{RP}_G$ such that $p \subseteq y^\star$ **do**
  With respect to p, add Constraint (2.4.2b) to **Main Program** (2.3.4)
  $y^\star \leftarrow$ an optimal solution for **Main Program** (2.3.4) .
end while
Return $y^\star$

Lemma 6. Algorithm 2 produces a rejection-proof matching.

Proof. By construction, the matching $y^\star$ produced by Algorithm 2 contains no rejection path. Since the presence of a rejection path is a necessary and sufficient condition for a matching to be non rejection-proof, $y^\star$ is therefore rejection-proof. □

With Algorithm 2, we are now equipped to compare rejection-proof matchings with matchings in the weak core.

2.4.2. Relation to the weak core

While looking at the weak core and the rejection-proof matching as solution concepts for the N-KEP, one may notice the similarities that the two share. Both rely on an independent agent first proposing a solution, and then the players have to decide whether they accept the solution. Moreover, we can also easily see that rejection paths are a special case of deviation paths, as remarked in the previous section.

These similarities lead us to compare the sets of matchings respecting the criteria for both solution concepts. Given a graph G, we will call $\mathcal{WC}_G$ the set of maximum matchings on G being in the weak core and $\mathcal{R}_G$ the set of maximum rejection-proof matchings on G. As before, the two players case is easier to analyze. In that case, we saw that every Nash equilibrium of the withholding game of Carvalho and Lodi [31] is in the weak core as well as being rejection-proof. Hence, $\mathcal{WC}_G \cap \mathcal{R}_G \neq \emptyset$. In fact, since every rejection-path is also a deviation path, and since the only possible deviating coalitions in the two players setup are $\{1\}$ and $\{2\}$, we have that in this case, $\mathcal{R}_G \subseteq \mathcal{WC}_G$. This leads us to think that the rejection-proof matching might be a coarser solution concept.

However, the situation differs when there are more than two players. As shown in section 2.5.2, there are instances for which the weak core might be empty, while it is always possible to find a rejection-proof matching. Therefore, the intersection between $\mathcal{WC}_G$ and $\mathcal{R}_G$ might be empty in some cases. In practice however, our computational results, presented in the next section, show that it is quite possible to find a rejection-proof matching in the weak core.

To compute a rejection-proof solution that is also in the weak core, we combine Algorithm 1 and Algorithm 2 to obtain the following:

Algorithm 3 Cutting plane method: Computation of a rejection-proof matching in the weak core

---

Input: A compatibility graph $G = (V,E)$ and set of players N

Output: A rejection-proof matching $y^\star$ in the weak core or a certificate of its non existence

$y^\star \leftarrow$ an optimal solution for **Main Program** (2.3.4) ▷ Binary vector encoding a matching in G

$\mathcal{P} \leftarrow 2^N$ ▷ Power set

while $\exists p \in \mathcal{RP}_G$ such that $p \subseteq y^\star$ **or** $\exists S \in \mathcal{P}$ s.t. there are $|S|$ S-deviation paths from $y^\star$ **do**

  if $\exists p \in \mathcal{RP}_G$ such that $p \subseteq y^\star$ **then**

    do Step 3 of Algorithm 2

  end if

  if $\exists S \in \mathcal{P}$ s.t. there are $|S|$ S-deviation paths from $y^\star$ **then**

    do Steps 4-18 of Algorithm 1

  end if

  $y^\star \leftarrow$ an optimal solution for **Main Program** (2.3.4) .

end while

Return $y^\star$

2.5. Computational experiments

In this section, we discuss and present the results of our computational experiments. Our goals are *(i)* to test if, in practice, we are able to find matchings in the weak core, and *(ii)* to ascertain whether, in practice, there is a non-empty overlap between the set of rejection-proof matchings and the set of matchings in the weak core. Section 2.5.1 presents an overview of our experimental setup, describing the instances we used as well as our computational environment. In Section 2.5.2, we illustrate an example of an instance with an empty weak core, motivating test *(i)*. Finally, Section 2.5.3 present the discussion of the results, where for the instances mimicking real-world compatibility graph topologies, we are able to find matchings in the weak core and rejection-proof matchings in the weak core.

2.5.1. Experimental setup

Instances generators. In our computational experiments, we generate random compatibility graphs that are based on real-world data. For this purpose, we use the generators provided in the Julia package developed by Omer et al. [17]. The compatibility graphs are created using three different generators found within the package.

The first is the Saidman generator. It works by generating patients with random characteristics influencing compatibility, taken from real-world distributions. We refer the reader to Saidman et al. [39] for further details. The second generator is the sparse generator, introduced in Dickerson et al. [40]. It is based on the Saidman generator. It uses updated distributions of characteristics for the patients. This results in sparser instances, since the probability of a patient being incompatible with a donor is increased. The last generator is the heterogeneous generator introduced in Ashlagi et al. [41]. Based on real world observations, it separates patients into two types: easy to match and hard to match. The graphs created therefore possess a disparity between the vertex degrees of their nodes.

All three generators produce directed graphs. This means that a directed edge will be placed from a vertex v to a vertex v' if the donor of v is compatible with the patient of v'. To adapt to our context, we transform an obtained directed graph $D = (V, A)$ into a non-directed one $G = (V, E)$ (representing possible 2-way exchanges) by setting $e = (v,v') \in E$ if and only if (v,v'), $(v',v) \in A$. On top of transforming the instances into non-directed graphs, we also need to assign a player to each vertex of the graph. In an n-player instance, we assign the first n vertices of the graph to players from 1 to n, ensuring that each player owns at least a vertex. Afterwards, every remaining vertex is randomly assigned a player, with each player having equal probabilities of being assigned.

Parameters and instances. In total, 720 instances were tested. Concretely, for each combination of the following parameters, we generated 20 instances: the topology of the graph (based on the generator used: Saidman, sparse or heterogeneous), the number of vertices (20, 30, 40 or 50) and the number of players (3, 4 or 5).

Computational environment. The algorithms to solve the weak core and the rejection-proof problems were implemented in Python 3.10.9. All the integer programs were solved using Gurobi 10.0.0 with two threads and the code runs on an Intel(R) Xeon(R) Gold 6226 CPU on 2.70GHz, running Linux 7.9.

2.5.2. The weak core can be empty

The example in Figure 2.6 shows that there are instances with empty weak core. Although this example shows that we have no guarantee of finding a solution in the weak core when dealing with an arbitrary compatibility graph, we observe that the compatibility graph looks quite *artificial*. Indeed, it seems improbable to encounter this kind of graph topology in a real-world example. As a matter of fact, the results of our computational experiments, presented in Section 2.5.3, consistently demonstrate that in practice, we are always able to find a solution in the weak core.

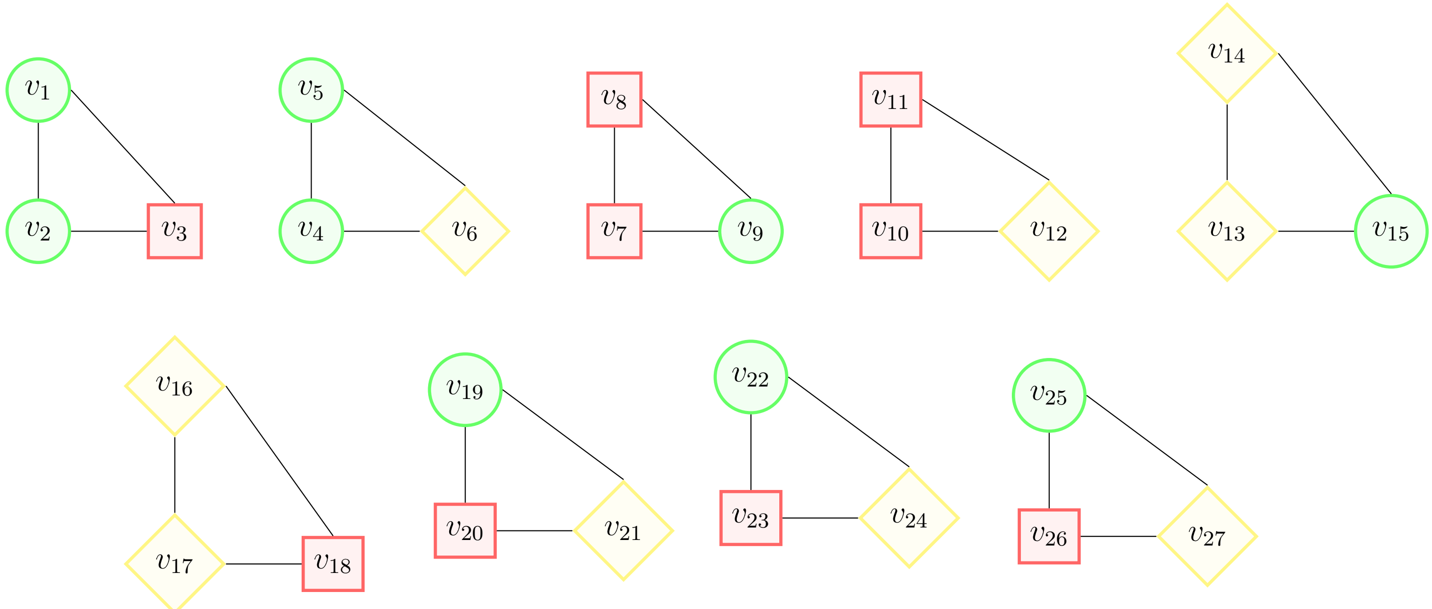


Figure 2.6. A kidney exchange instance with 3 players and an empty weak core

2.5.3. Computational results

For our computational experiments, we ran three approaches on all our instances: Algorithm 1, Algorithm 3 and a baseline consisting of solving Problem (2.3.3) with an off-the-shelf optimization solver.[5] We set a time limit of 30 minutes for applying each approach to each instance. Then, we identified instances for which we could find a matching in the weak core, as well as instances for which we could find a rejection-proof matching in the weak core.

Weak core in practice. Our computational experiments show that in practice, we are always able to find a matching in the weak core in the N-KEP. Indeed, for all the instances for which the execution time of the baseline or Algorithm 1 did not reach the limit, a matching in the weak core was found. Using Algorithm 1, we were able to prove the non-emptiness of the weak core for all of the sparse instances, for all but one of the heterogeneous instances and for 90% of the Saidman instances; for the remaining instances Algorithm 1 did not finish within the time limit. With the baseline, more instances lead to the time limit being reached, preventing its use to prove or disprove the non-emptiness of the weak core. Indeed, in this case, no sparse instances exceeded the time limit, but 3% of the heterogeneous instances and 15% of the Saidman instances did. We refer the reader to Appendix A for a details on these results, including a more thorough analysis of the performance of Algorithm 1 and the baseline. These results are encouraging as they support our claim that, despite the theoretical evidence of the possibility of an empty weak core, in practice, the weak core for the N-KEP

[5]Our interest is not solely in the computation of a rejection-proof matching. Hence, Algorithm 2 is not directly used (note that it is part of Algorithm 3).

is non-empty. The topology of graphs inspired by real-world data, combined with the fact that the weak core is a sufficiently relaxed solution concept, helps make this possible.

Comparison to rejection-proof matchings. Finally, we present and discuss the results of the application of Algorithm 3 on our instances. In summary, once again, in all the investigated instances for which the time limit was not attained, we were able to find a rejection-proof matching in the weak core. We were able to solve all the sparse and heterogeneous instances, and 95% of the Saidman instances. Having a matching meeting the requirements of multiple solution concepts is valuable, as the matching possesses more desirable qualities. In this case, a rejection-proof matching in the weak core has stability over deviations from coalitions of players and has stability over rejections of subsets of the matching from single players.

Tables 2.1, 2.2 and 2.3 provide further details on the instances and the existence of rejection-proof solutions in the weak core. Each row gives statistics over instances sharing the same number of vertices and players. The column *Solved*, gives the percentage of instances for which the algorithm ended before reaching the time limit and the column *Weak Core and Rejection-Proof* gives the percentage of instances among the solved ones for which we found a rejection-proof matching in the weak core. We remark here that the generator we used does not guarantee that all the instances have edges. For the sparse topology, many instances with up to 30 vertices had no edges. We refer the reader to Appendix A for more details on the instances.

Instances			Solved	Weak Core and Rejection-Proof
$\lvert V\rvert$	n	$\lvert E\rvert$	%	%
20	3	0.75	100	100
	4	0.35	100	100
	5	0.80	100	100
30	3	0.70	100	100
	4	0.55	100	100
	5	1.05	100	100
40	3	1.85	100	100
	4	2.05	100	100
	5	1.90	100	100
50	3	3.00	100	100
	4	2.55	100	100
	5	3.10	100	100

Table 2.1. Statistics on rejection-proof solutions in the weak core for sparse graphs using Algorithm 3

Instances			Solved	Weak Core and Rejection-Proof
$\lvert V \rvert$	n	$\lvert E \rvert$	%	%
20	3	11.74	100	100
	4	11.58	100	100
	5	10.58	100	100
30	3	27.63	100	100
	4	25.79	100	100
	5	26.00	100	100
40	3	47.79	100	100
	4	46.26	100	100
	5	47.63	100	100
50	3	76.42	100	100
	4	74.26	100	100
	5	76.47	100	100

Table 2.2. Statistics on rejection-proof solutions in the weak core for heterogeneous graphs using Algorithm 3

Instances			Solved	Weak Core and Rejection-Proof
$\lvert V \rvert$	n	$\lvert E \rvert$	%	%
20	3	10.42	100	100
	4	11.74	100	100
	5	9.00	100	100
30	3	26.47	100	100
	4	21.53	100	100
	5	26.84	100	100
40	3	43.68	100	100
	4	38.47	100	100
	5	43.26	100	100
50	3	66.11	90	100
	4	63.32	85	100
	5	52.16	70	100

Table 2.3. Statistics on rejection-proof solutions in the weak core for Saidman graphs using Algorithm 3

2.6. Conclusion

In this paper, we formalize the N-KEP as a cooperative non-transferable utility game. This allows us to model the fact that even if a coalition could obtain a better total payoff by deviating, some of its members might be worse off and might not want to deviate. We then apply a solution concept to this game, *the weak core*, which is a more flexible version of the widely used *core*. Using a weaker version of the core allows us to work with a solution concept that has a higher likelihood of existing. By investigating certain graph structures, we are able to find a necessary condition for a matching to be in the weak core and we use this necessary condition to develop a cutting plane method to solve the integer programming formulation of the weak core problem. Although, we provide an example of a KE instance with an empty weak core, we show through computational experiments that we are able to find a matching in the weak core for instances from the literature, based on real-world data.

We also compare our solution concept to a recent one from the literature in non-cooperative game theory: the rejection-proof matching. We theoretically analyze the relations between the two concepts and show through computational experiments that it is possible, in practice, to find rejection-proof matchings in the weak core. This, together with the brief comparison we make between the weak core and the Nash equilibrium, encourages further inquiries on the comparison of the cooperative and non-cooperative game theory approach for the N-KEP.

Finally, we analyze the performance of our proposed cutting plane method to compute matchings in the weak core. We show that it performs better in practice than a baseline consisting of solving an IP with a standard solver, although further work would be needed to help the method perform well on real-world size instances.

Our findings indicate that in practical scenarios, the need to use a credit system in cooperative approaches might not be necessary if the weak core concept is employed. Indeed, future research should explore this and other solution concepts for the N-KEP while accounting for its non-transferable utility, as well as encompassing general L-way exchanges.

Acknowledgements

The authors wish to thank the support of the Institut de valorisation des données and Fonds de Recherche du Québec through the FRQ–IVADO Research Chair in Data Science for Combinatorial Game Theory, and the Natural Sciences and Engineering Research Council of Canada through the discovery grant 2019-04557. They also wish to thank Hydro-Québec for their support through the Bourse d'excellence Hydro-Québec.

Chapter 3

Conclusion

In this thesis, we formulated the N-KEP as a non-transferable utility cooperative game. This enable us to address the inherent nature of kidney transplants as non-transferable resources. This means that for each coalition of agents, there is a set of feasible payoff vectors, underscoring the reality that not all possible payoff distributions are attainable, as is the case for transferable utility games. As a solution for our game, we introduced the weak core. A matching is in the weak core if no coalition can deviate and make all its members strictly better off. Using a special kind of alternating path, we gave a necessary condition for a matching to be in the weak core. This allowed us to devise a cutting plane method to solve the integer programming formulation of the weak core problem. We provided an example of a kidney exchange instance with an empty weak core. Despite this, we showed through computational experiments that we are able to find matchings in the weak core in practice. The experiments were conducted on a set of instances created with three generators from the literature that are based on real-world data. We also compared the weak core to the concept of rejection-proof matching. Once again, through computational experiments on the same set of instances, we showed that we are able to combine the two solution concepts and find rejection-proof matchings in the weak core. Finally, we analyzed the performance of our cutting plane method and showed that it generally performed better than a baseline consisting of solving an IP with an off-the-shelf solver. Our work indicates that in practice, the use of the weak core might reduce the need of a credit system. It also suggests that it is possible to ask for criteria from multiple solution concepts when searching for a matching, meaning that the solutions might possess various desirable properties.

For future research, it would be interesting to investigate the generalization of our work to L-way exchanges. Moreover, further inquiries on the intersection of different solution concepts from cooperative and non-cooperative game theory could help decide which of them to employ, and maybe combine.

References

[1] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton classic editions. Princeton University Press, Princeton, N.J. ; Woodstock, 60th anniversary ed edition, 2007. ISBN 978-0-691-13061-3.

[2] Alfonso H Santos Jr, Michael J Casey, Xuerong Wen, Ivan Zendejas, Shehzad Rehman, Karl L Womer, and Kenneth A Andreoni. Survival with dialysis versus kidney transplantation in adult hemolytic uremic syndrome patients: A fifteen-year study of the waiting list. *Transplantation*, 99(12):2608–2616, 2015.

[3] Felix T Rapaport. The case for a living emotionally related international kidney donor exchange registry. 18(3) Suppl. 2):5–9, 1986.

[4] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Kidney exchange. *The Quarterly journal of economics*, 119(2):457–488, 2004.

[5] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. A kidney exchange clearinghouse in new england. *American Economic Review*, 95(2):376–380, 2005.

[6] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Pairwise kidney exchange. *Journal of Economic theory*, 125(2):151–188, 2005.

[7] 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.

[8] 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.

[9] Alvin E Roth, Tayfun Sönmez, and M Utku Unver. Transplant center incentives in kidney exchange. *Harvard University and Koç University, unpublished memo*, 2005.

[10] Péter Biró, Joris Van de Klundert, David Manlove, William Pettersson, Tommy Andersson, Lisa Burnapp, Pavel Chromy, Pablo Delgado, Piotr Dworczak, Bernadette Haase, et al. Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research*, 291(2):447–456, 2021.

[11] 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*, pages 295–304, 2007.

[12] 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, 2007.

[13] 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.

[14] Radu-Stefan Mincu, Péter Biró, Márton Gyetvai, Alexandru Popa, and Utkarsh Verma. Ip solutions for international kidney exchange programmes. *Central European Journal of Operations Research*, 29:403–423, 2021.

[15] 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*, pages 25–42, 2016.

[16] Lizeth Carolina Riascos-Álvarez, Merve Bodur, and Dionne M Aleman. A branch-and-price algorithm enhanced by decision diagrams for the kidney exchange problem. *arXiv preprint arXiv:2009.13715*, 2020.

[17] Jérémy Omer, Ayse N Arslan, and Fulin Yan. Kidneyexchange. jl: A julia package for solving the kidney exchange problem with branch-and-price, 2022. URL `https://inria.hal.science/hal-03830810/document`.

[18] Georg A Böhmig, Jiří Fronek, Antonij Slavcev, Gottfried F Fischer, Gabriela Berlakovich, and Ondrej Viklicky. Czech-Austrian kidney paired donation: first European cross-border living donor kidney exchange. *Transplant International*, 30(6):638–639, 2017.

[19] María Oliva Valentín, Marta Garcia, Alessandro Nanni Costa, Catarina Bolotinha, Lluis Guirado, Fabio Vistoli, Alberto Breda, Pamela Fiaschetti, and Beatriz Dominguez-Gil. International cooperation for kidney exchange success. *Transplantation*, 103(6):e180–e181, 2019.

[20] APKD. Pilot Kidney Exchange Transplant Program Launched Between US, Italy | Alliance for Paired Kidney Donation, September 2022. URL `https://paireddonation.org/pilot-kidney-exchange-transplant-program-launched-between-us-italy/`. Accessed: 23/08/2023.

[21] Xenia Klimentova, Ana Viana, João Pedro Pedroso, and Nicolau Santos. Fairness models for multi-agent kidney exchange programmes. *Omega*, 102:102333, 2021.

[22] Péter Biró, Márton Gyetvai, Xenia Klimentova, João Pedro Pedroso, William Pettersson, and Ana Viana. Compensation scheme with shapley value for multi-country kidney exchange programmes, 2020.

[23] Lloyd S Shapley. Notes on the n-person game—ii: The value of an n-person game.(1951). *Lloyd S Shapley*, 7, 1951.

[24] Péter Biró, Walter Kern, Dömötör Pálvölgyi, and Daniel Paulusma. Generalized matching games for international kidney exchange. 2019.

[25] Márton Benedek, Péter Biró, Walter Kern, and Daniël Paulusma. Computing balanced solutions for large international kidney exchange schemes. *arXiv preprint arXiv:2109.06788*, 2021.

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

[27] Itai Ashlagi, Felix Fischer, Ian A Kash, and Ariel D Procaccia. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior*, 91:284–296, 2015.

[28] Chen Hajaj, John Dickerson, Avinatan Hassidim, Tuomas Sandholm, and David Sarne. Strategy-proof and efficient kidney exchange using a credit mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[29] John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[30] Margarida Carvalho, Andrea Lodi, Joao Pedro Pedroso, and Ana Viana. Nash equilibria in the two-player kidney exchange game. *Mathematical Programming*, 161:389–417, 2017.

[31] Margarida Carvalho and Andrea Lodi. A theoretical and computational equilibria analysis of a multi-player kidney exchange program. *European Journal of Operational Research*, 305(1):373–385, 2023.

[32] Danny Blom, Bart Smeulders, and Frits CR Spieksma. Rejection-proof kidney exchange mechanisms. *arXiv preprint arXiv:2206.11525*, 2022.

[33] Donald Gillies. Solutions to general nonzero sum games. *Annals of Mathematical Studies*, pages 47–85, 1959.

[34] Márton Benedek, Péter Biró, Walter Kern, Dömötör Pálvölgyi, and Daniël Paulusma. Partitioned matching games for international kidney exchange. *arXiv preprint arXiv:2301.13181*, 2023.

[35] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. *Computational aspects of cooperative game theory*. Springer Nature, 2022.

[36] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[37] Silvio Micali and Vijay V Vazirani. An o (v| v| c| e|) algoithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*, pages 17–27. IEEE, 1980.

[38] Claude Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences*, 43(9):842–844, 1957.

[39] Susan L Saidman, Alvin E Roth, Tayfun Sönmez, M Utku Ünver, and Francis L Delmonico. Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation*, 81(5):773–782, 2006.

[40] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Optimizing kidney exchange with transplant chains: Theory and reality. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 711–718, 2012.

[41] Itai Ashlagi, Patrick Jaillet, and Vahideh H. Manshadi. Kidney exchange in dynamic sparse heterogenous pools. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, page 25–26, New York, NY, USA, 2018. Association for Computing Machinery.

Appendix A

Appendix

A.1. Performance results

To help us validate the effectiveness of Algorithm 1, we compare it to a natural baseline consisting of solving Problem (2.3.3) directly with a mixed-integer programming solver. A time limit of 30 minutes was set to solve each instance for all our experiments.

Tables A.1, A.2 and A.3 display the results for the Saidman, heterogeneous and sparse instances respectively. Note that each row presents average statistics over 20 instances, except for the columns with %. Concretely, in the section *Instances*, the first three statistics provide an indication of their sizes and the percentage of graphs with no edges (Trivial). The values in the *Solved* section are the percentage of instances which were solved before reaching the time limit (%) and the average solving time for the instances which did not time out (Time). The values in the *IP* (2.3.3) section are the percentage of instances for which all the constraints of Problem (2.3.3) were generated, (Gen.), the time taken to generate those constraints for instances that did not reach the time limit (Cons. Time), the solving time of the IP once it is generated, but without its generation time (Solve time), and the number of coalition constraints (2.3.2a) and (2.3.2b) (# Cons.). Finally, the section *Weak Core* contains the percentage of instances for which we were able to find a matching in the weak core. The parameters registering the number of coalition constraints and the percentage of instances having a non-empty weak core have only been taken into account if the instance to which they are associated did not timed out.

First, as might be expected, we observe that the more edges there are in the graph, the more time it takes to solve the IP. This is because a higher number of edges corresponds to an increased number of variables within the Problem (2.3.3), primarily leading to a greater potential for a large number of maximum matchings on the subgraphs of the coalitions. Since generating constraints (2.3.2a) and (2.3.2b) involves listing all possible maximum matchings on G^S for each coalition S, (recall that this is a $\#P$-complete problem), it makes sense that augmenting the number of edges greatly affects the solving time of the IP. This means that sparse instances are significantly easier to solve than heterogeneous and Saidman instances, with the graphs of the latter being much denser on average. No sparse instances reached the time limit, while some heterogeneous instances with 50 vertices and many Saidman instances with 30, 40 or 50 vertices did. Another factor influencing the run time is the number of players. For a given graph topology with a fixed number of vertices, a higher number of players always leads to a greater number of constraints (2.3.2a) and (2.3.2b) and is almost always linked to a higher construction time for the model. Once again this is to be expected, since the number of constraints (2.3.2a) and (2.3.2b) is exponential in the number of players.

Following this, we also note that the vast majority of the time taken to solve the IP is the time used to construct the model, more specifically, to generate the constraints. In fact, the solving time is so negligible that we might say that only the construction time affects the total solving time.

Overall, the performance of the baseline is reasonable when the instances are small and the number of players is low. However, the performance drops significantly as the number of edges increases, resulting in a higher percentage of heterogeneous and Saidman instances timing out.

A.1.1. Algorithm 1 performance

Instances				Solved		IP (2.3.3)				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Gen. (%)	Cons. Time	Solve time	# Cons.	%
20	3	0.75	65	100	0.01	100	0.01	0.00	1.15	100
	4	0.35	80	100	0.01	100	0.01	0.00	2.25	100
	5	0.80	45	100	0.03	100	0.03	0.00	13.9	100
30	3	0.70	70	100	0.01	100	0.01	0.00	0.95	100
	4	0.55	65	100	0.02	100	0.02	0.00	3.60	100
	5	1.05	50	100	0.05	100	0.05	0.00	13.25	100
40	3	1.85	25	100	0.02	100	0.02	0.00	2.45	100
	4	2.05	15	100	0.05	100	0.05	0.00	9.35	100
	5	1.90	10	100	0.09	100	0.	0.00	23.95	100
50	3	3.00	10	100	0.07	100	0.07	0.00	3.05	100
	4	2.55	15	100	0.06	100	0.06	0.00	9.70	100
	5	3.10	0	100	0.14	100	0.14	0.00	28.55	100

Table A.1. Statistics for the baseline on sparse graphs.

Next, we analyze the performance of Algorithm 1 and compare it to the baseline. Tables A.4, A.5 and A.6 present the results for the sparse, heterogeneous and Saidman instances respectively.

Instances				Solved		IP (2.3.3)				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Gen. (%)	Cons. Time	Solve Time	# Cons.	%
20	3	11.74	0	100	0.05	100	0.04	0.00	4.42	100
	4	11.58	0	100	0.13	100	0.12	0.00	18.00	100
	5	10.58	0	100	0.23	100	0.22	0.00	38.47	100
30	3	27.63	0	100	0.17	100	0.15	0.00	5.47	100
	4	25.79	0	100	0.44	100	0.42	0.00	21.26	100
	5	26.00	0	100	0.85	100	0.83	0.01	52.53	100
40	3	47.79	0	100	0.94	100	0.91	0.00	5.68	100
	4	46.26	0	100	2.20	100	2.16	0.01	22.26	100
	5	47.63	0	100	8.27	100	8.23	0.01	56.89	100
50	3	76.42	0	100	79.49	100	79.43	0.00	5.68	100
	4	74.26	0	90	90.34	90	90.29	0.01	20.67	100
	5	76.47	0	70	284.95	70	284.88	0.02	65.08	100

Table A.2. Statistics for the baseline on heterogeneous graphs

Instances				Solved		IP (2.3.3)				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Gen. (%)	Cons. Time	Solve Time	# Cons.	%
20	3	10.42	0	100	0.34	100	0.33	0.00	4.79	100
	4	11.74	5	100	0.16	100	0.16	0.00	18.47	100
	5	9.00	0	100	0.23	100	0.23	0.00	44.53	100
30	3	26.47	0	100	40.47	100	40.45	0.00	7.16	100
	4	21.53	0	100	15.36	100	15.35	0.00	28.47	100
	5	26.84	0	95	28.61	95	28.58	0.02	81.94	100
40	3	43.68	0	95	16.78	95	16.75	0.00	9.22	100
	4	38.47	0	90	170.87	90	170.84	0.01	38.11	100
	5	43.26	0	45	109.65	45	109.58	0.05	150.67	100
50	3	66.11	0	80	163.48	80	163.43	0.00	11.13	100
	4	63.32	0	60	481.70	60	481.63	0.02	60.55	100
	5	52.16	0	55	120.49	55	120.42	0.04	132.40	100

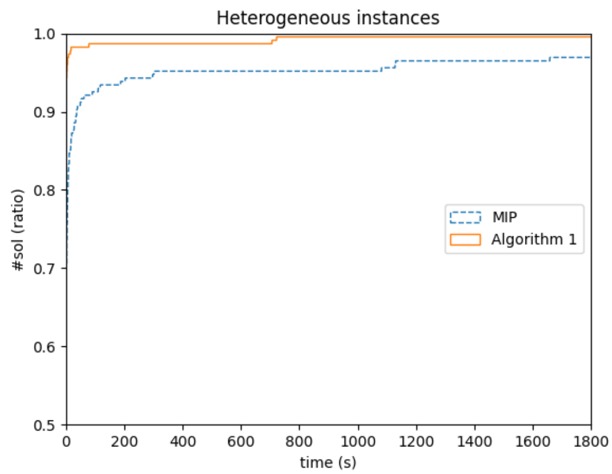
Table A.3. Statistics for the baseline on Saidman graphs

The parameters for the *Instances* and *Solved* sections are the same as in the previous tables. The parameters for the *Algorithm* section are the **main problem** 2.3.4 time (Time M.P.), the time taken to generate all the sets of constraints (2.3.2a) and (2.3.2b) for the instances that did not time out (Time Coal.), and all the instances (Time Coal. with T.O.), the number of sets of constraints (2.3.2a) and (2.3.2b) (# Cons.) and the number of times the main problem is solved within the while loop (Iterations). Finally, the parameter in the column *Weak Core* is the same as before. For the number of constraints, the number of iterations, and the percentage of instances having a solution in the weak core, only the instances that did not exceed the time limit were considered.

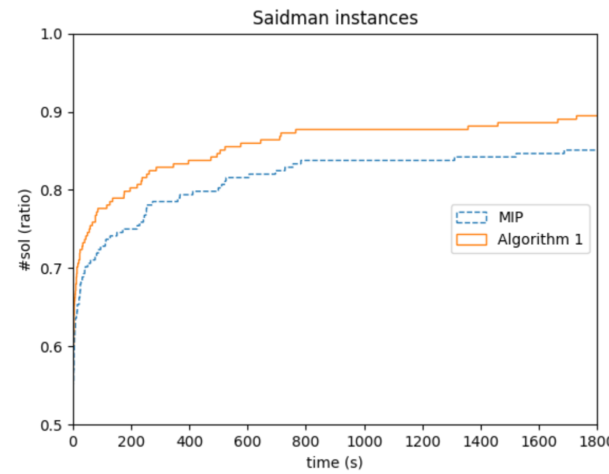
The first important aspect to notice is that Algorithm 1 is able to solve more instances under 30 minutes than the baseline. In total there were 44 instances that were not solved by the baseline, compared to 25 that were not solved by Algorithm 1. Although both algorithms offer similar performances on sparse graphs, with Algorithm 1 performing slightly worse than the base line, there is a significant improvement on larger instances. We can see that the number of instances solved augments for heterogeneous instances with 50 vertices and 4 or 5 players, and for all Saidman instances with 40 or 50 vertices. Furthermore, the time taken to solve these instances is smaller for Algorithm 1, as it needs to generate fewer constraints. Moreover, as presented in figures 1(a) and 1(b), when we compare the performance profiles of the baseline and Algorithm 1 for heterogeneous and Saidman instances, we see that Algorithm 1 always dominate the baseline for the proportion of instances solved at a given time.

The performance profiles also tells us that the performances of Algorithm 1 are not evenly distributed. Within a few seconds, more than 90 % of the heterogeneous instances and more than 60 % of the Saidman instances are solved. This can be explained by the fact that for many instances, only one iteration is needed, which means that the problem of generating the constraints is skipped altogether.

These results for Algorithm 1 indicate that it is preferable to use it over the baseline. Although the baseline might offer a slightly better performance on small instances, the difference is minor and is offset by the results offered by Algorithm 1 on larger instances. Algorithm 1 sets the stage for further improvements, namely the development of stronger cuts, which might be needed to solve realistically-sized instances.



(a) Heterogeneous instances



(b) Saidman instances

Figure A.1. Performance profiles comparing Algorithm 1 with the baseline

Instances				Solved		Algorithm 1				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Time M.P.	Time Coal.	# Cons.	Iterations	%
20	3	0.75	65	100	0.01	0.00	0.00	0.00	1.00	100
	4	0.35	80	100	0.02	0.00	0.00	0.00	1.00	100
	5	0.80	45	100	0.24	0.00	0.00	0.00	1.00	100
30	3	0.70	70	100	0.01	0.00	0.00	0.00	1.00	100
	4	0.55	65	100	0.09	0.00	0.00	0.00	1.00	100
	5	1.05	50	100	1.27	0.00	0.00	0.00	1.00	100
40	3	1.85	25	100	0.03	0.00	0.00	0.00	1.00	100
	4	2.05	15	100	0.50	0.00	0.00	0.00	1.00	100
	5	1.90	10	100	7.56	0.00	0.00	0.00	1.00	100
50	3	3.00	10	100	0.05	0.00	0.00	0.00	1.00	100
	4	2.55	15	100	1.00	0.00	0.00	0.00	1.00	100
	5	3.10	0	100	20.47	0.00	0.07	0.21	1.05	100

Table A.4. Statistics for Algorithm 1 on sparse graphs.

Instances				Solved		Algorithm 1				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Time M.P.	Time Coal.	# Cons.	Iterations	%
20	3	11.74	0	100	0.02	0.00	0.00	0.26	1.16	100
	4	11.58	0	100	0.04	0.00	0.00	1.00	1.53	100
	5	10.58	0	100	0.08	0.00	0.00	0.11	1.05	100
30	3	27.63	0	100	0.07	0.00	0.01	0.84	1.32	100
	4	25.79	0	100	0.11	0.00	0.01	0.42	1.21	100
	5	26.00	0	100	0.22	0.00	0.00	0.00	1.00	100
40	3	47.79	0	100	0.21	0.00	0.02	0.37	1.16	100
	4	46.26	0	100	0.29	0.00	0.01	0.32	1.16	100
	5	47.63	0	100	0.75	0.00	0.00	0.05	1.05	100
50	3	76.42	0	100	0.46	0.00	0.00	0.11	1.05	100
	4	74.26	0	100	40.18	0.00	0.02	0.26	1.16	100
	5	76.47	0	95	46.13	0.00	0.38	0.83	1.22	100

Table A.5. Statistics for Algorithm 1 on heterogeneous graphs.

Instances				Solved		Algorithm 1				Weak Core
$\|V\|$	n	$\|E\|$	Trivial (%)	%	Time	Time M.P.	Time Coal.	# Cons.	Iterations	%
20	3	10.42	0	100	0.30	0.00	0.28	0.58	1.16	100
	4	11.74	5	100	0.09	0.00	0.04	1.21	1.26	100
	5	9.00	0	100	0.21	0.00	0.00	0.00	1.00	100
30	3	26.47	0	100	38.20	0.00	38.15	1.89	1.53	100
	4	21.53	0	100	15.31	0.01	15.12	10.05	3.05	100
	5	26.84	0	95	15.60	0.01	14.54	15.06	2.89	100
40	3	43.68	0	100	102.96	0.00	102.84	5.95	2.58	100
	4	38.47	0	90	162.13	0.02	161.81	23.17	5.11	100
	5	43.26	0	80	62.54	0.09	60.14	76.93	9.60	100
50	3	66.11	0	80	159.31	0.01	159.11	7.20	2.93	100
	4	63.32	0	65	422.27	0.04	421.72	44.58	7.17	100
	5	52.16	0	65	104.05	0.08	95.19	67.58	8.75	100

Table A.6. Statistics for Algorithm 1 on Saidman graphs.