Optimization models and methods for kidney transplantation programs

Dissertation submitted
in fulfilment of the requirements for the degree
of Ph.D. in Economics and Management

Marie Baratto

Jury:

Yves Crama (Supervisor)
HEC Liège, Management School of the University of Liège

Bernard Fortz
HEC Liège, Management School of the University of Liège

João Pedro Pedroso
University of Porto

Michaël Schyns (Supervisor)
HEC Liège, Management School of the University of Liège

Bart Smeulders
Eindhoven University of Technology

Frits C.R. Spieksma
Eindhoven University of Technology

Acknowledgments

The completion of my thesis would not have been possible without certain people, and I would like to take the opportunity to thank them.

First, I would like to express my sincere gratitude to my thesis supervisor, Professor Yves Crama. I will never forget that moment when we first discussed the possibility of a thesis together, and I have never regretted for a single second in six years having accepted this opportunity a few days later. Writing a thesis is a bloody difficult thing to do, but with your help and support, I loved this first step in my professional career. Over the years, I am very grateful for your availability. You have always accepted my meeting requests and been there to answer my questions. You have encouraged me to look at problems differently or more thoroughly, to question myself, and to embrace new opportunities, which has allowed me to grow both professionally and personally. You have been the supervisor I needed. Thank you from the bottom of my heart for your time, your kindness, and your precious pieces of advice. You have shown me what it means to be a good researcher but also what a great teacher is.

In addition to my supervisor, I would also like to thank the other members of my thesis committee: Michaël Schyns, Bart Smeulders, and Frits Spieksma. Every year, our meetings have been productive and have enabled me to question myself about my work and to receive outside opinions on the progress of this thesis. I would especially like to thank Bart Smeulders and Frits Spieksma for involving me, as a first-year Ph.D. student, as part of one of their projects so that I could work on a real-life problem that led to a chapter of this thesis. In particular, Bart Smeulders for his day-to-day guidance. A special thanks to Michaël Schyns as well for accepting to take up the role of administrative supervisor for the last few months of my Ph.D. journey so that I could defend my dissertation. Next, I would like to thank the two additional jury members who agreed to sit on my jury and judge my work: Bernard Fortz and João Pedro Pedroso. In particular, João Pedro Pedroso who, with Ana Viana, agreed to host me for three months

in Porto for a research stay. This stay abroad wasn't easy at the time, but in retrospect, it was probably one of the best experiences of my Ph.D. journey and gave rise to my favorite chapter of this thesis. To all of you, thank you for your time and kindness over all the years.

I would also like to thank all the extraordinary colleagues I have had the chance to meet over the years. This QuantOM team is really special. Thank you for the few conferences we participated in together and those enjoyable lunchtimes. In particular, I want to thank the few colleagues with whom I was lucky enough to share office 334 for a little or a lot of time. Especially the colleagues who have become true friends: Anne-Sophie, Stéphanie, Célia, Elodie E, Elodie B, Justine. Thank you for the hours spent talking about work and everything else. Thank you for the support when needed. Thank you for those moments of friendship in and outside HEC.

On a personal note, I would also like to thank my close friends, my cousins, my grandfather, my brother and my sister who always tried very hard to understand what I was doing at work, only to say that I was doing a thesis in mathematics... Thank you for even trying to understand my thesis topic, for taking an interest in my work and for your support and kind words during the important stages of my Ph.D. journey. In particular, I would like to thank my life partner, my fiance, my future baby daddy, Romain, for his support over the last few years. I know I have put you in uncomfortable situations, and you have had to endure a lot of my stress. Thank you for being by my side, for celebrating the important steps during this journey with me, and for supporting me, especially during my stay in Porto.

Finally, I would like to thank my parents, who are my best friends and greatest support. I cherish our relationship, and I would like to take this opportunity to express all my love and gratitude. Whether in my education, professional career, or personal life, you have always been a tremendous support. You have been by my side in all my personal and professional projects, helped me grow, helped me question myself when necessary, and were the shoulders I could lean on when needed. You are my role models, and I hope to be as good a parent as you are to me. Thank you from the bottom of my heart.

Contents

List of Abbreviations

CCMC	Cardinality constrained multi-cycle problem
CIF	Cumulative incidence function
EA	Extended arc
ESRD	End-stage renal disease
ET	Eurotransplant
ETKAS	Eurotransplant kidney allocation system
HI	Hitting set problem
HLA	Human leukocyte antigens
ILP	Integer linear programming
KEP	Kidney exchange program
MEA	Modified extended arc
MWCS	Maximum weighted cycle selection problem
NDD	Non-directed donor
PI	Position-indexed
RR-2-KEP	Relaxed restricted two-stage KEP
SEA	Simple extended arc

Chapter 1

Introduction

This introduction aims to present the core chapters of the dissertation and their common topic: kidney transplantation. For a better understanding of the thesis, the last section of the introduction includes a review of fundamental concepts used throughout the chapters. It serves to clarify some theoretical concepts.

1.1 Kidney transplantation programs

Patients with an end-stage renal disease (ESRD) characterized by the loss of both kidneys' function, also known as kidney failure, require a treatment method that replaces the lost function. The two main treatment options available are dialysis and transplantation. Dialysis is a very time-consuming treatment that requires several sessions per week, each lasting several hours. Furthermore, patients undergoing dialysis experience a poor quality of life (Wolfe et al., 1999), and the treatment is expensive (Held et al., 2016). Kidney transplantation is a better treatment option regarding the patient's quality of life and cost for medical institutions. Kidney transplantation can occur in two different settings: the organ to be transplanted can be removed either from a deceased donor or from a living donor; the latter case arises if the patient has a relative or close friend willing to donate a healthy kidney (Rapaport, 1986). This thesis addresses both settings. The next three chapters deal with mathematical models related to living-donor kidney transplantation, whereas Chapter 5 is concerned with deceased donor kidney transplantation.

For a donor and a patient to be medically compatible, several conditions must be met (Ashlagi and Roth, 2021). A first one is that their blood types must be compatible, meaning that a patient can only receive a kidney from a donor who has the same blood antigens (A or B) or no antigen

at all (O). This is referred to as "ABO-compatibility". Additionally, the patient cannot have antibodies to the donor's human leukocyte antigens (HLA) since otherwise, the patient's immune system will try to reject the organ. This is called "tissue-type compatibility". To ensure that the patient and the donor are tissue-type compatible, it is necessary to perform a so-called *crossmatch test*. This can be done *virtually*, based on available data regarding the donor-patient pair, or *physically* by analyzing blood samples from the pair. A negative crossmatch result means that the patient is unlikely to reject the donor's kidney. Physical crossmatch tests are costly and time-consuming, so they are typically only used to verify medical compatibility in cases where transplantation is being seriously considered between a specified pair of individuals. It is important to note that a negative *virtual* crossmatch test is insufficient to perform a transplant; a negative *physical* crossmatch test is always required to carry out a transplant.

When a patient has the opportunity of receiving a transplant from a living donor, it is the preferred treatment since it offers the highest chance of success (Roth et al., 2004). In some cases, patients are unable to receive a kidney from their associated healthy donor because they are not medically compatible, but they can possibly receive a kidney from another compatible donor. Considering sufficiently many incompatible patient-donor pairs makes it potentially feasible to transplant kidneys along cycles of pairs. A 2-cycle, for example, is a cycle involving two patient-donor pairs such that the donor of pair 1 is compatible with the patient of pair 2 and the donor of pair 2 is compatible with the patient of pair 1. This situation can give rise to two matches whereby both pairs exchange kidneys, and both patients benefit from a transplant (Roth et al., 2004). Figure 1.1 illustrates a 2-cycle.

A collection of disjoint cycles of feasible transplants is called an *exchange*. In an exchange, all transplants associated with a cycle are usually performed simultaneously to prevent situations where donors drop out prematurely without donating a kidney after their preferred patients have received a transplant. Due to logistical constraints and limited resources, the maximum length of cycles is therefore typically limited to a small number of transplants, say, two, three, or four. A set of disjoint cycles, each involving at most K patient-donor pairs, is called a *K-way exchange*.

Kidney exchange programs (KEPs) have been set up in many countries (or associations of countries) for the benefit of ESRD patients who have a willing donor with whom they are medically incompatible. By enrolling sufficiently many incompatible patient-donor pairs, KEPs aim to identify exchanges that optimize a desired outcome, such as the number or the quality of the transplants to be performed. We generically refer to this optimization problem as the *KEP problem*. Biró et al. (2021) list 19 possible objectives for the KEP problem. Biró et al. (2019) report the current practices in European KEPs,

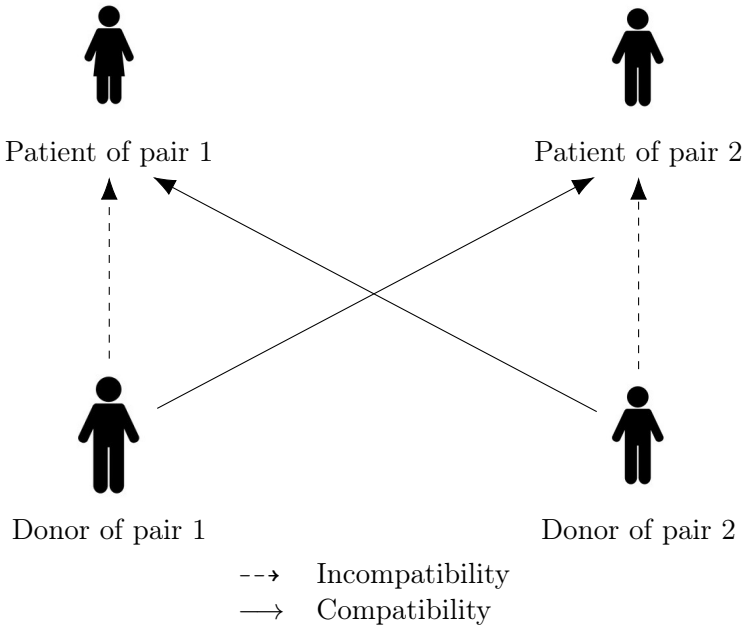


Figure 1.1: Illustration of a 2-cycle

and Ashlagi and Roth (2021) discuss other national and international KEPs worldwide, such as the MSTH program in the United States.

A *non-directed donor* (NDD) (sometimes called altruistic donor) is a person who is willing to donate a kidney to a patient with kidney failure, but who is not initially associated with a specific patient (Constantino et al., 2013). NDDs may be enlisted in KEPs. When this is the case, *chains* of transplants starting with an NDD are allowed to be part of an exchange. The NDD can initiate a sequence of transplants by donating a kidney to a patient in a patient-donor pair, the donor of that pair donates a kidney to another patient, and so forth until the last donor of the chain donates a kidney to a deceased donor program or becomes available as an NDD in the next run of the KEP. This is illustrated in Figure 1.2. Here again, a limit on the maximum chain length is usually imposed usually due to legal regulations. Note that some countries such as France, Belgium, Austria, Sweden, Switzerland, Poland, and Greece, for example, do not accept NDDs in their programs (Ashlagi and Roth, 2021).

Even though this dissertation mostly considers questions related to the optimization of kidney exchange programs, many actual transplants take place in a setting where a kidney is removed from a recently deceased person to be transplanted into an ESRD patient. Eurotransplant is an international nonprofit organization which coordinates the allocation, across several European countries, of deceased donor kidneys to patients enrolled on a waiting list.

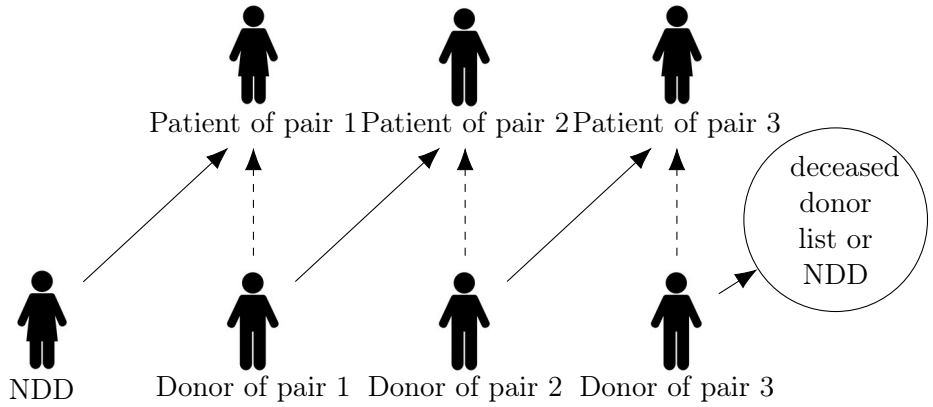


Figure 1.2: Illustration of a chain of length 3

The allocation is made according to a specific mechanism, the so-called *Eurotransplant Kidney Allocation System*, or ETKAS (Mayer and Persijn, 2006), (Eurotransplant, 2023). Some aspects of this mechanism will be discussed in Chapter 5 of the thesis.

1.2 Models and methods for KEP problems

1.2.1 The basic KEP problem

The KEP problem has been extensively studied in the operations research literature where it is usually modeled as a combinatorial optimization problem involving the following ingredients. First, the kidney exchange program is modeled as a loopless *compatibility digraph* $G = (V, A)$ where the set of vertices V represents the set of incompatible patient-donor pairs, and the set of arcs A expresses medical compatibility between each donor and each patient; that is, arc $(i, j) \in A$ if the donor of pair i is compatible with the patient of pair j. Since crossmatch tests are costly and time-consuming, the compatibility digraph is typically based on ABO-compatibility and on a virtual crossmatch based on the available data regarding the donor HLA and the patient antibodies. Crossmatch tests can be performed subsequently to verify tissue-type compatibility once promising matches have been identified. In some cases, a weight $w_{i,j}$ is assigned to each arc $(i, j) \in A$ to represent the quality or the utility of a transplant from the donor of pair i to the patient of pair j.

A (directed) *cycle* of a digraph G is a sequence $c = (v_1, a_1, v_2, a_2, \ldots, v_k, a_k, v_{k+1})$ where $k \geq 2$, $v_1, v_2, \ldots, v_k$ are distinct vertices of G, $v_1 = v_{k+1}$, $a_1, a_2, \ldots, a_k$ are distinct arcs, v_i is the tail of a_i, and v_{i+1} is its head for $i = 1, 2, \ldots, k$. The length of c is k, and we say that c is a k-cycle. When no confusion can

arise, we often identify a cycle with its sequence of arcs $(a_1, a_2, \ldots, a_k)$, or its sequence of vertices $(v_1, v_2, \ldots, v_k)$.

A *K-way exchange* is a set of pairwise vertex-disjoint cycles, each of length at most K. When the arc (i, j) is contained in one of the cycles of an exchange, the donor of pair i and the patient of pair j are often said to be *matched*. The exchange itself is sometimes called a *matching*.

As mentioned earlier, the KEP problem aims to find a K-way exchange that optimizes a given objective function. Much of the literature on the KEP problem considers the natural objective of maximizing the number of transplants, that is, the number of vertices (or arcs) contained in an exchange. Unless otherwise stated, we simply refer to this basic variant as "the KEP problem" in the remainder of this introductory chapter.

When $K = 2$ or $K = n$, the KEP problem reduces to a weighted matching problem and can be solved in polynomial time (Roth et al., 2005). However, the problem is NP-hard for any fixed value of $K \geq 3$ (Abraham et al., 2007). This NP-hardness result has prompted much research on integer programming formulations and on solution methods during the last two decades.

Roth et al. (2007) and Abraham et al. (2007) have introduced the first two integer programming formulations of the KEP problem, namely, the *cycle formulation* and the *edge formulation*. Both formulations are exponential. More precisely, the cycle formulation has an exponential number of variables, but this number is $O(|V|^K)$ which is polynomial for fixed cycle length K (remember that in practice, K is usually set to 3 or 4). The edge formulation has an exponential number of constraints, again $O(|V|^K)$. Constantino et al. (2013) have later proposed a compact extended formulation of the edge formulation, namely, *the extended edge formulation*. Then, another compact extension was proposed by Dickerson et al. (2016), namely, the *position-indexed edge formulation*. Compact formulations are polynomial in size, meaning that they only involve a polynomial number of variables and constraints. Hence, they can be solved without the need for column or row generation. We refer to Mak-Hau (2017) for a comprehensive literature review of formulations and solution methods up to that date. More recently, Delorme et al. (2023b) described a new exponential formulation and an associated solution method, the *half-cycle formulation*.

Since some formulations proposed in this dissertation are inspired by earlier formulations of the KEP problem, the first four ones cited above will be briefly recalled in Section 1.4.1.

In terms of tightness, it can be shown (Abraham et al., 2007; Constantino et al., 2013) that the cycle formulation has a tighter linear relaxation than the edge and the extended edge formulations. Dickerson et al. (2016) proved

that the position-indexed edge formulation yields the same linear relaxation bound as the cycle formulation. In practice, the efficiency of formulations largely depends on the value of K and on the size of the instances. The cycle formulation and position-indexed edge formulation are considered to be the most efficient ones (Mak-Hau, 2017; Dickerson et al., 2016).

Given the sizes of the formulations, several authors have developed column-generation approaches for the KEP problem. Abraham et al. (2007) proposed the first branch-and-price algorithm. Lam and Mak-Hau (2020), based on the work of Klimentova et al. (2014), proposed a branch-and-cut-and-price algorithm. Recent work focused on a branch-and-price algorithm which considers chains in addition to cycles, see, e.g., Pansart et al. (2022).

Different solution approaches are handled in this dissertation depending on the problem under study. For compact formulations and for formulations with an exponential number of variables, complete formulations are simply provided to a standard commercial solver; this remains tractable for the instance sizes under consideration. On the other hand, for formulations with an exponential number of constraints, branch-and-cut procedures are implemented. Details regarding the various approaches will be given in the chapters where they are developed.

As a last note on the basic KEP problem, it should be mentioned that chains of kidney exchanges are usually considered separately from cycles in the mathematical formulations and in the optimization methods proposed in the literature (Mak-Hau, 2017). However, chains can also be considered in the compatibility digraph model by adding a vertex for each NDD and by adding a dummy arc between each patient-donor pair and each NDD: in this way, a chain of length ℓ corresponds to a cycle of length $\ell + 1$ in the augmented digraph. Clearly, a main limitation of this modeling option is that it does not allow fixing a maximum chain length ℓ independently of the maximum cycle length K. In spite of this limitation, chains will not be explicitly handled in this dissertation and the focus will be exclusively placed on exchanges consisting of cycles.

1.2.2 Variants of the KEP problem

Except for Chapter 5 which is dedicated to deceased donor transplantation programs, the core of the thesis explores questions motivated by the consideration of different aspects of the KEP problem which have been recently investigated in the literature. As an introduction to these chapters, we next briefly discuss two such variants of the basic KEP problem.

Recourse

As we mentioned earlier, the physical compatibility between a donor and a patient is only crossmatch-tested after an intended transplant has been identified. Therefore, some authors took an interest in the situation where a donor and a patient are found to be incompatible *after* the computation of a best possible exchange. In such a configuration, not only the transplantation between these particular donor and patient fails, but also the whole cycle in which the transplantation is involved. This is an important issue; for example, Dickerson et al. (2018) report that in some KEPs, 93% of the identified transplantations fail due partly to results of crossmatch tests (but also for other reasons, e.g., because a patient has already received a kidney from another program or is too ill to undergo surgery).

One way to tackle this issue in mathematical models is to assess, for each arc $(i, j) \in A$, the probability $p_{i,j}$ that a crossmatch test would confirm donor i and patient j to be compatible. This gives rise to stochastic versions of the KEP optimization problem. Dickerson et al. (2018), Klimentova et al. (2016), Pedroso (2014), Smeulders et al. (2022), among others, considered such models. We refer to Smeulders et al. (2022) for an overview of different approaches.

A specific model proposed in Smeulders et al. (2022) assumes that a subset of arcs (i.e., potential transplants) can be selected in a first stage to be explicitly crossmatch-tested, before an optimal selection of cycles is computed in a second stage on the subset of arcs that successfully passed the crossmatch tests. From a mathematical perspective, this leads to a two-stage stochastic programming problem that Smeulders et al. (2022) define as follows: given a compatibility digraph $G = (V, A)$ and a testing budget b, identify (in stage 1) a subset of arcs $B \subseteq A$ with $|B| \leq b$ such that the expected number of transplants in the graph $G_B = (V, B)$ (in stage 2) is maximized. They propose several integer programming formulations for this two-stage stochastic KEP problem and different solution methods associated with each formulation. Furthermore, as the formulations are not solved efficiently, they tighten one formulation of the problem by imposing constraints on the set of arcs B to be tested. These constraints ensure that arcs not contained in any cycle should not be selected to be tested: indeed, such arcs cannot be used in an exchange.

This research motivated our interest to investigate the following concept: for a directed graph $G = (V, A)$, a subset of arcs $B \subseteq A$ is called a *cycle selection* if each arc of B is contained in a directed cycle of $G_B = (V, B)$. As an illustration, Figure 1.4 displays the collection of cycle selections of the digraph represented in Figure 1.3.

A better understanding of cycle selections may potentially improve the so-

lution of the two-stage stochastic KEP problem. Chapter 2 studies some properties of cycle selections, while Chapter 3 presents the results of computational experiments on related problems, including the application of our findings to the two-stage stochastic KEP problem of Smeulders et al. (2022).

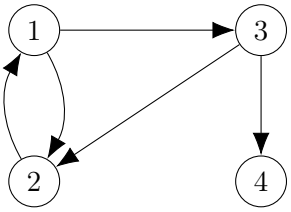


Figure 1.3: A directed graph.

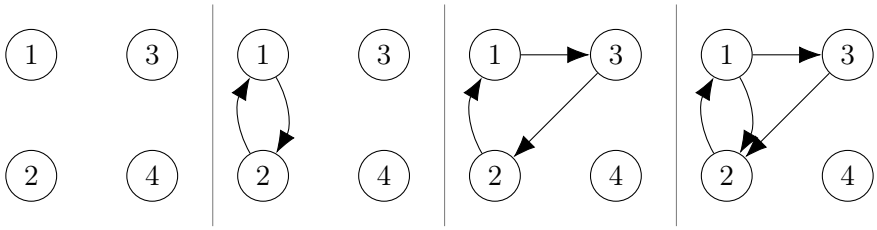


Figure 1.4: All cycle selections of the directed graph in Figure 1.3.

Preferences

A second variant of the KEP problem studied in this dissertation investigates the situation where each patient has a preference ranking over their set of compatible donors. Indeed, from the point of view of the patients or, rather, their respective medical teams, not all donors' kidneys are equal. Some kidneys are preferred over others because they have a higher chance of leading to successful transplants and longer survival. (For simplicity, we will say that a patient prefers a donor rather than saying that the medical team prefers a kidney from this donor.)

This setup leads to the following mathematical model: given the compatibility digraph $G = (V, A)$, we assume that each pair $i \in V$ has expressed preferences over their set of compatible donors or equivalently, over their set of *in-neighbors* $N_G^-(i) = \{j \in V : (j, i) \in A\}$. For example, in Figure 1.5 hereunder, the patient of pair 2 prefers the donor of pair 4 to the donor of pair 1.

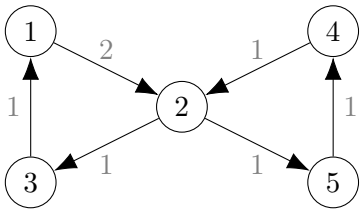


Figure 1.5: A small digraph with preferences on the arcs

Matching problems involving preferences arise in different economic contexts, often remote from kidney exchange programs. As such, they have been extensively studied in the operations research and economic literature, especially after the pioneering work on the stable marriage problem by Gale and Shapley (1962). A fundamental concept in this literature is that of a *stable exchange*. In the context of KEPs, a stable exchange can be informally defined as an exchange such that no subset of patients have the possibility to defect in order to benefit from a more attractive exchange. A more precise definition of such a situation will be given in Chapter 4. (The 2012 Nobel Memorial Prize in Economic Science was jointly awarded to Lloyd Shapley and to Alvin E. Roth for their contributions to "the theory of stable allocations and the practice of market design", including work on kidney exchange programs.) A central issue is then to compute a maximum stable exchange or to prove that none exists. Integer programming formulations describing stable K-way exchanges for the KEP problem have been recently proposed in the literature (Klimentova et al., 2023). Chapter 4 investigates an alternative, weaker concept of *local stability* for K-way exchanges.

1.3 Structure and contributions of the dissertation

The core of the dissertation consists of four chapters treating three different topics related to the optimization of kidney transplantation programs. The chapters are organized as follows: Chapter 2 and Chapter 3 investigate the cycle selection problem and its application to the KEP problem. Chapter 4 focuses on locally stable exchanges, their properties and their computation. Finally, Chapter 5 discusses some aspects of the simulation of mechanisms allocating deceased donor kidneys to patients enrolled on a waiting list.

In **Chapter 2**, we introduce the cycle selections of a directed graph and we define a related optimization problem: the maximum weighted cycle selection problem (**MWCS**). We prove that **MWCS** is strongly NP-hard and we provide six integer linear programming (ILP) formulations of the problem. One is an arc-based formulation, called the *arc formulation*, featuring an exponential number of constraints that can be separated in polynomial time. The next four formulations are arc-based compact formulations, respectively

called the *simple extended arc formulation*, the *extended arc formulation*, the *modified extended arc formulation*, and the *position-indexed formulation*. The final formulation is an extended non-compact cycle-based formulation, simply called the *cycle formulation*. We investigate the relative strength of these formulations and prove that the linear relaxation of the arc formulation, the projection of the linear relaxation of the extended arc formulation and the projection of the linear relaxation of the modified extended arc formulation are identical and are included in the projection of the linear relaxation of the other three formulations. Next, we focus on the arc formulation and on the description of the associated *cycle selection polytope* for complete digraphs. We prove that this polytope is full-dimensional and that all the inequalities used in the arc formulation are facet-defining. Furthermore, we describe three new classes of facet-defining inequalities and a class of valid inequalities. In the last section of the chapter, we investigate the cycle selection problem with a maximum cycle length set to $K = 3$. We provide an arc-based formulation with an exponential number of constraints that can be separated in polynomial time. We also prove that the associated polytope is full-dimensional for complete digraphs and that all inequalities describing the formulation are facet-defining. Except for its last section, Chapter 2 has been published as an article in *Discrete Applied Mathematics* (Baratto and Crama, 2023).

In **Chapter 3**, we conduct numerical experiments to test the efficiency of the integer linear programming formulations of the cycle selection problem. Specifically, we carry out experiments to identify a maximum weighted cycle selection in randomly generated digraphs, where each arc is assigned a random weight. The results show that these instances are relatively easy, and that the arc formulation and the simple extended arc formulation outperform the other four ILP formulations in terms of total running time. We also investigate some variants of the problem by adding a budget constraint and/or a maximum cycle length constraint to the models. These variants are more challenging to solve, especially when the model has a budget constraint. In the second part of the chapter, we apply our knowledge about cycle selection formulations and the previous numerical results to the two-stage stochastic KEP problem introduced in Smeulders et al. (2022). In particular, we propose alternative ways to solve this problem using different implementations of the Benders decomposition procedure and we explore several possibilities to enhance the optimization process. Although we could not improve the running time of the original formulations proposed in Smeulders et al. (2022), one of our new formulations has a similar performance.

Chapter 4 focuses on KEP problems where each patient has expressed a preference ranking over their set of compatible donors, that is, over the set of their in-neighbors in the compatibility digraph. We extend recent work

on stable exchanges (Klimentova et al., 2023) by introducing and underlining the relevance of a new concept of *locally stable exchanges*. We show that the locally stable exchanges of a compatibility digraph are exactly the so-called local kernels of an associated blocking digraph whereas the stable exchanges are the kernels of the blocking digraph. We also prove that finding a nonempty local kernel in an arbitrary digraph is NP-complete. We then propose four integer programming formulations for computing a locally stable exchange of maximum size. We conduct numerical experiments to assess the quality of our formulations and to compare the size of maximum locally stable exchanges with the size of maximum stable exchanges. The numerical experiments show that nonempty locally stable exchanges frequently exist in digraphs that do not have any stable exchange. All the previous results and observations hold true even when the concept of (locally) stable exchanges extends to (locally) strongly stable exchanges. Chapter 4 has been submitted to the *European Journal of Operational Research* and is currently under review.

Finally, **Chapter 5** deals with kidney transplantations from deceased donors. It is based on work carried out in the framework of a research project initiated by Eurotransplant, a European program that allocates deceased donor organs to patients placed on a waiting list. The project was supervised by Bart Smeulders and Frits Spieksma. It aimed to simulate and to numerically compare the possible outcomes of a mechanism allocating deceased donor kidneys under alternative allocation rules. Our contribution, as described in this chapter, relates to one specific modeling aspect of this project. Namely, when given the actual historical data recorded by Eurotransplant during a period of time, some data required to conduct the simulation was missing: for those patients who had been transplanted under the current mechanism (called EKTAS), the total time during which they would have remained in an appropriate medical condition to be transplanted (had they not been transplanted already) was unknown. Indeed, in the absence of a transplant, those patients would have eventually left the waiting list due to their medical condition or death. Chapter 5 explains the approach used to simulate these missing data using survival analysis and the Kaplan-Meier method. While the principal advantage of the Kaplan-Meier method is its simplicity and ease of computation, its main weakness stems from an assumption concerning the independence of certain random events which can hardly be verified in practice. We provide some insights into the validity of the method.

1.4 A short refresher

In order to keep the different parts of the dissertation self-contained and independent of each other, to the largest possible extent, some of the defi-

nitions and of the material presented in this Introduction will be repeated in other chapters. This is especially true for those chapters which have been either published or submitted for publication in journals.

On the other hand, and in contrast to the previous remark, some problem formulations and methods are used without much prior explanation throughout the dissertation. The purpose of the current section is to recall to the readers a few facts about formulations of the KEP problem, about polyhedral theory, and about the Benders decomposition procedure, with the hope to facilitate the understanding of subsequent chapters.

1.4.1 Formulations of the KEP problem

Over the last two decades, several ILP formulations of the KEP problem have been proposed in the literature. Since some formulations proposed in this dissertation are inspired by this earlier work, four formulations of the KEP problem are recalled below, namely: the *cycle formulation* (Roth et al., 2007; Abraham et al., 2007); the *edge formulation* (Roth et al., 2007; Abraham et al., 2007); the *extended edge formulation* (Constantino et al., 2013); and the *position indexed edge formulation* (Dickerson et al., 2016). The first two formulations are exponential in size, whereas the next two ones are compact.

All formulations refer to the KEP problem associated with a compatibility digraph $G = (V, A)$, where the set of incompatible patient-donor pairs is represented as $V = \{1, \ldots, n\}$. For the sake of simplicity, we assume that the objective of the KEP problem is to identify a K-way exchange, for K fixed, which maximizes the total number of transplants.

Cycle formulation

To write the cycle formulation (Roth et al., 2007; Abraham et al., 2007), let

- $\mathcal{C}_K(G)$ be the set of cycles of G with length less than or equal to K;
- $V(c)$ be the set of vertices of cycle c, for all $c \in \mathcal{C}_K(G)$;
- $\diamond$ z_c be a binary variable such that $z_c = 1$ if cycle c is selected in the exchange, and $z_c = 0$ otherwise, for all $c \in \mathcal{C}_K(G)$.

Then, the cycle formulation of the KEP problem is as follows:

$$\max \sum_{c \in \mathcal{C}_K(G)} |V(c)| \, z_c \tag{1.1}$$

$$\text{s.t.} \sum_{c : i \in V(c)} z_c \leq 1 \qquad \forall i \in V \tag{1.2}$$

$$z_c \in \{0, 1\} \qquad \forall c \in \mathcal{C}_K(G). \tag{1.3}$$

Constraints (1.2) ensure that for each vertex i of the digraph, at most one cycle of $C_K(G)$ containing i is selected in the exchange. In the context of KEP, these constraints ensure that each patient-donor pair only participates in one exchange, preventing a donor from donating more than one kidney.

Edge formulation

To state the edge formulation of Roth et al. (2007), Abraham et al. (2007), we define a *minimal cardinality violation path* to be a directed path involving $K+1$ distinct vertices of G, i.e., a sequence $\pi = (v_1, a_1, v_2, a_2, \ldots, v_K, a_K, v_{K+1})$ where $v_1, v_2, \ldots, v_{K+1}$ are distinct vertices of G, $a_1, a_2, \ldots, a_K$ are distinct arcs, v_i is the tail of a_i, and v_{i+1} is its head for $i = 1, 2, \ldots, K$. By a slight abuse of notations, we write $(i,j) \in \pi$ if (i,j) is one of the arcs of π. Let

- Π_K be the set of all minimal cardinality violation paths of G;
- $\diamond$ $x_{i,j}$ be a binary variable such that $x_{i,j} = 1$ if arc (i,j) is contained in the exchange and $x_{i,j} = 0$ otherwise, for all $(i,j) \in A$.

Then, the edge formulation of the KEP problem is defined as:

$$\max \sum_{(i,j)\in A} x_{i,j} \tag{1.4}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} x_{i,j} = \sum_{j:(j,i)\in A} x_{j,i} \qquad \forall i \in V \tag{1.5}$$

$$\sum_{j:(i,j)\in A} x_{i,j} \leq 1 \qquad \forall i \in V \tag{1.6}$$

$$\sum_{(i,j)\in \pi} x_{i,j} \leq K-1 \qquad \forall \pi \in \Pi_K \tag{1.7}$$

$$x_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{1.8}$$

Constraints (1.5) ensure flow balance at each vertex, meaning that if an arc enters a vertex i, then another arc must leave i. In the context of KEP, if the patient of pair i receives a kidney then the donor of pair i must donate a kidney. Constraints (1.6) ensure that for each vertex i, at most one arc leaving vertex i can be selected in the exchange; that is, the donor of pair i gives at most one kidney. Lastly, constraints (1.7) express that for each minimal infeasible path $\pi \in \Pi_K$, no more than $K-1$ arcs of the path can be included in an exchange; these constraints impose that each cycle in the exchange has length at most K.

Extended edge formulation

The idea behind the extended edge formulation of Constantino et al. (2013) is to consider multiple copies of the original compatibility digraph G. The

formulation ensures that at most K arcs can be selected in each copy. As the number of cycles in the exchange cannot exceed the number of vertices of G, $|V|$ copies are considered.

Let

- $G^l = (V^l, A^l)$ be a copy of G, with $V^l = V = \{1, \ldots, n\}$ and $A^l = A$ for all $l \in V$;
- $\diamond$ $x^l_{i,j}$ be a binary variable such that $x^l_{i,j} = 1$ if arc (i, j) is selected in A^l to be part of the exchange and $x^l_{i,j} = 0$ otherwise, for all $(i, j) \in A$, for all $l \in V$.

Then, the extended edge formulation is defined as:

$$\max \sum_{l \in V} \sum_{(i,j) \in A} x^l_{i,j} \tag{1.9}$$

$$\text{s.t.} \sum_{j:(i,j) \in A} x^l_{i,j} = \sum_{j:(j,i) \in A} x^l_{j,i} \qquad \forall i \in V,\ \forall l \in V \tag{1.10}$$

$$\sum_{l \in V} \sum_{j:(i,j) \in A} x^l_{i,j} \leq 1 \qquad \forall i \in V \tag{1.11}$$

$$\sum_{(i,j) \in A} x^l_{i,j} \leq K \qquad \forall l \in V \tag{1.12}$$

$$\sum_{j:(i,j) \in A} x^l_{i,j} \leq \sum_{j:(l,j) \in A} x^l_{l,j} \qquad \forall l \in V\ \forall i > l \tag{1.13}$$

$$\sum_{j:(i,j) \in A} x^l_{i,j} = 0 \qquad \forall l \in V,\ \forall i < l \tag{1.14}$$

$$x^l_{i,j} \in \{0, 1\} \qquad \forall (i, j) \in A,\ \forall l \in V. \tag{1.15}$$

Constraints (1.10) ensure the flow balance at each vertex in each copy of G. Constraints (1.11) ensure that each vertex is used at most once. Constraints (1.12) imply that cycles of length more than K cannot be selected in any copy G^l. Finally, constraints (1.13) and (1.14) are not really necessary, but they eliminate symmetries induced in the ILP model by permutations of cycle indices. They express that if any vertex i is the endpoint of an arc selected in cycle G^l, then $i \geq l$ and l also is the endpoint of an arc selected in G^l.

Position-indexed edge formulation

The position-indexed edge formulation (Dickerson et al., 2016) is a natural extension of the extended edge formulation. It indexes the variables by using copies of the compatibility digraph G as well as the positions of the arcs in a cycle. Let

- $G^l = (V^l, A^l)$ with $V^l = \{l, \cdots, n\}$ and $A^l = \{(i,j) \in A : i \in V^l, j \in V^l\}$ for all $l \in V$;
- $\kappa(i,j,l)$ be the set of positions at which arc (i,j) is allowed in a cycle in copy G^l. For $i,j,l \in V$ such that $(i,j) \in A^l$, $\kappa(i,j,l) = \begin{cases} \{1\} \text{ if } i = l, \\ \{2,...,K\} \text{ if } j = l, \\ \{2,...,K-1\} \text{ if } i,j > l; \end{cases}$
- $\diamond$ $x^l_{i,j,k}$ be a binary variable such that $x^l_{i,j} = 1$ if arc (i,j) is selected at position k in a cycle in copy G^l, for all $i,j,l \in V$ such that $(i,j) \in A^l$ and $k \in \kappa(i,j,l)$.

The position-indexed edge formulation is defined as:

$$\max \sum_{l \in V} \sum_{(i,j) \in A^l} \sum_{k \in \kappa(i,j,l)} x^l_{i,j,k} \tag{1.16}$$

$$\text{s.t.} \sum_{j:(j,i) \in A^l \wedge k \in \kappa(j,i,l)} x^l_{j,i,k} = \sum_{j:(i,j) \in A^l \wedge k+1 \in \kappa(i,j,l)} x^l_{i,j,k+1} \quad \begin{array}{l} \forall l \in V, \\ \forall i \in \{l+1, \cdots, n\} \\ \forall k \in \{1, \cdots, K-1\} \end{array} \tag{1.17}$$

$$\sum_{l \in V} \sum_{j:(j,i) \in A^l} \sum_{k \in \kappa(j,i,l)} x^l_{j,i,k} \leq 1 \quad \forall i \in V \tag{1.18}$$

$$x^l_{i,j,k} \in \{0,1\} \quad \begin{array}{l} \forall l \in V, \\ \forall (i,j) \in A^l, \\ \forall k \in \kappa(i,j,l). \end{array} \tag{1.19}$$

Constraints (1.17) ensure the flow balance at each vertex and constraints (1.18) ensure that each vertex is selected at most once. The maximum cycle length is induced by the allowed positions of each arc in each copy l.

1.4.2 Polyhedral theory

In Chapter 2, we perform a polyhedral study of the set of feasible solutions of an optimization problem associated with cycle selections. The present section aims to recall a few basic concepts to understand the purpose of the work done in that chapter. The definitions and results presented in this section are from the book Nemhauser and Wolsey (1999).

An integer linear programming problem (ILP) is an optimization problem of the form

$$\max\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\} \tag{1.20}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Given an ILP (1.20), its set of feasible solutions (or feasible set) is:

$$S := \{x \in \mathbb{Z}^n : Ax \leq b\}$$

and so (1.20) can be rewritten as

$$\max\{c^T x : x \in S\}.$$

Definition 1. The **linear relaxation** of the set $S := \{x \in \mathbb{Z}^n : Ax \leq b\}$ is the set $\{x \in \mathbb{R}^n : Ax \leq b\}$.

Definition 2. Given a set $Q \subset \mathbb{R}^n$, a point $x \in \mathbb{R}^n$ is a **convex combination** of points of Q if there exists a finite set $\{x^1, x^2, \cdots, x^t\} \subseteq Q$ and $\lambda \in \mathbb{R}^t_+$ such that $x = \sum_{i=1}^t \lambda_i x^i$ and $\sum_{i=1}^t \lambda_i = 1$. The **convex hull** of Q, denoted by $conv(Q)$, is the set of all convex combinations of points in Q.

We are interested in studying and describing the convex hull of the feasible set of an ILP because of the following result:

Theorem 1. $\max\{c^T x : x \in S\} = \max\{c^T x : x \in conv(S)\}$.

Note that if $S = \{x \in \mathbb{Z}^n : Ax \leq b\}$, then its convex hull $conv(S)$ is included in its linear relaxation, and the inclusion is generally strict. Generally speaking, it is difficult to find a complete description of the convex hull of the set S. In fact, it is already NP-hard to decide whether a point $x \in \mathbb{R}^n$ is in $conv(S)$ or not. Still, from an algorithmic point of view, a partial description of the convex hull can help with the solution of the ILP problem.

Let us next recall a few definitions used in polyhedral studies.

Definition 3. A **polyhedron** P is a subset of $\mathbb{R}^n$ that satisfies a finite number of linear inequalities. More precisely, a polyhedron is a set of the form

$$P := \{x \in \mathbb{R}^n : Ax \leq b\}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. A bounded polyhedron is called a **polytope**.

Definition 4. If the maximum number of affinely independent vectors in a polyhedron P is $k + 1$, then we say that the **dimension** of P, denoted $dim(P)$, is k. If $P \subseteq \mathbb{R}^n$ and $dim(P) = n$, then P is said to be of maximal dimension or **full-dimensional**.

Definition 5. Let $P \subseteq \mathbb{R}^n$ be a polyhedron, $w \in \mathbb{R}^n$ and $t \in \mathbb{R}$. The inequality $w^T x \leq t$ is **valid** for P if $P \subseteq \{x \in \mathbb{R}^n : w^T x \leq t\}$, that is, if the points of P satisfy the inequality $w^T x \leq t$.

Definition 6. $F \subseteq P$ is a **face** of P if there exists an inequality $w^T x \leq t$ which is valid for P and such that

$$F = \{x \in P : w^T x = t\}.$$

Definition 7. A **facet** of P if a face of dimension $dim(P) - 1$. If $w^T x \leq t$

is a valid inequality for P such that $F = \{x \in P : w^T x = t\}$ is a facet of P, then we say that the inequality is **facet-defining** for P.

Informally speaking, it can be shown that all facet-defining inequalities are necessary and sufficient to describe a full-dimensional polyhedron.

In Chapter 2, we aim to identify facet-defining inequalities for the convex hull of the feasible solutions of the so-called arc formulation of the cycle selection problem. The following theorem can be used to demonstrate that an inequality is a facet of a given polyhedron.

Theorem 2. *Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a full-dimensional polyhedron and let $F = \{x \in P : w^T x = t\}$ be a face of P. Then the following statements are equivalent:*

- *F is a facet of P;*
- *if $c^T x = \gamma$ for all $x \in F$, then $\begin{pmatrix} c \\ \gamma \end{pmatrix}$ is a multiple of $\begin{pmatrix} w \\ t \end{pmatrix}$.*

The following definition will be needed in the next section (Wolsey, 2020):

Definition 8. For a nonempty polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, $r \in \mathbb{R}^n$ is a **ray** of P if $r \neq 0$ and if $x \in P$ implies $x + \mu r \in P$ for all $\mu \in \mathbb{R}_+$. Moreover, r is an **extreme ray** of P if r cannot be written in the form $r = \mu_1 r^1 + \mu_2 r^2$ where $\mu_1 > 0$, $\mu_2 > 0$, and r^1, r^2 are two distinct rays of P.

1.4.3 Benders decomposition

In Chapter 3, the Benders decomposition procedure is used. This section recalls the general principles of the procedure. It is based on Wolsey (2020).

Consider a mixed integer linear programming problem (MILP) :

$$\begin{aligned} \max\ & cx + hy \\ \text{subject to}\ & Ax + Gy \leq b \\ & x \in X \subseteq \mathbb{Z}^n_+ \\ & y \in \mathbb{R}^p_+. \end{aligned}$$

The main idea behind the Benders decomposition of (MILP) is to solve it by iteratively solving two simpler problems, namely, a master problem (M):

$$\begin{aligned} \max\ & cx + \theta(x) \\ \text{subject to}\ & x \in X \subseteq \mathbb{Z}^n_+ \end{aligned}$$

and a slave problem (B) where x is assumed to be fixed:

$$\begin{aligned} \theta(x) = \max\ & hy \\ \text{subject to } & Gy \leq b - Ax \\ & y \in \mathbb{R}^p_+. \end{aligned}$$

Concretely, the Benders optimization procedure goes as follows. Consider the following problem (M'):

$$\begin{aligned} \max\ & cx + \eta & & \\ \text{subject to } & v^t(b - Ax) \geq 0 & & \text{for all } t \in T \\ & u^s(b - Ax) \geq \eta & & \text{for all } s \in S \\ & x \in X \subseteq \mathbb{Z}^n_+ & & \\ & \eta \in \mathbb{R} & & \end{aligned}$$

In this formulation, u_s, $s \in S$, are solutions of the dual problem of (B), and v_t, $t \in T$, are extreme rays of the dual problem of (B). For any choice of S and T, (M') is a relaxation of the (MILP) problem. (M') is exactly equivalent to (MILP) when S and T contain all the dual solutions and all the extreme rays, respectively.

For the initialization phase, set $T = \emptyset$, $S = \emptyset$, and place (M') on the list of open nodes. (We voluntarily simplify this step to avoid unnecessary theory that won't be used later in the dissertation.) During the course of the procedure, we call *incumbent* the best value of a feasible solution of (M') found so far. Initially, the incumbent is set to $-\infty$. Then the following iterations are performed:

1. Select and remove a node from the list of open nodes. Let (N) denote the problem associated with that given node.
2. Solve the linear relaxation of (N). If it is infeasible, close the node and go to step 1.
3. Otherwise, let (x^*, η^*), be the optimal solution of the linear relaxation of (N). If $cx^* + \eta^*$ is smaller than the incumbent, close the node and go to step 1.
4. Let (DB) be the dual of (B) with $x = x^*$:

$$\begin{aligned} \min\ & u\,(b - Ax^*) \\ \text{such that } & u\,G \geq h \\ & u \in \mathbb{R}^m_+ \end{aligned}$$

   Solve (DB).

i. If (DB) is unbounded, find v^t an extreme ray such that $v^t(b - Ax^*) < 0$, and add the following inequality to (N) and to all the problems on the list of open nodes:

$$v^t(b - Ax) \geq 0$$

Such an inequality is called a *feasibility cut.* Set $T = T \cup \{t\}$ and go to step 2.

ii. If (DB) is feasible and u^s is an optimal solution of (DB) such that $u^s(b - Ax^*) < \eta^*$, then add the following inequality to (N) and to all the problems on the list of open nodes:

$$u^s(b - Ax) \geq \eta.$$

Such an inequality is called an *optimality cut.* Set $S = S \cup \{s\}$ and go to step 2.

iii. If (DB) is feasible and u^s is an optimal solution of (DB) such that $u^s(b - Ax^*) = \eta^*$, then all constraints of (MILP) are satisfied.

- If x^* is integer and y^* is the optimal solution of (B) (given by the dual variables of (DB)), then the incumbent is updated, the associated solution (x^*, y^*) is stored, and the node can be closed.
- If x^* is not integer, branch on one of the variables taking a fractional value, add the corresponding two new nodes (i.e., subproblems of (N)) so created to the list of open nodes, close the current node, and go to step 1.

Once the list of open nodes is empty, an optimal solution of (MILP) is the solution (x^*, y^*) associated with the last incumbent value.

Chapter 2

Cycle selections

The content of this chapter is based on joint work with Yves Crama, in particular on the article Baratto and Crama (2023) published in Discrete Applied Mathematics. In addition to the article, Section 2.6 studies a variant of the original cycle selection problem; it has not been submitted for publication.

Contents

2.1 Introduction

2.1.1 Problem definition

This chapter introduces and investigates a combinatorial optimization problem originally motivated by an application to kidney exchange programs. The motivation will be further developed in Section 2.1.2 hereunder but for now, we start with a mathematical definition of the problem.

Our graph-theoretic terminology is standard and follows Bang-Jensen and Gutin (2009). All directed graphs (or digraphs) we consider in this chapter are loopless and have no parallel arcs. For a digraph $G = (V, A)$, we let $|V| = n$ and $|A| = m$. The digraph $G = (V, A)$ is *complete* if A contains all pairs of distinct vertices (i, j), for $i, j \in V$. A *(directed) cycle* of a digraph G is a sequence of the form $C = (v_1, a_1, v_2, a_2, \ldots, v_k, a_k, v_{k+1})$ where $k \geq 2$, $v_1, v_2, \ldots, v_k$ are distinct vertices of G, $v_1 = v_{k+1}$, $a_1, a_2, \ldots, a_k$ are distinct arcs, v_i is the tail of a_i and v_{i+1} is its head for $i = 1, 2, \ldots, k$. The *length* of C is k, and we say that C is a k-cycle. When no confusion can arise, we often identify a cycle with its set of arcs, so that we can speak of a union of cycles, for example.

Let us now introduce a new definition. Given a directed graph $G = (V, A)$, where V is the set of vertices and A is the set of arcs of G, we say that a subset of arcs $B \subseteq A$ is a *cycle selection* in G if the arcs of B form a union (possibly empty) of directed cycles. Equivalently, B is a cycle selection if and only if each arc of B is contained in a directed cycle of $G_B = (V, B)$. And equivalently again, B is a cycle selection of G if and only if each arc of B is contained in a strong (or strongly connected) component of $G_B = (V, B)$: the equivalence follows from the observation that an arc of B, say (i, j), is in a cycle of G_B if and only if i and j are in a same strong component of G_B.

As an illustration, Figure 2.2 displays the collection of selections of the digraph represented in Figure 2.1.

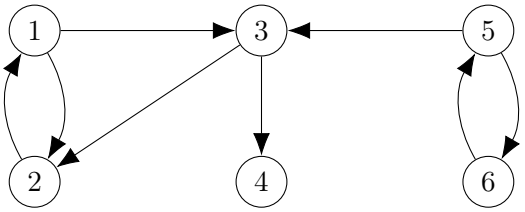


Figure 2.1: A directed graph.

In view of the above definitions, the time complexity to verify that a subset $B \subseteq A$ is a cycle selection is $O(n+m)$, using for example Tarjan's algorithm

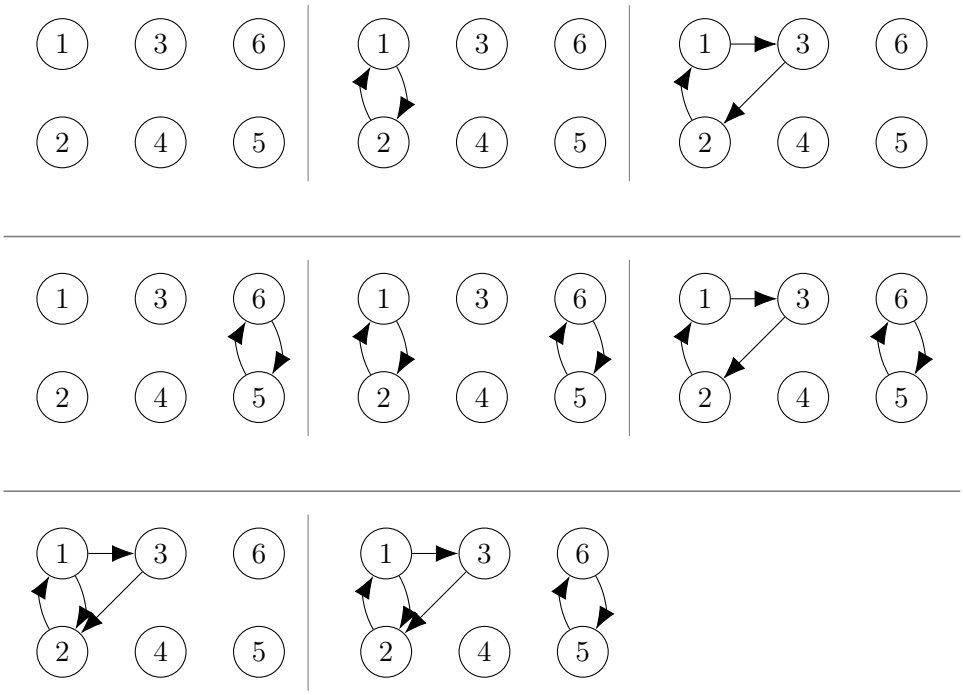


Figure 2.2: All cycle selections of the directed graph in Figure 2.1.

to identify all strong components of G_B (Tarjan (1972)).

The *maximum weighted cycle selection problem* (**MWCS**), or cycle selection problem for short, is the following optimization problem: given a digraph $G = (V, A)$ and a weight $w_{i,j} \in \mathbb{R}$ for each arc $(i, j) \in A$, find a cycle selection B which maximizes $w(B) = \sum_{(i,j)\in B} w_{i,j}$.

This chapter investigates several properties of the cycle selection problem. Section 2.1.2 lays out the motivation for studying it. Section 2.1.3 provides a literature review of previous related work in order to contextualize the cycle selection problem and to position our contributions. Section 2.2 discusses the complexity of the problem. Next, various integer linear programming formulations of the cycle selection problem are proposed in Section 2.3, namely, an arc-based formulation (Section 2.3.1), several extended compact formulations (Section 2.3.2), and an extended non compact one (Section 2.3.3). We establish the relative strength of the linear relaxations of these formulations. Section 2.4 investigates the facial structure of the cycle selection polytope associated with the arc formulation for a complete digraph. We prove that the polytope is full-dimensional and that all the inequalities used in the ILP formulation are facet-defining. Furthermore, we describe three additional classes of facet-defining inequalities and one class of valid inequalities. Sec-

tion 2.5 considers the extension of the cycle selection problem which arises when a constraint is placed on the cardinality of the selection and of the cycles that it includes. Section 2.6 investigates the cycle selection problem with a maximum cycle length set to $K = 3$. In particular, we provide an arc-based formulation and prove that the associated polytope for complete digraphs is full-dimensional and that all constraints describing the formulation are facet-defining. Finally, Section 2.7 presents some conclusions and perspectives for future research.

2.1.2 Motivation

Our motivation to study cycle selections originally stems from optimization problems arising in the context of kidney exchange programs (KEPs). Let us briefly explain how KEPs work. Nowadays, the preferred treatment option offered to patients with an end-stage renal disease is kidney transplant from a living donor. This option exists primarily when a patient has a relative willing to donate one of their healthy kidneys. However, in many situations, patients are unable to receive a kidney from their associated healthy donor because of ABO blood type incompatibility or tissue type incompatibility. Kidney exchange programs try to alleviate this difficulty by enlisting a large number of incompatible patient-donor pairs, say, pairs (P_i, D_i) made up of patient P_i and donor D_i, for $i = 1, \ldots, n$. Considering such a pool makes it potentially feasible to transplant kidneys in cyclic fashion with, for example, D_1 donating a kidney to P_2, D_2 donating one to P_3, and D_3 donating one to P_1 (Roth et al. (2004)).

Given a pool of patients, one can build a *compatibility digraph* $G = (V, A)$ whose vertices are the pairs $v_i = (P_i, D_i)$, and A contains the arc (v_i, v_j) if D_i appears to be compatible with P_j, based on blood and tissue type. Maximizing the number of feasible cyclic transplants amounts now to finding in G a collection of *vertex-disjoint* cycles whose union contains as many arcs as possible. (Beside cycles, some KEPs may also take non-closed directed paths in consideration, but we disregard this option here.) There is a large amount of literature documenting various formulations and matching algorithms to solve this optimization problem; see, e.g., Constantino et al. (2013), Dickerson et al. (2016), Biró et al. (2021) and the literature review in Section 2.1.3.

One of the remaining issues with this approach, is that in reality, blood type and tissue type are not the only determinants of the feasibility of a transplant. The decision to perform a transplant is based on more complex, so-called crossmatch tests of compatibility between donor and patient. In practice, for cost- and time-efficiency reasons, crossmatch tests are only performed *after* an intended transplant has been identified.

As a result, incompatibilities may be revealed *after* the identification of potential exchange cycles, which, as as consequence, may completely fail to be implemented.

A way to tackle this issue is to *first* select a restricted subset of arcs to crossmatch them in order to test their compatibility, and only then to run the matching algorithm in order to find an optimal set of exchange cycles. Smeulders et al. (2022) have proposed a stochastic integer programming formulation of this approach. Namely, they introduce a *two-stage selection problem* which, given a *testing budget* $\mathbf{B}$, identifies (in stage 1) a subset of arcs $B \subseteq A$ with $|B| \leq \mathbf{B}$ such that the expected number of transplants in the graph (V, B) (in stage 2) is maximized. Solving this optimization problem is numerically challenging.

Smeulders et al. (2022) tightened the formulation of the two-stage selection problem by adding constraints which enforce that the set B must be a cycle selection: indeed, arcs that are not contained in any cycle cannot be used in transplants and hence, should not be selected in the first stage. Their work motivates our attempt to develop a better understanding of the cycle selection problem and of its ILP formulations.

2.1.3 Literature review

Rather surprisingly in view of their natural definition, cycle selections have apparently not been previously investigated in the literature, except for the paper of Smeulders et al. (2022) mentioned above and to which we return in Section 2.3.2. Our review, therefore, is limited to a number of related, but different combinatorial problems.

The weighted *girth* problem asks for a simple cycle of minimum total weight in a weighted *undirected* graph G. The *cycle cone* and *cycle polytope* are, respectively the cone generated by the incidence vectors of cycles of G and the convex hull of these vectors. Thus, the weighted girth problem is the optimization problem over the cycle polytope. It is NP-hard in general, but is polynomially solvable when certain restrictions are placed on the cost vectors. On the other hand, the optimization problem over the cycle cone is polynomially solvable. A linear system describing the cycle cone is given in Seymour (1979). An alternative proof of this result, as well as additional properties of the cycle cone and the cycle polytope, are established in Coullard and Pulleyblank (1989).

Bauer (1997) studies the facial structure of the cycle polytope associated with a complete undirected graph on n vertices. She proves that this polytope is full-dimensional for $n \geq 4$, she provides an ILP formulation for it, and she proves that all inequalities in the ILP formulation are facet-defining when $n \geq 6$. She also presents additional classes of facet-defining valid

inequalities, as well as a complete linear description of the cycle polytope when $n \leq 6$. Bauer et al. (2002) extend the previous results to the case where the cycles are restricted to have length at most $\mathbf{K}$, where $0 \leq \mathbf{K} \leq n$. They also experiment with a branch-and-cut algorithm for the solution of the corresponding optimization problem.

Balas and Oosten (2000) investigate the minimum girth problem and the cycle polytope associated with complete *directed* graphs. The optimization problem is again NP-hard, but it is polynomially solvable when all cycles have a positive weight. Balas and Oosten (2000) propose an arc-based ILP formulation of the problem. They prove that the cycle polytope on a complete graph with n vertices is a face of the related polytope on a complete graph with $n + 1$ vertices. This leads them to an efficient general lifting procedure. They also give a partial description of the facets of the cycle polytope. The article Balas and Rüdiger (2009) is a continuation of the previous one. It further studies the cycle polytope, the cycle cone, the upper cycle polyhedron, the dominant of the cycle polytope and their relationships.

Hartmann and Özlük (2001) carry out a polyhedral analysis of the **K**-cycle polytope, which is the convex hull of the incidence vectors of all simple directed cycles with length exactly **K**. They determine the dimension of the **K**-cycle polytope. They describe several sets of valid inequalities and discuss the complexity of the associated separation problems. They also investigate the relationship between the **K**-cycle polytopes of directed and undirected graphs.

In a separate stream of research, the *cardinality constrained multi-cycle problem*(**CCMC**) has been recently studied by several researchers. Given a weighted digraph $G = (V, A)$ and an integer $\mathbf{K}$, the problem is here to find a set of arcs with maximum weight forming a union of *vertex-disjoint* cycles of length at most $\mathbf{K}$. **CCMC** is the underlying combinatorial optimization problem to be solved by kidney exchange programs, as explained in Section 2.1.2. It is NP-hard for each fixed $\mathbf{K} \geq 3$, and polynomially solvable when $\mathbf{K} = 2$ or when $\mathbf{K} = n$ (see Abraham et al. (2007), Roth et al. (2007)). Several IP formulations have been proposed for this problem and are reviewed in Mak-Hau (2017) and Biró et al. (2021). In particular, Abraham et al. (2007) and Roth et al. (2007) give two formulations of exponential size, one where the variables are associated with the arcs of G, and another one where the variables are associated with cycles. Later, Constantino et al. (2013) and Dickerson et al. (2016), among others, proposed more complex but compact (polynomial-size) formulations of **CCMC**, including an *extended edge* formulation and a *position-indexed* formulation. Dickerson et al. (2016) also study the relative strength of the linear relaxation of different formulations.

Mak-Hau (2018) focuses on the polyhedral structure of the arc-based formulation proposed by Roth et al. (2007) when G is a complete digraph. The author proves that three classes of constraints in the initial formulation of the problem are facet-defining for the **CCMC** polytope. Furthermore, she identifies four new classes of valid inequalities. Lam and Mak-Hau (2020) extend the theoretical results of Mak-Hau (2018) and build on them to develop an efficient branch-and-bound-and-cut algorithm for the **CCMC**.

Obviously, cycles and unions of vertex-disjoint cycles of a digraph G are cycle selections of G, so that the following inclusions hold:

$$\text{cycle polytope} \subseteq \text{cycle selection polytope}$$

and

$$\textbf{CCMC} \text{ polytope} \subseteq \text{cycle selection polytope}.$$

The cycle selection problem and the associated polytope have apparently not been investigated until now, but we will be able to draw some inspiration from previous work on related problems in the remainder of the chapter.

2.2 Complexity

The maximum weighted cycle selection problem (**MWCS**) has been introduced in Section 2.1. When all arc weights are nonnegative, a maximum cycle selection of $G = (V, A)$ consists of all the arcs contained in strong components of G. Therefore, in this case, **MWCS** is solvable in linear, $O(n+m)$ time by a simple application of Tarjan's strong component algorithm (Tarjan (1972)).

For arbitrary weights, however, **MWCS** is NP-hard. To see this, consider the corresponding decision problem: given a digraph $G = (V, A)$, a weight $w_{i,j} \in \mathbb{N}$ for each arc $(i, j) \in A$, and a number $w_0 \in \mathbb{N}$, is there a cycle selection B such that $w(B) \geq w_0$?

Theorem 3. *The decision version of the maximum weighted cycle selection problem is strongly NP-complete, even when G is a complete digraph.*

Proof. **MWCS** is clearly in NP. We will prove that **MWCS** is strongly NP-complete by reducing the *hitting set* problem **HS** to it. Recall the definition of the hitting set problem: given a finite set X, a collection $T = \{T_1, \ldots, T_r\}$ of subsets of X, and $t \in \mathbb{N}$, is there a subset $H \subseteq X$ such that $|H| \leq t$ and $T_j \cap H \neq \emptyset$ for all $j \in \{1, \ldots, r\}$? Note that for any instance of **HS**, we can assume without loss of generality that each element of X is in one of the subsets $T_1, \ldots, T_r$, and that $t < r$. **HS** is known to be strongly NP-complete (Karp (1972)).

With an instance (X, T, t) of **HS**, we associate an instance (G, w, w_0) of **MWCS** where G is the complete digraph on the set of vertices $V = \{0\} \cup X \cup T$, the weights on the arcs are:

- for all $j = 1, \ldots, r$, $w(T_j, 0) = r$,
- for all $i \in X$, $w(0, i) = -1$,
- for all $i \in X$ and for all $j = 1, \ldots, r$, if $i \in T_j$, then $w(i, T_j) = 0$,
- all the other arcs have weight $-r$,

and $w_0 = r^2 - t$.

We claim that this instance of **MWCS** has a YES answer if and only if the original instance of **HS** has a YES answer.

First, suppose that the original instance of **HS** has a YES answer, i.e., suppose there exists $H \subseteq X$ where $|H| \leq t$ and $H \cap T_j \neq \emptyset$ for all $j \in \{1, \ldots, r\}$. Then, let us define a cycle selection B in the following way:

$$B = \{(T_j, 0) : j \in \{1, \ldots, r\}\} \cup \{(0, i) : i \in H\} \\ \cup \{(i, T_j) : j \in \{1, \ldots, r\}, i \in H \cap T_j\}.$$

Since each element of H is in one of $T_1, \ldots, T_r$, and since H is a hitting set, B is the union of the 3-cycles $(T_j, 0, i)$, for $j = 1, 2, \ldots, r$ and $i \in H \cap T_j$. Thus, B is indeed a cycle selection and its weight is $w(B) = r^2 - |H| \geq r^2 - t = w_0$, so that the instance of **MWCS** has a YES answer.

Next, suppose conversely that the instance of **MWCS** has a YES answer, in other words that there is a cycle selection B with weight at least $w_0 = r^2 - t$. First, note that each arc of B should be in one of the three sets below:

- $\{(T_j, 0) : j \in \{1, \ldots, r\}\}$,
- $\{(0, i) : i \in X\}$,
- $\{(i, T_j) : j \in \{1, \ldots, r\}, i \in T_j\}$.

Indeed, all arcs not in these three sets have a negative weight $(-r)$ and their inclusion in B would result in a total weight at most equal to $r^2 - r < r^2 - t$, which contradicts our assumption. Moreover, B must contain all arcs $(T_j, 0)$ for all $j \in \{1, \ldots, r\}$, because otherwise $w(B) \leq (r-1)r < r^2 - t$, again a contradiction.

Let now $H = \{i \in X : (0, i) \in B\}$. Then,

$$w(B) = r^2 + \sum_{i \in H} w(0, i) = r^2 - |H|.$$

Since $w(B) \geq r^2 - t$, it follows that $|H| \leq t$.

Finally, we claim that H is a hitting set of T. Indeed, for each $j = 1, \ldots, r$, the arc $(T_j, 0) \in B$ must lie in a cycle of $G_B = (V, B)$. Hence, B must also contain at least one arc of the form (i, T_j) for some i in T_j. Then, $(0, i)$ also is in B, so that i is in H. In conclusion, H is a hitting set with size $|H| \leq t$, meaning that **HS** has a YES answer. □

Since all cycles considered in the proof have length exactly 3, it follows that **MWCS** is NP-complete even when the cycle selection is restricted to contain cycles of length at most 3. On the other hand, **MWCS** is trivially solved when restricted to cycles of length 2: indeed, in this case, the arc $(i, j) \in A$ is in an optimal cycle selection if and only if $(j, i) \in A$ and $w_{i,j} + w_{j,i} \geq 0$.

2.3 Formulations

2.3.1 Arc formulation

Let $G = (V, A)$ be an arbitrary directed graph, with $|V| = n$ and $|A| = m$. In order to obtain a first IP formulation for cycle selections, we introduce the "natural" arc variables $\beta_{i,j} \in \{0, 1\}$, where $\beta_{i,j} = 1$ if arc (i, j) is selected and 0 otherwise, for all $(i, j) \in A$.

The set of vectors of $\{0, 1\}^m$ associated with cycle selections is denoted by P_G, or simply P (we usually omit the reference to G, which will be clear from the context). The convex hull of P (or P_G) is denoted by P^* (or P_G^*) and is called the *cycle selection polytope* associated with G. Since **MWCS** is the linear optimization problem over P^* and is NP-hard, it is probably hopeless to obtain a complete linear description of P^*. One of our main goals in this chapter will be to produce a (partial) description of P^* for complete digraphs.

For now, consider the following set of constraints:

$$\beta_{i,j} \in \{0, 1\} \qquad \forall (i, j) \in A \quad (2.1)$$

$$\beta_{i,j} \leq \sum_{(l,k) \in A : l \in V \setminus S, k \in S} \beta_{l,k} \qquad \forall (i, j) \in A, \forall S \subseteq V : i \in S, j \in V \setminus S. \quad (2.2)$$

We call (2.2) the *return inequalities* for the set P. (The return inequalities are formally similar to the inequalities defining the cycle cone of an undirected graph; see Seymour (1979), Bauer (1997).) Figure 2.3 illustrates the idea behind the return inequalities. For an arc $(i, j) \in A$ and a subset $S \subseteq V$ such that $i \in S, j \in V \setminus S$ fixed, if arc (i, j) is selected, then at least one arc $(l, k) \in A$ such that $l \in V \setminus S, k \in S$ should be selected

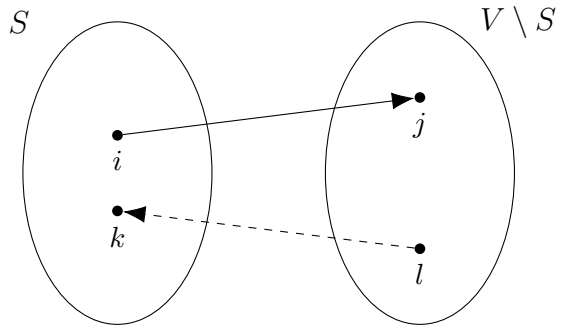


Figure 2.3: Illustration of a return inequality.

Theorem 4. *The constraints* (2.1)-(2.2) *provide a correct formulation of the cycle selection problem, that is,*

$$P = \left\{\beta \in \{0,1\}^m : \beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k} \quad \forall (i,j) \in A, \forall S \subseteq V : i \in S, j \in V \setminus S\right\}.$$

Proof. To show that (2.2) is valid for P, suppose that β describes a cycle selection B which contains the arc (i,j), so that $\beta_{i,j} = 1$. Let $S \subset V$ be a subset of vertices containing i, but not j. Since B is a cycle selection, there is a directed cycle C such that $(i,j) \in C \subseteq B$, i.e., $\beta_{l,k} = 1$ for all $(l,k) \in C$. At least one arc (l,k) of C must have $l \notin S$ and $k \in S$, and hence (2.2) is satisfied.

Conversely, suppose that the point $\beta \in \{0,1\}^m$ satisfies the return inequalities (2.2). Let $B = \{(i,j) : \beta_{i,j} = 1\}$ and $G_B = (V, B)$. For any fixed arc (i,j) such that $\beta_{i,j} = 1$, we must show that (i,j) is contained in at least one directed cycle of G_B. Let $S \subseteq V$ be the set of those vertices k such that there is a directed path $\pi_{k,i}$ from k to i in G_B. Note that $i \in S$. If $j \in S$, then (i,j) is indeed in a directed cycle which is the union of the path $\pi_{j,i}$ and the arc (i,j). Otherwise, $j \in V \setminus S$ and $i \in S$. Because β satisfies the inequality (2.2), there exists $(l,k) \in A$ such that $l \in V \setminus S$, $k \in S$ and $\beta_{l,k} = 1$. But then (l,k) and $\pi_{k,i}$ together form a path from l to i and thus l should be in S, which brings a contradiction. □

We refer to (2.1)-(2.2) as the *arc formulation* of the cycle selection problem, and we define the associated relaxed polytope

$$PL = \left\{\beta \in [0,1]^m : \beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k} \quad \forall (i,j) \in A, \forall S \subseteq V : i \in S, j \in V \setminus S\right\}. \tag{2.3}$$

There holds

$$P \subseteq P^* \subseteq PL.$$

Because of the exponential number of return inequalities (2.2), even optimizing a linear function over PL may not be easy. But our next result implies that cutting plane methods can be used efficiently (and that linear optimization over PL is polynomial, by virtue of the equivalence of optimization and separation; see Grötschel et al. (1981),Conforti et al. (2014)).

Theorem 5. *The separation problem for the relaxed polytope PL is solvable in polynomial time.*

Proof. The separation problem is the following: given a vector $\beta \in [0,1]^m$, is there $(i,j) \in A$ and $S \subset V$ such that $i \in S, j \in V \setminus S$, and $\beta_{i,j} > \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k}$? There are m arcs (i,j) to check, so we can ask the question for each such arc successively.

Since $\beta_{i,j}$ is fixed, we just need to solve $\min_{S\subset V} \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k}$ which is the min-cut problem with source j, sink i, and capacity $\beta_{l,k}$ on each arc (l,k). This (j,i)-min-cut problem is solvable in polynomial time. □

2.3.2 Compact extended formulations

The arc formulation presented in the previous section contains an exponential number of return inequalities (2.2). The aim of this section is to present several compact, polynomial-size formulations of the cycle selection problem and to compare them with the arc formulation.

Extended arc formulations

We start with three formulations based on the relation between cycle selections and circulations. Recall that a *circulation* in a directed graph $G = (V, A)$ is a flow-vector $x \in \mathbb{R}_+^{|A|}$ which is balanced at every vertex, that is, such that

$$\sum_{h:(h,k)\in A} x_{h,k} = \sum_{h:(k,h)\in A} x_{k,h} \qquad \forall k \in V.$$

The *support* of a circulation x is the set $C(x) = \{(i,j) \in A : x_{i,j} > 0\}$. It can be viewed as a cycle selection consisting of m cycles or less (see, e.g., Corollary 4.3.3 in Bang-Jensen and Gutin (2009)). Conversely, every cycle selection B gives rise to a circulation x^B whose support is exactly B, as follows. For each arc $(u,v) \in B$, let $C^{(u,v)}$ be a cycle of G_B containing (u,v), and put a flow of one unit on $C^{(u,v)}$, that is define $x_{i,j}^{(u,v)} = 1$ if $(i,j) \in C^{(u,v)}$, $x_{i,j}^{(u,v)} = 0$ otherwise. Finally, define a circulation x^B as the sum of the cycle flows $x^{(u,v)}$, that is, let $x^B = \sum_{(u,v)\in B} x^{(u,v)}$. Note that this construction does not univocally define x^B, because the choice of the cycles $C^{(u,v)}$ is not unique, but this will be irrelevant for our purpose.

In particular, when x is a binary circulation, then $C(x) = \{(i,j) : x_{i,j} = 1\}$ is an *arc-disjoint* union of cycles, i.e., a special type of cycle selection. If moreover

$$\sum_{h:(h,k)\in A} x_{h,k} \leq 1 \qquad \forall k \in V,$$

then the support of a binary circulation is a union of *vertex-disjoint* cycles.

These observations lead to different formulations for cycle selections. A first *simple extended arc formulation* is as follows: vector $\beta \in \mathbb{R}^{|A|}$ defines a selection if and only there exists $x \in \mathbb{R}_+^{|A|}$ such that

$$x_{i,j} \leq m\,\beta_{i,j} \qquad \forall (i,j) \in A \tag{2.4}$$

$$\beta_{i,j} \leq x_{i,j} \qquad \forall (i,j) \in A \tag{2.5}$$

$$\sum_{h:(h,k)\in A} x_{h,k} = \sum_{h:(k,h)\in A} x_{k,h} \qquad \forall k \in V \tag{2.6}$$

$$0 \leq \beta_{i,j} \leq 1 \qquad \forall (i,j) \in A \tag{2.7}$$

$$\beta_{i,j} \text{ integer} \qquad \forall (i,j) \in A. \tag{2.8}$$

Indeed, any feasible solution of (2.4)-(2.8) defines a subset of arcs B (associated with β) and a circulation x such that B is exactly the support of x. Therefore, B is a cycle selection. Conversely, every cycle selection B gives rise to a feasible solution (β, x^B) as explained above.

A second, more complex but as we will see, tighter formulation relies on expressing that each arc (u,v) of a cycle selection must be contained in the support $C^{(u,v)}$ of a representative *binary* circulation $x^{(u,v)}$ (note that $C^{(u,v)}$ and $x^{(u,v)}$ may differ for each arc (u,v)). We define $x_{i,j}^{(u,v)} = 1$ if $(i,j) \in C^{(u,v)}$, and we interpret $x_{i,j}^{(i,j)} = 1$ to mean that arc (i,j) is in the cycle selection, that is, we identify $x_{i,j}^{(i,j)}$ with $\beta_{i,j}$.

The cycle selection problem can now be formulated as follows:

$$x_{i,j}^{(u,v)} \leq x_{i,j}^{(i,j)} \qquad \forall (i,j) \in A, \forall (u,v) \in A \tag{2.9}$$

$$\sum_{h:(h,k)\in A} x_{h,k}^{(u,v)} = \sum_{h:(k,h)\in A} x_{k,h}^{(u,v)} \qquad \forall k \in V, \forall (u,v) \in A \tag{2.10}$$

$$0 \leq x_{i,j}^{(u,v)} \leq 1 \qquad \forall (i,j) \in A, \forall (u,v) \in A \tag{2.11}$$

$$x_{i,j}^{(u,v)} \text{ integer} \qquad \forall (i,j) \in A, \forall (u,v) \in A \tag{2.12}$$

Constraints (2.10)-(2.12) enforce that each vector $x^{(u,v)}$ is indeed a binary circulation, and constraints (2.9) enforce that arc (i,j) can be in the support $C^{(u,v)}$ only if it is selected at all (if $x_{i,j}^{(u,v)} = 1$, then $x_{i,j}^{(i,j)} \equiv \beta_{i,j}$ must be 1 as well).

The constraints (2.9)-(2.12) provide a correct extended formulation of the cycle selection problem. We refer to it as the *extended arc formulation* of the cycle selection problem, and we note that it is formally similar to the *extended edge formulation* of Constantino et al. (2013) for the cardinality-constrained multi-cycle problem (**CCMC**, see Section 2.1.3). It contains a polynomial number of variables ($O(n^4)$) and a polynomial number of constraints ($O(n^4)$).

Finally, the extended arc formulation can be further adapted by insisting that the support of each binary circulation $x^{(u,v)}$ should consist of vertex-disjoint cycles. This can be achieved by adding the following constraints to the extended arc formulation:

$$\sum_{h:(k,h)\in A} x_{k,h}^{(u,v)} \leq 1 \qquad \forall k \in V, \forall (u,v) \in A. \tag{2.13}$$

We refer to (2.9)-(2.13) as the *modified extended arc formulation* for cycle selections.

Remark 1. One may want to further strengthen these extended formulations so that $C^{(u,v)}$ is a single cycle for each (u,v). This would require to include additional exponential families of inequalities describing the cycle polytope, see Balas and Oosten (2000), Balas and Rüdiger (2009). □

We now aim to establish the relation between the arc formulation of Section 2.3 and the different extended arc formulations introduced above. Let us first denote as P_{EA} the polytope defined by the linear relaxation (2.9)-(2.11) of the extended arc formulation, and recall that PL is the solution set of the relaxation (2.3) of the arc formulation.

Theorem 6. *The polytope PL is the projection of the polytope P_{EA} on the space $\mathbb{R}^{|A|}$ of the variables $\beta_{i,j} \equiv x_{i,j}^{(i,j)}$, $(i,j) \in A$.*

Proof. To prove first that the projection of P_{EA} is contained in PL, let us consider a point $x \in P_{EA}$ and let us set $\beta_{i,j} = x_{i,j}^{(i,j)}$ for all $(i,j) \in A$. We must show that $\beta \in PL$.

The bounding constraints $0 \leq \beta_{i,j} \leq 1$ are satisfied. So, we only need to show that, for each fixed arc $(i,j) \in A$ and each fixed subset $S \subseteq V$ with $i \in S$, $j \notin S$, the return inequality

$$\beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k} \tag{2.14}$$

can be deduced from the inequalities defining P_{EA}. Let us add up the fol-

lowing inequalities:

$$x_{l,k}^{(i,j)} \leq x_{l,k}^{(l,k)} \qquad \forall (l,k) \in (V \setminus S, S), \tag{2.15}$$

$$\sum_{h:(k,h)\in A} x_{k,h}^{(i,j)} - \sum_{h:(h,k)\in A} x_{h,k}^{(i,j)} = 0 \qquad \forall k \in S. \tag{2.16}$$

This yields a new inequality with right-hand side equal to:

$$\sum_{(l,k)\in A: l\in V\setminus S, k\in S} x_{l,k}^{(l,k)} = \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k}.$$

In the left-hand side of the summation, each variable $x_{l,k}^{(i,j)}$, $(l,k) \in (V \setminus S, S)$, appears once with coefficient $+1$ in (2.15) and once with coefficient -1 in (2.16). Also, each variable $x_{k,h}^{(i,j)}$ with $k, h \in S$ appears once with coefficient $+1$ and once with coefficient -1 in (2.16). As a result, the left-hand side of the summation boils down to $\sum_{(k,h)\in A: k\in S, h\in V\setminus S} x_{k,h}^{(i,j)}$. Since $x_{k,h}^{(i,j)} \geq 0$ for all $(k,h) \in A$, the left-hand side of the inequality is at least $x_{i,j}^{(i,j)} = \beta_{i,j}$, and hence we obtain the return inequality (2.14).

Next, to prove that PL is contained in the projection of P_{EA}, we must show that for each feasible solution $\beta^* \in PL$ of the relaxed arc formulation, there exists a solution $x \in P_{EA}$ with $x_{i,j}^{(i,j)} = \beta_{i,j}^*$ for all $(i,j) \in A$.

For every fixed arc $(i,j) \in A$, denote as $G^{(i,j)} = (V, A)$ the digraph (V, A) equipped with the following lower bound $\ell_{h,k}^{(i,j)}$ and upper bound $c_{h,k}^{(i,j)}$ on each arc (h,k) in A:

- if $h \neq i$ and $k \neq j$, then $\ell_{h,k}^{(i,j)} = 0$ and $c_{h,k}^{(i,j)} = \beta_{h,k}^*$;
- if $h = i$ and $k \neq j$, then $\ell_{i,k}^{(i,j)} = c_{i,k}^{(i,j)} = 0$;
- if $h \neq i$ and $k = j$, then $\ell_{h,j}^{(i,j)} = c_{h,j}^{(i,j)} = 0$;
- if $h = i$ and $k = j$, then $\ell_{i,j}^{(i,j)} = c_{i,j}^{(i,j)} = \beta_{i,j}^*$.

We say that a circulation x is *feasible* in $G^{(i,j)}$ if $\ell_{h,k}^{(i,j)} \leq x_{h,k} \leq c_{h,k}^{(i,j)}$ for each arc (h,k) in A. In view of Hoffman's circulation theorem (Hoffman (1960), Bang-Jensen and Gutin (2009)), there exists a feasible circulation in $G^{(i,j)}$ if and only if

$$\sum_{(h,k)\in A: h\in S, k\notin S} \ell_{h,k}^{(i,j)} \leq \sum_{(h,k)\in A: h\notin S, k\in S} c_{h,k}^{(i,j)} \quad \text{for all } S \subseteq V. \tag{2.17}$$

Let us verify that this is indeed the case. Fix the set $S \subseteq V$. With our definition of the lower and upper bounds, the left-hand side of the inequality (2.17) is zero (and hence, the inequality is trivially satisfied) unless $i \in S$

and $j \notin S$, in which case it is equal to $\ell_{i,j}^{(i,j)} = \beta_{i,j}^*$. But then, the right-hand side of (2.17) is equal to $\sum_{(h,k)\in A: h\notin S, k\in S} \beta_{h,k}^*$. So, (2.17) boils down to a return inequality, and it is satisfied in view of the feasibility of β^* for the relaxed arc formulation.

So, we conclude from Hoffman's theorem that for each $(i,j) \in A$, there exists a feasible circulation $x^{(i,j)}$ in $G^{(i,j)}$. Note that due to the bounds on the arcs of $G^{(i,j)}$, $x_{i,j}^{(i,j)} = \beta_{i,j}^*$. Moreover, the collection of circulations $x^{(i,j)}$, for all $(i,j) \in A$, satisfies the constraints (2.9)–(2.11) of the extended arc formulation. Indeed, the constraints (2.10) are satisfied by definition of circulations, and the constraints (2.11) are satisfied because of the lower and upper bounds on the arcs of $G^{(i,j)}$. As for constraints (2.9), consider $(u,v) \in A$ and $(i,j) \in A$.

- If $(u,v) = (i,j)$, then (2.9) is trivial.
- If $u = i$ and $v \neq j$, or if $u \neq i$ and $v = j$, then $x_{i,j}^{(u,v)} \leq c_{i,j}^{(u,v)} = 0$ (upper bound on $x_{i,j}^{(u,v)}$ in the graph $G^{(u,v)}$).
- If $u \neq i$ and $v \neq j$, then $x_{i,j}^{(u,v)} \leq c_{i,j}^{(u,v)} = \beta_{i,j}^*$.

Hence, constraints (2.9) are indeed satisfied. This concludes the proof. □

In view of Theorem 6, the linear relaxation P_{EA} of the extended arc formulation is equivalent to the linear relaxation PL of the arc formulation when it comes to solving the maximum weighted cycle selection problem. We are now going to show that the same conclusion applies when we consider the modified extended arc formulation. We denote by P_{MEA} the polytope defined by inequalities (2.9)-(2.11) and (2.13).

Theorem 7. *The polytope PL is the projection of the polytope P_{MEA} on the space $\mathbb{R}^{|A|}$ of the variables $\beta_{i,j} \equiv x_{i,j}^{(i,j)}$, $(i,j) \in A$.*

Proof. Since $P_{MEA} \subseteq P_{EA}$, Theorem 6 immediately implies that the projection of P_{MEA} is contained in PL.

For the reverse inclusion, consider the collection of circulations $x^{(i,j)}$ obtained in the proof of Theorem 6. Each circulation $x^{(i,j)}$ can be written as a positive linear combination of the form

$$x^{(i,j)} = \sum_{C\in\mathcal{C}^{(i,j)}} \lambda_C^{(i,j)} \xi^C, \tag{2.18}$$

where $\mathcal{C}^{(i,j)}$ is a collection of directed cycles forming the support of $x^{(i,j)}$, ξ^C is the incidence vector of cycle C, and $\lambda_C^{(i,j)} > 0$ for all $C \in \mathcal{C}^{(i,j)}$ (see, e.g.,

Bang-Jensen and Gutin (2009)). If some cycle $C \in \mathcal{C}^{(i,j)}$ does not contain the arc (i,j), then we can remove this cycle from the collection $\mathcal{C}^{(i,j)}$, and the right-hand side of (2.18) still defines a feasible circulation as required in the proof of Theorem 6. So, we can assume without loss of generality that $(i,j) \in C$, or equivalently, $\xi^C_{i,j} = 1$, for all $C \in \mathcal{C}^{(i,j)}$. It then follows from (2.18) that

$$x^{(i,j)}_{i,j} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda^{(i,j)}_C. \tag{2.19}$$

We want to show now that inequality (2.13) is satisfied, for an arbitrary $k \in V$ and for $(u,v) = (i,j)$. The left-hand side of (2.13) is

$$\begin{aligned}\sum_{h:(k,h)\in A} x^{(i,j)}_{k,h} &= \sum_{h:(k,h)\in A} \sum_{C \in \mathcal{C}^{(i,j)}} \lambda^{(i,j)}_C \xi^C_{k,h} \\ &= \sum_{C \in \mathcal{C}^{(i,j)}} \lambda^{(i,j)}_C \Big(\sum_{h:(k,h)\in A} \xi^C_{k,h} \Big).\end{aligned}$$

For each cycle C, $\sum_{h:(k,h)\in A} \xi^C_{k,h}$ is either 1 (if vertex k is on the cycle) or 0 (otherwise). So, we get:

$$\sum_{h:(k,h)\in A} x^{(i,j)}_{k,h} \leq \sum_{C \in \mathcal{C}^{(i,j)}} \lambda^{(i,j)}_C,$$

and from (2.19),

$$\sum_{h:(k,h)\in A} x^{(i,j)}_{k,h} \leq x^{(i,j)}_{i,j} \leq 1,$$

so that the constraint (2.13) is satisfied. This implies that the collection of circulations $x^{(i,j)}$, for all $(i,j) \in A$, satisfies the constraints (2.9)–(2.11) and (2.13) of the modified extended arc formulation, which concludes the proof. □

Finally, we return to the simple extended arc formulation. Let us denote as P_{SEA} the set of solutions of the relaxed formulation (2.4)-(2.7). The proof of the following result suggests that this formulation is quite loose.

Theorem 8. *The polytope PL is included in the projection of the polytope P_{SEA} on the space $\mathbb{R}^{|A|}$ of the variables $\beta_{i,j}$, $(i,j) \in A$, and the inclusion is strict for complete digraphs on $n \geq 2$ vertices.*

Proof. Given $\beta^* \in PL$, consider again the collection of circulations $x^{(u,v)}$, $(u,v) \in A$ obtained in the proof of Theorem 6. Let us define $x^* = \sum_{(u,v)\in A} x^{(u,v)}$, and let us show that $(\beta^*, x^*) \in P_{SEA}$. First, it is clear that x^* is a circulation, i.e., it satisfies the equations (2.6). Since $\beta^*_{i,j} = x^{(i,j)}_{i,j}$, it follows that $\beta^*_{i,j} \leq \sum_{(u,v)\in A} x^{(u,v)}_{i,j} = x^*_{i,j}$, meaning that equation (2.5) is satisfied. Finally,

in view of equation (2.9), $x^*_{i,j} = \sum_{(u,v)\in A} x^{(u,v)}_{i,j} \leq \sum_{(u,v)\in A} x^{(i,j)}_{i,j} = m\beta^*_{i,j}$, hence equation (2.4) is satisfied and $(\beta^*, x^*) \in P_{SEA}$, as required.

To prove that the inclusion is strict when $n \geq 2$, consider the following assignment (only the nonzero values are displayed):

- $\beta_{1,2} = 1.0$, $\beta_{2,1} = 0.5$,
- $x_{1,2} = x_{2,1} = 1$.

Then, $(\beta, x) \in P_{SEA}$, but $\beta \notin PL$ since β does not satisfy the return inequality

$$\beta_{1,2} \leq \sum_{k\in V\setminus\{2\}} \beta_{2,k}. \qquad \square$$

Position-indexed formulation

Another extended formulation has been proposed by Smeulders et al. (2022); it is inspired by the position-indexed edge formulation of the **CCMC** (Dickerson et al. (2016)).

Assuming (without loss of generality) that the vertex-set of the digraph $G = (V, A)$ is $V = \{1, \ldots, n\}$, let us denote by V^l the subset of vertices $\{l, \ldots, n\}$, for each l in V. Given binary values for the arc variables $\beta_{i,j}$, define $B^l = \{(i,j) \in A : i \in V^l, j \in V^l, \beta_{i,j} = 1\}$. Let us then introduce a new set of position-indexed binary variables:

$$\phi^l_{i,j,k} \text{ for all } (i,j) \in A, l \in V, k \in \kappa(i,j,l) \text{ where } \kappa(i,j,l) = \begin{cases} \{1\} \text{ if } i = l \\ \{2, ..., n\} \text{ if } j = l \\ \{2, ..., n-1\} \text{ if } i, j > l \end{cases}$$

with the interpretation that $\phi^l_{i,j,k}$ is equal to 1 if arc (i,j) is in position k in a cycle of the digraph (V^l, B^l) containing vertex l, and 0 otherwise.

Smeulders et al. (2022) propose the following formulation:

$$\beta_{i,j} \leq \sum_{l \in V} \sum_{k \in \kappa(i,j,l)} \phi^l_{i,j,k} \qquad \forall (i,j) \in A \tag{2.20}$$

$$\phi^l_{i,j,k} \leq \beta_{i,j} \qquad \forall l \in V, (i,j) \in A^l, k \in \kappa(i,j,l) \tag{2.21}$$

$$\phi^l_{i,j,k} \leq \sum_{h:(h,i) \in A^l \wedge k-1 \in \kappa(h,i,l)} \phi^l_{h,i,k-1} \qquad \forall l \in V, (i,j) \in A^l, k \in \kappa(i,j,l), k > 1 \tag{2.22}$$

$$\phi^l_{i,j,k} \leq \sum_{h:(j,h) \in A^l \wedge k+1 \in \kappa(j,h,l)} \phi^l_{j,h,k+1} \qquad \forall l \in V, (i,j) \in A^l, j \neq l, k \in \kappa(i,j,l) \tag{2.23}$$

$$0 \leq \phi^l_{i,j,k} \leq 1 \qquad \forall l \in V, (i,j) \in A^l, k \in \kappa(i,j,l) \tag{2.24}$$

$$0 \leq \beta_{i,j} \leq 1 \qquad \forall (i,j) \in A \tag{2.25}$$

$$\phi^l_{i,j,k} \text{ integer} \qquad \forall l \in V, (i,j) \in A^l, k \in \kappa(i,j,l) \tag{2.26}$$

$$\beta_{i,j} \text{ integer} \qquad \forall (i,j) \in A \tag{2.27}$$

Constraints (2.20) express that if an arc is selected, then it is part of at least one cycle, and constraints (2.21) ensure that if an arc is in a cycle, then it has to be selected. Constraints (2.22) enforce that if arc (i,j) is in position k in some cycle of (V^l, B^l), then there must be a preceding arc in position $k-1$, unless $k = 1$ (when $k = 1$, then there is no preceding arc, but because of the definition of $\kappa(i,j,l)$, i is necessarily equal to l for the variables $\phi^l_{i,j,1}$). Similarly, constraints (2.23) enforce that arc (i,j) must have a succeeding arc unless $j = l$ which means that a cycle is completed.

The inequalities (2.20)-(2.27) provide a compact *position-indexed formulation* for cycle selections, with $O(n^4)$ variables and constraints. Their linear relaxation (2.20)-(2.25) describes a polytope P_{PI}. We next show that this relaxation is weaker than the relaxation PL of the arc formulation.

Theorem 9. *The polytope PL is included in the projection of the polytope P_{PI} on the space $\mathbb{R}^{|A|}$ of the variables $\beta_{i,j}$, $(i,j) \in A$, and the inclusion is strict for complete digraphs on $n \geq 4$ vertices.*

Proof. We must prove that for any feasible solution $\beta \in PL$, there exists a solution (β, ϕ) in P_{PI}.

As in the proof of Theorem 7, consider a collection of circulations $x^{(i,j)}$,

$(i,j) \in A$, and write each $x^{(i,j)}$ as a positive linear combination of the form

$$x^{(i,j)} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)} \xi^C, \tag{2.28}$$

where $\mathcal{C}^{(i,j)}$ is a collection of directed cycles containing the arc (i,j), ξ^C is the incidence vector of cycle c, $\lambda_C^{(i,j)} > 0$ for all $C \in \mathcal{C}^{(i,j)}$, and

$$\beta_{i,j} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)}. \tag{2.29}$$

For any two arcs (i,j), $(u,v) \in A$, and for all k,l, define $\mathcal{C}^{(u,v)}(i,j,k,l)$ as the set of cycles $C \in \mathcal{C}^{(u,v)}$ such that arc (i,j) is in position k in C, and the lowest-indexed vertex of C is l.

For all $(i,j) \in A, l \in V, k \in \kappa(i,j,l)$, set now

$$\phi_{i,j,k}^l = \max_{(u,v)\in A} \sum_{C \in \mathcal{C}^{(u,v)}(i,j,k,l)} \lambda_C^{(u,v)}. \tag{2.30}$$

We claim that (β, ϕ) satisfies inequalities (2.20)-(2.25). First, for each $(i,j) \in A$, in view of (2.29), of $\mathcal{C}^{(i,j)} = \bigcup_{k,l} \mathcal{C}^{(i,j)}(i,j,k,l)$, and of (2.30), we get

$$\beta_{i,j} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)} = \sum_{k,l} \sum_{C \in \mathcal{C}^{(i,j)}(i,j,k,l)} \lambda_C^{(i,j)} \leq \sum_{k,l} \phi_{i,j,k}^l,$$

which is exactly inequality (2.20).

Consider next the inequalities (2.21). For given i,j,k,l, the maximum in the right-hand side of (2.30) is achieved for some arc (u,v). With this value of (u,v),

$$\phi_{i,j,k}^l = \sum_{C \in \mathcal{C}^{(u,v)}(i,j,k,l)} \lambda_C^{(u,v)} \leq \sum_{C \in \mathcal{C}^{(u,v)}:(i,j)\in C} \lambda_C^{(u,v)} \leq \beta_{i,j}.$$

The last inequality holds by construction of the circulation $x^{(u,v)}$ in Theorem 6: the sum of the weights $\lambda_c^{(u,v)}$ of the cycles involved in $\mathcal{C}^{(u,v)}$ cannot exceed the upper bound $\beta_{s,t}$ on any arc (s,t). In particular, it cannot exceed $\beta_{i,j}$.

For the inequality (2.22) associated with i,j,k,l, where $k > 1$, assume again that the maximum in equation (2.30) is achieved for arc (u,v). For each

cycle $C \in \mathcal{C}^{(u,v)}(i,j,k,l)$, there is an arc $(h(C),i)$ in position $k-1$ in c. So,

$$\begin{aligned}\phi^l_{i,j,k} &= \sum_{C\in\mathcal{C}^{(u,v)}(i,j,k,l)} \lambda_C^{(u,v)} \\ &= \sum_{C\in\mathcal{C}^{(u,v)}(h(C),i,k-1,l)} \lambda_C^{(u,v)} \\ &\leq \sum_{h\in V^l} \sum_{C\in\mathcal{C}^{(u,v)}(h,i,k-1,l)} \lambda_C^{(u,v)} \\ &\leq \sum_{h\in V^l} \phi^l_{h,i,k-1},\end{aligned}$$

as required.

The case of inequalities (2.23) is similar. Finally, the bounds (2.24) are implied by (2.30) and by (2.21). Thus, we conclude that (β,ϕ) satisfies all inequalities (2.20)-(2.25), and that PL is indeed contained in the projection of P_{PI}.

To prove strict inclusion for complete digraphs with $n \geq 4$ vertices, consider the following assignment for the (β,ϕ) variables (we only list the variables with nonzero value):

- $\beta_{1,3} = \beta_{3,4} = \beta_{4,3} = 1, \beta_{3,1} = 0.5$,
- $\phi^1_{1,3,1} = 1, \phi^1_{3,1,2} = \phi^1_{3,1,4} = \phi^1_{3,4,2} = \phi^1_{4,3,3} = 0.5$,
- $\phi^3_{3,4,1} = \phi^3_{4,3,2} = 0.5$.

These values satisfy all constraints of the relaxed position-indexed formulation. However, with $S = \{1\}$, $i = 1$ and $j = 3$, the return inequality

$$\beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k}$$

is violated by the given assignment since $\beta_{1,3} > \beta_{3,1}$. □

2.3.3 Cycle formulation

Let us define Γ_G (or simply, Γ) as the set of all directed cycles in the digraph G. Like Abraham et al. (2007) and Roth et al. (2007) for **CCMC**, we next propose a formulation for cycle selections based on the cycle variables z_C, $C \in \Gamma$, where $z_C = 1$ if cycle C is selected and 0 otherwise. Then, together with the arc variables $\beta_{i,j}$, the cycle selection problem can be for-

mulated as follows:

$$z_C \leq \beta_{i,j} \qquad \forall C \in \Gamma, \forall (i,j) \in C \qquad (2.31)$$

$$\beta_{i,j} \leq \sum_{C \in \Gamma : (i,j) \in C} z_C \qquad \forall (i,j) \in A \qquad (2.32)$$

$$0 \leq z_C \leq 1 \qquad \forall C \in \Gamma \qquad (2.33)$$

$$0 \leq \beta_{i,j} \leq 1 \qquad \forall (i,j) \in A \qquad (2.34)$$

$$z_C \text{ integer} \qquad \forall C \in \Gamma \qquad (2.35)$$

$$\beta_{i,j} \text{ integer} \qquad \forall (i,j) \in A \qquad (2.36)$$

Constraints (2.31) enforce that if cycle C is selected then all arcs $(i,j) \in C$ must be selected. Constraints (2.32) enforce that if arc $(i,j) \in A$, is selected then at least one cycle containing (i,j) must be selected as well.

The constraints (2.31)-(2.36) provide a valid formulation of the cycle selection problem. We refer to it as the *cycle formulation*. Note that it is an exponential formulation due to the number of potential cycles ($|\Gamma| = O(2^{|m|})$) in graph G.

Denote by P_C the linear relaxation (2.31)-(2.34) of the cycle formulation. This relaxation is again weaker than the relaxation of the arc formulation:

Theorem 10. *The polytope PL is included in the projection of the polytope P_C on the space $\mathbb{R}^{|A|}$ of the variables $\beta_{i,j}$, $(i,j) \in A$, and the inclusion is strict for complete digraphs on $n \geq 4$ vertices.*

Proof. In view of Theorem 6, it suffices to prove that given a feasible solution $x \in P_{EA}$, there exists a solution (β, z) in P_C with $\beta_{i,j} = x_{i,j}^{(i,j)}$ for all $(i,j) \in A$. With the same notations as in the proof of Theorem 7, consider the positive linear combination

$$x^{(i,j)} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)} \xi_C, \qquad (2.37)$$

and the associated expression of $\beta_{i,j}$:

$$\beta_{i,j} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)}. \qquad (2.38)$$

For all $C \in \Gamma$, define

$$z_C = \max_{(u,v) \in A} \lambda_C^{(u,v)}. \qquad (2.39)$$

Consider constraint (2.31) for a given cycle $C^* \in \Gamma$ and an arc $(i,j) \in C^*$. The maximum in the right-hand side of (2.39) is achieved for some arc (u,v),

say, $z_{C^*} = \lambda_{C^*}^{(u,v)}$. So,

$$z_{C^*} = \lambda_{C^*}^{(u,v)} \leq \sum_{C \in \mathcal{C}^{(u,v)}:(i,j) \in C} \lambda_C^{(u,v)} \leq \beta_{i,j}.$$

The last inequality holds by construction of the circulation $x^{(u,v)}$: the sum of the weights of the cycles involved in $\mathcal{C}^{(u,v)}$ cannot exceed the upper bound $\beta_{s,t}$ on any arc (s,t). In particular, it cannot exceed $\beta_{i,j}$ on arc (i,j).

Next, consider constraint (2.32) for an arc $(i,j) \in A$. In view of equations (2.38)-(2.39),

$$\beta_{i,j} = \sum_{C \in \mathcal{C}^{(i,j)}} \lambda_C^{(i,j)} \leq \sum_{C \in \mathcal{C}^{(i,j)}} z_C \leq \sum_{C \in \Gamma:(i,j) \in C} z_C.$$

This shows that PL is contained in the projection of P_C, as required. To prove that the containment is strict when $n \geq 4$, consider the following assignment (only the nonzero values are displayed):

- $\beta_{1,2} = 1.0$, $\beta_{2,3} = \beta_{3,4} = \beta_{4,1} = \beta_{3,1} = 0.5$,
- $z_{(1,2),(2,3),(3,4),(4,1)} = z_{(1,2),(2,3),(3,1)} = 0.5$.

The point (β, z) in in P_C, but $\beta \notin PL$ since β does not satisfy the return inequality

$$\beta_{1,2} \leq \sum_{k \in V \setminus \{2\}} \beta_{2,k}$$

associated with $i = 1$, $j = 2$ and $S = V \setminus \{2\}$. □

2.3.4 Relative strength of formulations

In conclusion, six different formulations of the selection problem have been proposed in this section.

The relative strength of the linear relaxation of these formulations can be described as follows:

- (Theorem 6, Theorem 7.) The arc formulation is equivalent to the extended arc formulation and to the modified extended arc formulation, in the sense that PL is equal to the projection of P_{EA} and of P_{MEA} on the space of the β variables.
- (Theorem 8, Theorem 9, Theorem 10.) The arc formulation is strictly tighter than the simple extended arc formulation, the position-indexed formulation, and the cycle formulation, in the sense that PL is strictly contained in the projection of P_{SEA}, of P_{PI} and of P_C on the space of the β variables.

In view of these results, we focus for the rest of the chapter on the arc formulation of the selection problem.

2.4 Polyhedral structure

Note that any instance of **MWCS** on an incomplete digraph $G = (V, A)$ can be transformed into an instance on a complete digraph by setting a large negative weight $w_{i,j}$ on all pairs $(i, j) \notin A$. Therefore, from now on, we restrict our attention to the case of a complete directed graph $G = (V, A)$, where $|V| = n$ and A contains $m = n(n-1)$ arcs. Our objective is to investigate the polyhedral structure of the cycle selection polytope P^*, which only depends on n in this case.

2.4.1 Dimension

When $|V| = 2$, say $V = \{1, 2\}$, the directed graph only has two arcs $(1, 2), (2, 1)$. The only two feasible cycle selections are the empty cycle selection and the 2-cycle $\{(1, 2), (2, 1)\}$. In this case the dimension of P^* is 1. For the rest of the document, we assume that $|V| \geq 3$.

Theorem 11. *When $|V| \geq 3$, $P^* = conv(P)$ is full-dimensional, that is, $dim(P^*) = n(n-1)$.*

Proof. Suppose that P^* is contained in a hyperplane defined by the equation

$$\sum_{(u,v)\in A} b_{u,v}\beta_{u,v} = b_0. \tag{2.40}$$

We are going to show that the equation (2.40) is of the form: $0 = 0$, which implies that P^* is full-dimensional.

1. Since $0 \in P^*$, we get $b_0 = 0$.
2. Let i, j, k be three distinct vertices in V. Consider the point β^1 with $\beta^1_{i,j} = \beta^1_{j,k} = \beta^1_{k,i} = 1$ and $\beta^1_{u,v} = 0$ for all others arcs $(u, v) \in A$. Since $\beta^1 \in P^*$, it follows that $b_{i,j} + b_{j,k} + b_{k,i} = 0$.
3. For the same three vertices i, j, k as above, let β^2 be such that $\beta^2_{i,j} = \beta^2_{j,k} = \beta^2_{k,i} = \beta^2_{j,i} = 1$ and $\beta^2_{u,v} = 0$ for all others arcs $(u, v) \in A$. Again, $\beta^2 \in P^*$, and the previous conclusions imply that $b_{j,i} = 0$.

It follows that $b_{u,v} = 0$ for all arcs $(u, v) \in A$, as claimed. □

2.4.2 Facets

In this section, we are going to show that the constraints of the arc formulation (2.3) are facet-defining for the cycle selection polytope P^*. As

mentioned before, we assume that $|V| \geq 3$.

Lower bound inequalities

Theorem 12. *For all $(i,j) \in A$, the inequality $\beta_{i,j} \geq 0$ defines a facet of P^*.*

Proof. Fix $(i,j) \in A$, and let F be the face of P^* defined as

$$F = \{\beta \in P^* : \beta_{i,j} = 0\}.$$

Suppose that F is included in a hyperplane defined by the equation

$$\sum_{(u,v)\in A} b_{u,v}\beta_{u,v} = b_0 \tag{2.41}$$

and consider the following binary points $\beta^1, \ldots, \beta^6$ which are all in F.

1. Let $\beta^1 \in F$ be defined by $\beta^1_{u,v} = 0$ for all arcs $(u,v) \in A$, hence $b_0 = 0$.
2. For each $(l,k) \notin \{(i,j),(j,i)\}$, let $\beta^2 \in F$ be defined by $\beta^2_{l,k} = \beta^2_{k,l} = 1$ and $\beta^2_{u,v} = 0$ for all others arcs $(u,v) \in A$. From equation (2.41), We obtain: $b_{l,k} = -b_{k,l}$.
3. For $l \notin \{i,j\}$, let $\beta^3 \in F$ be such that $\beta^3_{j,i} = \beta^3_{l,i} = \beta^3_{i,l} = \beta^3_{l,j} = \beta^3_{j,l} = 1$ and $\beta^3_{u,v} = 0$ for all others arcs $(u,v) \in A$. This yields $b_{j,i} = 0$.
4. For $l \notin \{i,j\}$, let $\beta^4 \in F$ be such that $\beta^4_{j,i} = \beta^4_{l,i} = \beta^4_{i,l} = \beta^4_{l,j} = 1$ and $\beta^4_{u,v} = 0$ for all others arcs $(u,v) \in A$. From (2.41), we get $b_{l,j} = 0$, and together with point 2 here above, $b_{j,l} = 0$.
5. For $l \notin \{i,j\}$, let $\beta^5 \in F$ be such that $\beta^5_{j,i} = \beta^5_{i,l} = \beta^5_{l,j} = 1$ and $\beta^5_{u,v} = 0$ for all others arcs $(u,v) \in A$. We deduce $b_{i,l} = 0$ and from point 2, $b_{l,i} = 0$.
6. If $|V| \geq 4$, fix $l,k \notin \{i,j\}$, and define $\beta^6 \in F$ by $\beta^6_{j,l} = \beta^6_{l,k} = \beta^6_{k,j} = 1$ and $\beta^6_{u,v} = 0$ for all others arcs $(u,v) \in A$. We then obtain $b_{l,k} = 0$.

In conclusion, we find that the equation (2.41) is identical to $b_{i,j}\beta_{i,j} = 0$, and hence F is a facet of the convex hull polytope P^*. □

Upper bound inequalities

Theorem 13. *For all $(i,j) \in A$, the inequality $\beta_{i,j} \leq 1$ defines a facet of P^*.*

Proof. Fix $(i,j) \in A$ and define the face $F = \{\beta \in P^* : \beta_{i,j} = 1\}$. Assume that F is contained in a hyperplane of the form (2.41) and consider the

binary points $\beta^1, \ldots, \beta^6$ below, which are all in $P \cap F$. (From now on, for the sake of brevity, we only explicitly list the nonzero components of each such point.)

1. Let β^1 be such that $\beta^1_{i,j} = \beta^1_{j,i} = 1$.
2. Fix $(l,k) \notin \{(i,j),(j,i)\}$ and let β^2 be such that $\beta^2_{i,j} = \beta^2_{j,i} = \beta^2_{l,k} = \beta^2_{k,l} = 1$.
3. Fix $l \notin \{i,j\}$ and let β^3 be such that $\beta^3_{i,j} = \beta^3_{j,i} = \beta^3_{l,i} = \beta^3_{j,l} = \beta^3_{l,j} = 1$.
4. Fix $l \notin \{i,j\}$ and let β^4 such that $\beta^4_{i,j} = \beta^4_{j,i} = \beta^4_{l,i} = \beta^4_{i,l} = \beta^4_{l,j} = 1$.
5. Fix $l \notin \{i,j\}$ and let β^5 be such that $\beta^5_{i,j} = \beta^5_{j,l} = \beta^5_{l,i} = 1$.
6. If $|V| \geq 4$, fix $l,k \notin \{i,j\}$ and let β^6 be such that $\beta^6_{i,j} = \beta^6_{j,l} = \beta^6_{l,k} = \beta^6_{k,j} = 1$.

By successively substituting these points in (2.41), one concludes that the equation of the hyperplane is of the form $\beta_{i,j} = 1$, up to a multiplicative constant. □

Return inequalities

Theorem 14. *For all $(i,j) \in A$ and for all $S \subseteq V$ such that $i \in S, j \in V \setminus S$, the return inequality*

$$\beta_{i,j} \leq \sum_{(l,k) \in A : l \in V \setminus S, k \in S} \beta_{l,k}$$

defines a facet of P^*.

Proof. Fix $(i,j) \in A$ and S such that $i \in S, j \in V \setminus S$. Let F be the face of P^* defined as

$$F = \left\{ \beta \in P^* : \beta_{i,j} = \sum_{(l,k) \in A : l \in V \setminus S, k \in S} \beta_{l,k} \right\},$$

and consider the following points $\beta^1, \ldots, \beta^{14}$:

1. $\beta^1 = 0$.
2. Let β^2 be such that $\beta^2_{i,j} = \beta^2_{j,i} = 1$.
3. If $|S| \geq 2$, fix $k \in S, k \neq i$, let β^3 be such that $\beta^3_{i,j} = \beta^3_{j,k} = \beta^3_{k,i} = 1$, and let $\beta^{3'}$ be such that $\beta^{3'}_{i,j} = \beta^{3'}_{j,k} = \beta^{3'}_{k,i} = \beta^{3'}_{i,k} = 1$.
4. If $|S| \geq 2$, fix $k \in S, k \neq i$, and let β^4 be such that $\beta^4_{i,k} = \beta^4_{k,i} = 1$.

5. If $|S| \geq 3$, fix $h, k \in S, k \neq i, h \neq i$, and let β^5 be such that $\beta^5_{i,k} = \beta^5_{k,h} = \beta^5_{h,i} = 1$.

6. If $|V \setminus S| \geq 2$, fix $l \in V \setminus S, l \neq j$, let $\beta^6_{i,j} = \beta^6_{j,l} = \beta^6_{l,i} = 1$, and let $\beta^{6'}_{i,j} = \beta^{6'}_{j,l} = \beta^{6'}_{l,j} = \beta^{6'}_{l,i} = 1$.

7. If $|V \setminus S| \geq 2$, fix $l \in V \setminus S, l \neq j$, and let $\beta^7_{j,l} = \beta^7_{l,j} = 1$.

8. If $|V \setminus S| \geq 3$, fix $l, k \in V \setminus S, l \neq j, k \neq j$, and let $\beta^8_{j,l} = \beta^8_{l,k} = \beta^8_{k,j} = 1$.

9. If $|S| \geq 2$, fix $k \in S, k \neq i$, and let $\beta^9_{i,j} = \beta^9_{j,i} = \beta^9_{i,k} = \beta^9_{k,j} = 1$.

10. If $|V \setminus S| \geq 2$, fix $l \in V \setminus S, l \neq j$, and let $\beta^{10}_{i,j} = \beta^{10}_{j,i} = \beta^{10}_{i,l} = \beta^{10}_{l,j} = 1$.

11. If $|S| \geq 2$ and $|V \setminus S| \geq 2$, fix $k \in S, k \neq i$, fix $l \in V \setminus S, l \neq j$, and let $\beta^{11}_{i,j} = \beta^{11}_{j,i} = \beta^{11}_{i,k} = \beta^{11}_{k,l} = \beta^{11}_{l,j} = 1$.

12. If $|S| \geq 2$, fix $k \in S, k \neq i$, and let $\beta^{12}_{i,j} = \beta^{12}_{j,k} = \beta^{12}_{k,i} = 1$.

13. If $|V \setminus S| \geq 2$, fix $l \in V \setminus S, l \neq j$, and let $\beta^{13}_{i,j} = \beta^{13}_{j,l} = \beta^{13}_{l,i} = 1$.

14. If $|S| \geq 2$ and $|V \setminus S| \geq 2$, fix $k \in S, k \neq i$, fix $l \in V \setminus S, l \neq j$, and let $\beta^{14}_{i,j} = \beta^{14}_{j,l} = \beta^{14}_{l,k} = \beta^{14}_{k,i} = 1$.

Note that all the points $\beta^1, \ldots, \beta^{14}$ are in F. Suppose now that F is included in a hyperplane defined by the equation

$$\sum_{(u,v) \in A} b_{u,v} \beta_{u,v} = b_0. \tag{2.42}$$

By successively substituting the points $\beta^1, \ldots, \beta^{14}$ in this equation, one can easily conclude that, up to a multiplicative constant, (2.42) is equivalent to the equation defining F. This proves that F is a facet of P^*. □

We have numerically verified that when $|V| = 3$, the bound inequalities and the return inequalities completely describe the cycle selection polytope P^*. In the following sections, we introduce several additional classes of facet-defining inequalities for the case where $|V| \geq 4$.

Out-star inequalities

Let $t \in \mathbb{N}$, let $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)\}$ be a subset of arcs, and let $I = \{i_1, i_2, \ldots, i_t\}$, $J = \{j_1, j_2, \ldots, j_t\}$. Assume that $I \cap J = \emptyset$ and $1 \leq |I| \leq |J| = t$ (meaning that $j_1, j_2, \ldots, j_t$ are pairwise distinct, but $i_1, i_2, \ldots, i_t$ are not necessarily distinct), so that (V, E) is a collection of disjoint out-stars: in (V, E), each vertex of I has indegree 0 and outdegree at least 1, whereas each vertex of J has indegree 1 and outdegree 0. Let p

and q be two distinct vertices not in $I \cup J$. Then, we can define two *out-star inequalities*:

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \beta_{p,q} \leq \sum_{k \in V \setminus I} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{k,p}, \qquad (2.43)$$

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \beta_{p,q} \leq \sum_{k \in V \setminus I} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{q,k}. \qquad (2.44)$$

As an illustration, Figure 2.4 displays an example of the structure of the arcs involved in the left-hand side of inequalities (2.43) and (2.44).

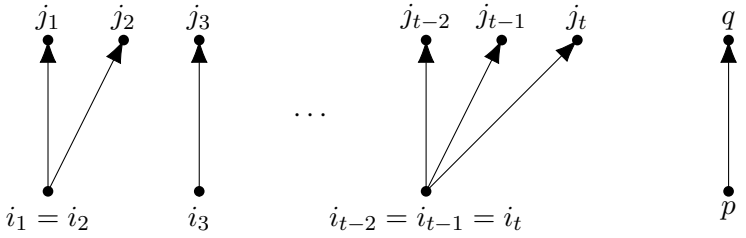


Figure 2.4: Structure of the arcs involved in the left-hand side of inequalities (2.43)-(2.44).

Remark 2. When the out-star inequalities are formally written for $t = 0$, they boil down to special cases of the return inequalities. On the other hand, when $t = 1$, the following point

$$\beta_{12} = \beta_{13} = \beta_{23} = \beta_{41} = \beta_{42} = 0.5, \beta_{34} = 1$$

satisfies all return inequalities of the arc formulation, but not the out-star inequality (2.43) with $i_1 = 1$,$j_1 = 2$, $p = 3$, $q = 4$. □

Let us now focus on the out-star inequality (2.43). We are first going to prove that it is valid for the cycle selection polytope P^*, and next that it is facet defining for P^*. When stating these results, we implicitly assume that $|V| \geq |I| + |J| + 2 \geq 4$ since (2.43) is not defined without this assumption.

Theorem 15. *Let $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)\} \subseteq A$, and let $I = \{i_1, i_2, \ldots, i_t\}$, $J = \{j_1, j_2, \ldots, j_t\}$. Assume that $I \cap J = \emptyset$ and $1 \leq |I| \leq |J| = t$. Let $p, q \in V \setminus (I \cup J)$, $p \neq q$. Then, the out-star inequality* (2.43) *is valid for P^*.*

Proof. Consider any cycle selection B containing exactly s arcs of E, say, the arcs $(i_l, j_l) \in H$ with $H = E \cap B$, $|H| = s$. So, the left-hand side of (2.43) is at most $s + 1$.

If $s \geq 1$, then the arcs in H cover a subset of vertices $I_H \subseteq I$ and a subset of vertices $J_H \subseteq J$, with $|I_H| \geq 1$ and $|J_H| = s$. For each vertex $j \in J_H$, the cycle selection B must contain an arc leaving j, that is, an arc of the form (j, h). All these arc are distinct, hence there are exactly s of them. Moreover, since $I \cap J = \emptyset$, every arc of H leaves I. Hence, there must also be (at least) one arc of B entering I, that is, an arc of the form (k, i) for $k \in V \setminus I$ and $i \in I$. So, in total, the right-hand side of (2.43) is at least $s + 1$, and the inequality holds.

If $s = 0$, then the left-hand side of (2.43) is exactly $\beta_{p,q}$. Assume that $\beta_{p,q} = 1$ (otherwise, the inequality is trivially satisfied). There must be an arc of B entering p, say (h, p). The vertex h is either in I, or in J, or in $V \setminus (I \cup J)$. In the first case, since (h, p) leaves I, there must be an arc entering I, that is, an arc of the form (k, i), $k \notin I$, $i \in I$. So, in all three cases, the right-hand side of (2.43) is at least 1, and this completes the proof. □

Theorem 16. *Let $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)\} \subseteq A$, and let $I = \{i_1, i_2, \ldots, i_t\}$, $J = \{j_1, j_2, \ldots, j_t\}$. Assume that $I \cap J = \emptyset$ and $1 \leq |I| \leq |J| = t$. Let $p, q \in V \setminus (I \cup J)$. Then, the out-star inequality* (2.43) *defines a facet of P^*.*

Proof. Consider the equation

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \beta_{p,q} = \sum_{k \in V \setminus I} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{k,p}. \tag{2.45}$$

Let F be the face of P^* defined as $F = \{\beta \in P^* : \beta \text{ satisfies } (2.45)\}$, and suppose that F is included in a hyperplane defined by the equation

$$\sum_{(u,v) \in A} b_{u,v} \beta_{u,v} = b_0. \tag{2.46}$$

We must show that, up to a multiplicative constant, the equation (2.46) is identical to (2.45).

Hereunder, as usual, we consider a list of points $\beta \in P$ defined by their nonzero components. We denote as $e^{(u,v)}$ the unit vector with $e^{(u,v)}_{u,v} = 1$.

1. Since $\beta^1 = 0 \in F$, we get $b_0 = 0$.
2. Let β^2 be such that $\beta^2_{p,q} = \beta^2_{q,p} = 1$. The point β^2 is in F, and therefore it satisfies equation (2.46). This implies that $b_{p,q} = -b_{q,p}$.

Fix $l \in V, l \notin I \cup J \cup \{p, q\}$.

3. Let $\beta^3_{q,l} = \beta^3_{l,q} = 1$. Since $\beta^3 \in F$, $b_{q,l} = -b_{l,q}$.

4. Let $\beta^4_{p,q} = \beta^4_{q,l} = \beta^4_{l,p} = 1$. This point is in F because it defines a 3-cycle and because it makes both sides of equation (2.45) equal to 1. Next, let $\beta^{4'} = \beta^4 + e^{(l,q)}$. Again, $\beta^{4'}$ is in F. Since β^4 and $\beta^{4'}$ only differ in their component (l, q), equation (2.46) implies that $b_{l,q} = 0$. From point 3 above, we also obtain $b_{q,l} = 0$, and from (2.46), $b_{l,p} = -b_{p,q}$.

So, from points 3 and 4, we know that $b_{q,l} = b_{l,q} = 0$ and $b_{l,p} = b_{q,p} = -b_{p,q}$ for all $l \notin I \cup J \cup \{p, q\}$. Now, fix a pair $(i_s, j_s), 1 \leq s \leq t$, and fix $l \notin I \cup J \cup \{p, q\}$.

5. Let $\beta^5_{i_s,j_s} = \beta^5_{j_s,p} = \beta^5_{p,q} = \beta^5_{q,l} = \beta^5_{l,i_s} = 1$. The point β^5 is in F because it defines a 5-cycle and it makes both sides of (2.45) equal to 2. Hence it satisfies equation (2.46).

   a) The point $\beta^5 + e^{(i_s,l)}$ is in F and by comparison with β^5, it immediately follows that $b_{i_s,l} = 0$ for all $i_s \in I$, for all $l \notin I \cup J \cup \{p, q\}$.

   b) The point $\beta^5 + e^{(l,j_s)}$ is in F (it defines a cycle selection which is the union of a 5-cycle and a 4-cycle), and hence $b_{l,j_s} = 0$ for all $l \notin I \cup J \cup \{p, q\}$, for all $j_s \in J$.

   c) The point $\beta^5 + e^{(i_s,p)}$ is in F, and hence $b_{i_s,p} = 0$ for all $i_s \in I$.

   d) The point $\beta^5 + e^{(i_s,q)}$ is in F, and hence $b_{i_s,q} = 0$ for all $i_s \in I$.

   e) The point $\beta^5 + e^{(p,j_s)}$ is in F, and hence $b_{p,j_s} = 0$ for all $j_s \in J$.

   f) The point $\beta^5 + e^{(q,j_s)}$ is in F, and hence $b_{q,j_s} = 0$ for all $j_s \in J$.

   g) The point $\beta^5 + e^{(p,l)}$ is in F, and hence $b_{p,l} = 0$ for all $l \notin I \cup J \cup \{p, q\}$.

Point 5 has established that all coefficients $b_{i_s,l}$ and b_{l,j_s} are zero, except possibly when $l \in I \cup J$. The coefficients of the variables $\beta_{i_s,i_r}, \beta_{i_s,j_r}, \beta_{j_r,j_s}, \beta_{j_r,i_s}$ for $r \neq s$ will be taken care of at the end of the proof.

6. Let $\beta^6_{i_s,p} = \beta^6_{p,q} = \beta^6_{q,l} = \beta^6_{l,i_s} = 1$. The point β^6 is in F. Since we already know that $b_{i_s,p} = b_{q,l} = 0$, we can conclude $b_{l,i_s} = -b_{p,q}$ for all $l \notin I \cup J \cup \{p, q\}$, for all $i_s \in I$.

7. Let $\beta^7_{i_s,p} = \beta^7_{p,q} = \beta^7_{q,i_s} = 1$. Again, the point β^7 is in F and since $b_{i_s,p} = 0$, we obtain $b_{q,i_s} = -b_{p,q}$ for all $i_s \in I$.

8. Let $\beta^8_{j_s,p} = \beta^8_{p,q} = \beta^8_{q,j_s} = 1$. Since β^8 is in F, we obtain $b_{j_s,p} = -b_{p,q}$ for all $j_s \in J$.

9. Let $\beta^9_{i_s,j_s} = \beta^9_{j_s,p} = \beta^9_{p,q} = \beta^9_{q,i_s} = 1$. The point β^9 is in F because it defines a 4-cycle and it makes both sides of equation (2.45) equal to 2.

   Since we know that $b_{j_s,p} = b_{q,i_s} = -b_{p,q}$, it follows that $b_{i_s,j_s} = b_{p,q}$ for

all pairs (i_s, j_s).

10. Let $\beta^{10}_{i_s,p} = \beta^{10}_{i_s,j_s} = \beta^{10}_{j_s,l} = \beta^{10}_{l,i_s} = \beta^{10}_{p,q} = \beta^{10}_{q,j_s} = 1$. The point β^{10} is in F. Since $b_{l,i_s} = -b_{p,q}$, $b_{i_s,j_s} = b_{p,q}$, and $b_{i_s,p} = b_{q,j_s} = 0$, we conclude $b_{j_s,l} = -b_{p,q}$ for all $l \notin I \cup J \cup \{p,q\}$, for all $j_s \in J$.

11. Let $\beta^{11}_{i_s,j_s} = \beta^{11}_{j_s,q} = \beta^{11}_{q,i_s} = \beta^{11}_{i_s,p} = \beta^{11}_{p,q} = \beta^{11}_{p,j_s} = 1$. Again, the point β^{11} is in F, and since $b_{i_s,j_s} = b_{p,q}$, $b_{q,i_s} = -b_{p,q}$, $b_{i_s,p} = b_{p,j_s} = 0$, we obtain $b_{j_s,q} = -b_{p,q}$ for all $j_s \in J$.

12. Let $\beta^{12}_{i_s,j_s} = \beta^{12}_{j_s,p} = \beta^{12}_{p,q} = \beta^{12}_{q,j_s} = \beta^{12}_{p,i_s} = 1$. The point β^{12} is in F and $b_{i_s,j_s} = b_{p,q}$, $b_{j_s,p} = -b_{p,q}$, $b_{q,j_s} = 0$, so that $b_{p,i_s} = -b_{p,q}$ for all $i_s \in I$.

13. Let $\beta^{13}_{i_s,j_s} = \beta^{13}_{j_s,i_s} = \beta^{13}_{q,j_s} = \beta^{13}_{p,q} = \beta^{13}_{i_s,p} = 1$. The point β^{13} is in F (note that $\beta^{13}_{j_s,i_s}$ contributes for two units to the right-hand side of equation (2.45)). Since $b_{i_s,p} = b_{q,j_s} = 0$ and $b_{i_s,j_s} = b_{p,q}$, we derive $b_{j_s,i_s} = -2b_{p,q}$ for all pairs (i_s, j_s).

Fix now $k, l \in V \setminus (I \cup J \cup \{p,q\})$.

14. Let $\beta^{14}_{l,k} = \beta^{14}_{k,l} = 1$. The point β^{14} is in F and it follows that $b_{l,k} = -b_{k,l}$ for all $k, l \in V \setminus (I \cup J \cup \{p,q\})$.

15. Let $\beta^{15}_{i_s,j_s} = \beta^{15}_{j_s,p} = \beta^{15}_{p,k} = \beta^{15}_{k,l} = \beta^{15}_{l,i_s} = \beta^{15}_{p,q} = \beta^{15}_{q,j_s} = 1$. The point β^{15} is in F, and since $b_{i_s,j_s} = b_{p,q}$, $b_{j_s,p} = b_{l,i_s} = -b_{p,q}$, $b_{p,k} = b_{q,j_s} = 0$, we obtain $b_{k,l} = 0$ for all $k, l \in V \setminus (I \cup J \cup \{p,q\})$.

For the rest of the proof, let $r, s \in \{1, \ldots, t\}$ with $r \neq s$.

16. Let i_r, i_s be two distinct vertices in I, and let $\beta^{16}_{i_r,i_s} = \beta^{16}_{i_s,p} = \beta^{16}_{p,q} = \beta^{16}_{q,i_r} = 1$. The point β^{16} is in F and $b_{i_s,p} = 0$, $b_{q,i_r} = -b_{p,q}$. Therefore, $b_{i_r,i_s} = 0$ for all distinct $i_r, i_s \in I$.

17. Let (i_s, j_s) be an arc in E, and let $i_r \in I$, $i_r \neq i_s$. Let $\beta^{17}_{i_r,i_s} = \beta^{17}_{i_s,j_s} = \beta^{17}_{i_r,j_s} = \beta^{17}_{j_s,p} = \beta^{17}_{p,q} = \beta^{17}_{q,i_r} = 1$. The point β^{17} defines the union of a 5-cycle and of a 4-cycle, and it is in F. From $b_{i_r,i_s} = 0$, $b_{i_s,j_s} = b_{p,q}$, $b_{j_s,p} = b_{q,i_r} = -b_{p,q}$, we derive $b_{i_r,j_s} = 0$ for all $i_r \in I, j_s \in J$, $i_r \neq i_s$.

18. Let (i_s, j_s) be an arc in E, and let $i_r \in I$, $i_r \neq i_s$. Let $\beta^{18}_{i_s,j_s} = \beta^{18}_{j_s,i_r} = \beta^{18}_{i_r,p} = \beta^{18}_{p,q} = \beta^{18}_{q,j_s} = \beta^{18}_{i_r,i_s} = 1$. The point β^{18} is in F because $\beta^{18}_{j_s,i_r}$ contributes for two units to the right-hand side of equation (2.45). From $b_{i_s,j_s} = b_{p,q}$ and $b_{i_r,p} = b_{q,j_s} = b_{i_r,i_s} = 0$, we deduce $b_{j_s,i_r} = -2b_{p,q}$ for all $r \neq s$.

19. Finally, let j_r, j_s be two distinct vertices in J, with $(i_r, j_r) \in E$, $(i_s, j_s) \in E$. If $i_r \neq i_s$, let $\beta^{19}_{i_s,j_s} = \beta^{19}_{j_s,j_r} = \beta^{19}_{j_r,p} = \beta^{19}_{p,q} = \beta^{19}_{q,i_r} = \beta^{19}_{i_r,i_s} = \beta^{19}_{i_r,j_r} = 1$. The point β^{19} is in F: it defines the a union of a 6-cycle and of a 4-cycle, and it makes both sides of equation (2.45) equal

to 3. Since $b_{i_s,j_s} = b_{i_r,j_r} = b_{p,q}$, $b_{j_r,p} = b_{q,i_s} = -b_{p,q}$, and $b_{i_r,i_s} = 0$, we obtain $b_{j_s,j_r} = -b_{p,q}$.
If $i_r = i_s$, the same reasoning applies by simply disregarding the arc (i_r, i_s) in the definition of β^{19}. So, in all cases, $b_{j_s,j_r} = -b_{p,q}$ for all distinct $j_s, j_r \in J$.

The previous observations imply that the equation (2.46) is identical to (2.45), up to a multiplicative constant $b_{p,q}$, and hence F is a facet of the convex hull polytope P^*. □

Theorem 15 and Theorem 16 can be extended to deal with the case of the out-star inequalities (2.44).

Theorem 17. *Let $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)\} \subseteq A$, and let $I = \{i_1, i_2, \ldots, i_t\}$, $J = \{j_1, j_2, \ldots, j_t\}$. Assume that $I \cap J = \emptyset$ and $1 \leq |I| \leq |J| = t$. Let $p, q \in V \setminus (I \cup J)$. Then, the out-star inequality* (2.44) *is valid and defines a facet of P^*.*

Proof. The proof is similar to the previous ones. In particular, points 1-2-7-8-9-11-12-13-14-15-16-17-18-19 in the proof of Theorem 16 can be handled in exactly the same way. The remaining cases involve predecessors of p other than vertices in $I \cup J \cup \{q\}$ and/or successors of q other than vertices in $I \cup J \cup \{p\}$. To deal with these cases, denote as F the face of P^* defined by equation (2.44). Fix $l \in V, l \notin I \cup J \cup \{p, q\}$.

3'. With $\beta^3_{p,l} = \beta^3_{l,p} = 1$, we can conclude that $b_{p,l} = -b_{l,p}$.

4'. Let $\beta^4_{p,q} = \beta^4_{q,l} = \beta^4_{l,p} = 1$, and let $\beta^{4'}_{p,q} = \beta^{4'}_{q,l} = \beta^{4'}_{l,p} = \beta^{4'}_{p,l} = 1$. Both β^4 and $\beta^{4'}$ are in the face F. It easily follows that $b_{p,l} = b_{l,p} = 0$ and $b_{q,l} = -b_{p,q}$ for all $l \notin I \cup J \cup \{p, q\}$.

Now, fix a pair $(i_s, j_s), 1 \leq s \leq t$, and fix $l \notin I \cup J \cup \{p, q\}$.

5'. Let $\beta^5_{i_s,j_s} = \beta^5_{j_s,l} = \beta^5_{l,p} = \beta^5_{p,q} = \beta^5_{q,i_s} = 1$. The point β^5 is in F.

a) Since $\beta^5 + e^{(i_s,l)} \in F$, it follows that $b_{i_s,l} = 0$ for all $i_s \in I$, for all $l \notin I \cup J \cup \{p, q\}$.

b) $\beta^5 + e^{(i_s,p)} \in F$, hence $b_{i_s,p} = 0$ for all $i_s \in I$.

c) $\beta^5 + e^{(i_s,q)} \in F$, hence $b_{i_s,q} = 0$ for all $i_s \in I$

d) $\beta^5 + e^{(l,j_s)} \in F$, hence $b_{l,j_s} = 0$ for all $j_s \in J$, for all $l \notin I \cup J \cup \{p, q\}$.

e) $\beta^5 + e^{(p,j_s)} \in F$, hence $b_{p,j_s} = 0$ for all $j_s \in J$.

f) $\beta^5 + e^{(q,j_s)} \in F$, hence $b_{q,j_s} = 0$ for all $j_s \in J$.

g) $\beta^5 + e^{(l,q)} \in F$, hence $b_{l,q} = 0$ for all $l \notin I \cup J \cup \{p,q\}$.

6'. With $\beta^6_{l,p} = \beta^6_{p,q} = \beta^6_{q,j_s} = \beta^6_{j_s,l} = 1$, we can conclude that $b_{j_s,l} = -b_{p,q}$ for all $l \notin I \cup J \cup \{p,q\}$, for all $j_s \in J$.

10'. Finally, let $\beta^{10}_{i_s,p} = \beta^{10}_{i_s,j_s} = \beta^{10}_{j_s,l} = \beta^{10}_{l,i_s} = \beta^{10}_{p,q} = \beta^{10}_{q,j_s} = 1$ (as in the proof of Theorem 16). Since $\beta^{10} \in F$, we can conclude that $b_{l,i_s} = -b_{p,q}$ for all $i_s \in I$, for all $l \notin I \cup J \cup \{p,q\}$. □

In-star inequalities

Symmetrically with the case of out-star inequalities, we can introduce the class of *in-star inequalities*. With the same notations as in Section 2.4.2, assume that $1 \leq |J| \leq |I| = t$ (meaning that $i_1, i_2, \ldots, i_t$ are all distinct, but $j_1, j_2, \ldots, j_t$ are not necessarily distinct), so that (V, E) is a collection of disjoint in-stars: in (V, E), each vertex of I has indegree 0 and outdegree 1, each vertex of J has outdegree 0 and indegree at least 1. Then, the in-star inequalities are defined as

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \beta_{p,q} \leq \sum_{k \in V} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V \setminus J} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{k,p}, \qquad (2.47)$$

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \beta_{p,q} \leq \sum_{k \in V} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V \setminus J} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{q,k}. \qquad (2.48)$$

Theorem 18. *Let $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)\} \subseteq A$, and let $I = \{i_1, i_2, \ldots, i_t\}$, $J = \{j_1, j_2, \ldots, j_t\}$. Assume that $I \cap J = \emptyset$ and $1 \leq |J| \leq |I| = t$. Let $p, q \in V \setminus (I \cup J)$. Then, the in-star inequalities* (2.47)-(2.48) *are valid and define facets of P^*.*

This theorem can be established by mimicking the proofs in Section 2.4.2. But we prefer to propose here a more insightful argument.

Proof. Define the bijection rev which associates with each $\beta \in \mathbb{R}^{|A|}$ another point $rev(\beta) = \beta^r \in \mathbb{R}^{|A|}$ such that $\beta^r_{i,j} = \beta_{j,i}$ for all $(i,j) \in A$. Intuitively, when $\beta \in \{0,1\}^{|A|}$, then rev simply reverses the direction of each arc in the support of β (remember that we consider here a complete digraph $G = (V, A)$). In particular, if β defines a cycle selection, then so does $rev(\beta)$. As a consequence, rev maps P and P^* onto themselves: $rev(P) = P$ and $rev(P^*) = P^*$.

Consider now the in-star inequality (2.47) associated with E and (p,q), and let $F^{(E,p,q)}$ be the face that it defines. Moreover, consider the out-star inequality (2.44) associated with $E^r = \{(j_1, i_1), (j_2, i_2), \ldots, (j_t, i_t)\}$ and with

the arc (q, p), that is:

$$\sum_{l=1}^{t} \beta_{j_l,i_l} + \beta_{q,p} \leq \sum_{k \in V \setminus J} \sum_{j \in J} \beta_{k,j} + \sum_{i \in I} \sum_{k \in V} \beta_{i,k} + \sum_{k \in V \setminus (J \cup I)} \beta_{p,k}. \qquad (2.49)$$

Let $F^{(E^r,q,p)}$ be the face of P^* defined by (2.49).

If β is in $F^{(E,p,q)}$, that is, if β satisfies (2.47) as an equality, then it is immediately obvious that $rev(\beta)$ satisfies (2.49) as an equality, and hence $rev(\beta)$ is in $F^{(E^r,q,p)}$. The converse relation holds as well, meaning that $rev(F^{(E,p,q)}) = F^{(E^r,q,p)}$.

Since we know from Theorem 17 that $F^{(E^r,q,p)}$ is a facet of P^*, it follows that $F^{(E,p,q)}$ also is a facet of P^*. □

Path inequalities

Let $I = \{i_1, i_2, \ldots, i_t\}$ and $J = \{j_1, j_2, \ldots, j_t\}$ be two subsets of pairwise distinct vertices, $I \cap J = \emptyset$ and $|I| = |J| = t$. Let p and q be two distinct vertices not in $I \cup J$. Then, we define the *path inequality*

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \sum_{l=1}^{t-1} \beta_{i_l,j_{l+1}} + \beta_{p,q} \leq \sum_{k \in V} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{k,p}. \qquad (2.50)$$

Observe that except for (p, q), the arcs involved in the left-hand side of (2.50) define a non directed path $\pi = (j_1, i_1, j_2, i_2, \ldots, j_t, i_t)$. In this path, each arc leaves a vertex of I and enters a vertex of J. As an illustration, Figure 2.5 displays an example of the structure of the arcs involved in the left-hand side of inequality (2.50).

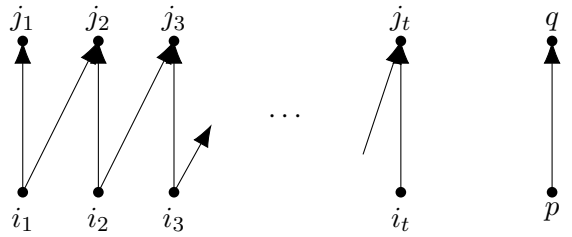


Figure 2.5: Structure of the arcs involved in the left-hand side of inequality (2.50).

We will prove that inequality (2.50) is valid and that it is facet defining for P^*.

Theorem 19. *Let $I = \{i_1, i_2, \ldots, i_t\}$ and $J = \{j_1, j_2, \ldots, j_t\}$ be two subsets of vertices with $I \cap J = \emptyset$ and $1 \leq |I| = |J| = t$. Let $p, q \in V \setminus (I \cup J)$, $p \neq q$. Then, the path inequality* (2.50) *is valid for P^*.*

Proof. Assume that we have a cycle selection B containing exactly s arcs of the path π, say, the arcs $(k,l) \in H$, $|H| = s$. So, the left-hand side of (2.50) is at most $s+1$.

If $s \geq 1$, then the arcs in H form a collection of disjoint subpaths. These subpaths contain a subset of vertices $I_H \subseteq I$ and a subset of vertices $J_H \subseteq J$, and there holds $|I_H| + |J_H| \geq s+1$. For each vertex $i \in I_H$, the cycle selection B must contain an arc entering i. And for each vertex $j \in J_H$, B must contain an arc leaving j. So, the right-hand side of (2.50) is at least $|I_H| + |J_H| \geq s+1$, which implies that (2.50) is satisfied.

If $s = 0$, then the left-hand side of (2.50) is exactly $\beta_{p,q}$. So, assume that $\beta_{p,q} = 1$. There must be an arc of B entering p, say (h,p). The vertex h is either $i_l \in I$ (in which case B must also contain an arc (k,i_l) entering i_l), or $j_l \in J$, or h is not in $I \cup J$. In all three cases, the right-hand side of (2.50) is at least 1, which completes the proof. □

Theorem 20. *Let $I = \{i_1, i_2, \ldots, i_t\}$ and $J = \{j_1, j_2, \ldots, j_t\}$ be two subsets of vertices with $1 \leq I \cap J = \emptyset$ and $|I| = |J| = t$. Let $p, q \in V \setminus (I \cup J)$, $p \neq q$. Then, the path inequality* (2.50) *defines a facet of P^*.*

Proof. Let F be the face of P^* defined by

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \sum_{l=1}^{t-1} \beta_{i_l,j_{l+1}} + \beta_{p,q} = \sum_{k \in V} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{k,p}, \tag{2.51}$$

and suppose that F is included in a hyperplane

$$\sum_{(u,v) \in A} b_{u,v} \beta_{u,v} = b_0. \tag{2.52}$$

The first 15 conclusions in the proof of Theorem 16 can be drawn here in exactly the same way. To see this, it is enough to notice that the cycle selections B^n defined by the points β^n ($n = 1, \ldots, 15$) in the proof of Theorem 16 do not contain any arc (i_s, j_r) with $s \neq r$. Hence the left-hand sides of (2.45) and (2.51) are equal for all these points. Moreover, the cycle selections B^n do not contain any arc of the form (i_s, i_r) with $s \neq r$, so that the right-hand sides of (2.45) and (2.51) are also equal in all cases.

For the remainder of the proof, we have to determine the coefficients of the variables representing the arcs with both vertices in $I \cup J$ (except for the pairs (i_s, j_s) and (j_s, i_s), since we already know the coefficients b_{i_s,j_s} and b_{j_s,i_s} as functions of $b_{p,q}$). Let us consider two distinct pairs (i_s, j_s) and (i_r, j_r) with $s < r$ (note that their order matters, because of the definition of the

path inequalities). We have to find the value of the coefficients b_{i_s,j_r}, b_{j_r,i_s}, b_{i_s,i_r}, b_{j_s,j_r}, b_{j_r,j_s}, b_{j_s,i_r}, b_{i_r,j_s}, b_{i_r,i_s}.

For the coefficients b_{i_s,j_r} and b_{j_r,i_s}, we further have to deal with two subcases:

(i) s and r are consecutive, i.e., $r = s+1$; in that case, we must show that $b_{i_s,j_{s+1}} = b_{p,q}$ and $b_{j_{s+1},i_s} = -2b_{p,q}$;

(ii) s and r are not consecutive, i.e., $r > s+1$; in that case, we must show that $b_{i_s,j_r} = 0$ and $b_{j_r,i_s} = -2b_{p,q}$.

For all the other coefficients, a single analysis will cover both situations.

Case $r = s + 1$.

16. Let β^{16} be such that $\beta^{16}_{i_s,j_{s+1}} = \beta^{16}_{j_{s+1},p} = \beta^{16}_{p,q} = \beta^{16}_{q,i_s} = 1$ and $\beta^{16}_{u,v} = 0$ for all other arcs $(u,v) \in A$. This point is in F because it makes both sides of (2.51) equal to 2. From (2.52), we get: $b_{i_s,j_{s+1}} + b_{j_{s+1},p} + b_{p,q} + b_{q,i_s} = b_0$. Since we already know that $b_0 = 0$ and $b_{j_{s+1},p} = b_{q,i_s} = -b_{p,q}$, we can conclude that $b_{i_s,j_{s+1}} = b_{p,q}$.

17. Next, let $\beta^{17}_{i_s,j_{s+1}} = \beta^{17}_{j_{s+1},i_s} = \beta^{17}_{q,j_{s+1}} = \beta^{17}_{p,q} = \beta^{17}_{i_s,p} = 1$. The point β^{17} is in F (it makes again both sides of (2.51) equal to 2). Hence: $b_{i_s,j_{s+1}} + b_{j_{s+1},i_s} + b_{q,j_{s+1}} + b_{p,q} + b_{i_s,p} = 0$. Since $b_{i_s,p} = b_{q,j_{s+1}} = 0$ and $b_{i_s,j_{s+1}} = b_{p,q}$, we can conclude $b_{j_{s+1},i_s} = -2b_{p,q}$.

Case $r > s + 1$.

18. Let $\beta^{18}_{p,q} = \beta^{18}_{j_r,i_s} = \beta^{18}_{i_h,j_h} = \beta^{18}_{i_s,j_{s+1}} = \beta^{18}_{i_h,j_{h+1}} = \beta^{18}_{q,i_h} = \beta^{18}_{j_h,p} = 1$ for all h with $s < h < r$.

    The point β^{18} defines a cycle selection which is the union of the following 4-cycles: $\{(i_h,j_h),(j_h,p),(p,q),(q,i_h)\}$ for $s < h < r$, $\{(i_h,j_{h+1}),(j_{h+1},p),(p,q),(q,i_h)\}$ for $s < h < r-1$, and of the 6-cycle $\{(i_s,j_{s+1}),(j_{s+1},p),(p,q),(q,i_{r-1}),(i_{r-1},j_r),(j_r,i_s)\}$. Moreover, β^{18} is in F as well, as it makes both sides of (2.51) equal to $2(r-s)$.

    From (2.52), we get:

$$b_{p,q} + b_{j_r,i_s} + \sum_{h=s+1}^{r-1} b_{i_h,j_h} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s+1}^{r-1} b_{q,i_h} + \sum_{h=s+1}^{r-1} b_{j_h,p} = 0.$$

    Since we know that $b_{i_h,j_h} = b_{i_h,j_{h+1}} = -b_{q,i_h} = -b_{j_h,p} = b_{p,q}$ for $s \le h < r$, we can conclude that $b_{j_r,i_s} = -2b_{p,q}$.

    Moreover, the point $\beta^{18} + e^{(i_s,j_r)}$ also is in F, and hence $b_{i_s,j_r} = 0$.

Remaining coefficients b_{i_s,i_r}, b_{j_s,j_r}, b_{j_r,j_s}, b_{j_s,i_r}, b_{i_r,j_s}, b_{i_r,i_s}.

19. Let $\beta^{19}_{p,q} = \beta^{19}_{i_s,i_r} = 1$, $\beta^{19}_{i_h,j_h} = \beta^{19}_{j_h,p} = 1$ for $s \le h \le r$, $\beta^{19}_{i_h,j_{h+1}} = \beta^{19}_{q,i_h} = 1$ for $s \le h < r$.

    The point β^{19} defines a cycle selection which is the union of the following 4-cycles and 5-cycle: $\{(i_h,j_h),(j_h,p),(p,q),(q,i_h)\}$ for $s \le h < r$, $\{(i_h,j_{h+1}),(j_{h+1},p),(p,q),(q,i_h)\}$ for $s \le h < r$,
    and $\{(i_s,i_r),(i_r,j_r),(j_r,p),(p,q),(q,i_s)\}$. Moreover, $\beta^{19} \in F$.

    From (2.52), we get:

$$b_{p,q} + b_{i_s,i_r} + \sum_{h=s}^{r} b_{i_h,j_h} + \sum_{h=s}^{r} b_{j_h,p} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s}^{r-1} b_{q,i_h} = 0.$$

    Since $b_{i_h,j_h} = b_{i_h,j_{h+1}} = -b_{q,i_h} = -b_{j_h,p} = b_{p,q}$ for $s \le h \le r$, we obtain $b_{i_s,i_r} = -b_{p,q}$.

20. Let $\beta^{20}_{p,q} = \beta^{20}_{j_s,j_r} = 1$, $\beta^{20}_{i_h,j_h} = \beta^{20}_{i_h,j_{h+1}} = \beta^{20}_{q,i_h} = 1$ for $s \le h < r$, $\beta^{20}_{j_h,p} = 1$ for $s < h \le r$.

    The point β^{20} defines a cycle selection which is the union of the following cycles: $\{(i_h,j_h),(j_h,p),(p,q),(q,i_h)\}$ for $s < h < r$,
    $\{(i_h,j_{h+1}),(j_{h+1},p),(p,q),(q,i_h)\}$ for $s \le h < r$, and
    $\{(i_s,j_s),(j_s,j_r),(j_r,p),(p,q),(q,i_s)\}$. Since β^{20} is in F, we obtain

$$b_{p,q} + b_{j_s,j_r} + \sum_{h=s}^{r-1} b_{i_h,j_h} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s}^{r-1} b_{q,i_h} + \sum_{h=s+1}^{r} b_{j_h,p} = 0.$$

    We know that $b_{i_h,j_h} = b_{i_h,j_{h+1}} = -b_{q,i_h} = -b_{j_{h+1},p} = b_{p,q}$ for $s \le h < r$, and hence $b_{j_s,j_r} = -b_{p,q}$.

21. Let $\beta^{21}_{p,q} = \beta^{21}_{j_r,j_s} = \beta^{21}_{q,j_r} = \beta^{21}_{i_h,j_h} = \beta^{21}_{i_h,j_{h+1}} = \beta^{21}_{p,i_h} = \beta^{21}_{j_h,p} = 1$ for $s \le h < r$.

    Then, β^{21} defines a cycle selection, as the union of the following cycles: $\{(i_h,j_h),(j_h,p),(p,i_h)\}$ for $s \le h < r$, $\{(i_h,j_{h+1}),(j_{h+1},p),(p,i_h)\}$ for $s \le h < r$, $\{(p,q),(q,j_r),(j_r,j_s),(j_s,p)\}$.

    Since $\beta^{21} \in F$, we get:

$$b_{p,q} + b_{j_r,j_s} + b_{q,j_r} + \sum_{h=s}^{r-1} b_{i_h,j_h} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s}^{r-1} b_{p,i_h} + \sum_{h=s}^{r-1} b_{j_h,p} = 0.$$

    From $b_{i_h,j_h} = b_{i_h,j_{h+1}} = -b_{p,i_h} = -b_{j_h,p} = b_{p,q}$ for $s \le h < r$ and $b_{q,j_r} = 0$, we conclude that $b_{j_r,j_s} = -b_{p,q}$.

22. Let $\beta^{22}_{p,q} = \beta^{22}_{j_s,i_r} = 1$, $\beta^{22}_{i_h,j_h} = 1$ for $s \le h \le r$, $\beta^{22}_{i_h,j_{h+1}} = \beta^{22}_{q,i_h} = 1$ for $s \le h < r$, $\beta^{22}_{j_h,p} = 1$ for $s < h \le r$. The point β^{22} defines a cycle selection which is the union of the cycles: $\{(i_h, j_h), (j_h, p), (p, q), (q, i_h)\}$ for $s < h < r$, $\{(i_h, j_{h+1}), (j_{h+1}, p), (p, q), (q, i_h)\}$ for $s \le h < r$, and $\{(i_s, j_s), (j_s, i_r), (i_r, j_r), (j_r, p), (p, q), (q, i_s)\}$. Because $\beta^{22} \in F$, there holds:

$$b_{p,q} + b_{j_s,i_r} + \sum_{h=s}^{r} b_{i_h,j_h} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s}^{r-1} b_{q,i_h} + \sum_{h=s+1}^{r} b_{j_h,p} = 0.$$

Since $b_{i_h,j_h} = -b_{q,i_h} = -b_{j_h,p} = b_{p,q}$ for $s \le h \le r$, and $b_{i_h,j_{h+1}} = b_{p,q}$ for $s \le h < r$, we obtain $b_{j_s,i_r} = -2b_{p,q}$.

23. Considering the point $\beta^{23} = \beta^{22} + e^{(i_r,j_s)} \in F$, we further derive $b_{i_r,j_s} = 0$.

24. Let $\beta^{24}_{p,q} = \beta^{24}_{i_r,i_s} = \beta^{24}_{j_r,q} = 1$, $\beta^{24}_{i_h,j_h} = 1$ for $s \le h \le r$, $\beta^{24}_{i_h,j_{h+1}} = \beta^{24}_{j_h,i_{h+1}} = \beta^{24}_{q,j_h} = 1$ for $s \le h < r$, $\beta^{24}_{i_h,p} = 1$ for $s < h \le r$.

The point β^{24} defines a cycle selection as union of the cycles:
$\{(i_h, j_h), (j_h, i_{h+1}), (i_{h+1}, p), (p, q), (q, j_{h-1}), (j_{h-1}, i_h)\}$ for $s < h < r$,
$\{(i_h, j_{h+1}), (j_{h+1}, i_{h+2}), (i_{h+2}, p), (p, q), (q, j_{h-1}), (j_{h-1}, i_h)\}$ for $s < h < r-1$,
$\{(i_s, j_s), (j_s, i_{s+1}), (i_{s+1}, p), (p, q), (q, j_{r-1}), (j_{r-1}, i_r), (i_r, i_s)\}$,
$\{(i_r, j_r), (j_r, q), (q, j_{r-1}), (j_{r-1}, i_r)\}$,
and $\{(i_{r-1}, j_r), (j_r, q), (q, j_{r-2}), (j_{r-2}, i_{r-1})\}$.

Since $\beta^{24} \in F$, there holds

$$b_{p,q} + b_{i_r,i_s} + b_{j_r,q} + \sum_{h=s}^{r} b_{i_h,j_h} + \sum_{h=s}^{r-1} b_{i_h,j_{h+1}} + \sum_{h=s}^{r-1} b_{j_h,i_{h+1}} + \sum_{h=s}^{r-1} b_{q,j_h} + \sum_{h=s+1}^{r} b_{i_h,p} = 0.$$

Using $b_{i_h,j_h} = -b_{j_r,q} = b_{p,q}$ for $s \le h \le r$, $b_{i_h,j_{h+1}} = b_{p,q}$ for $s \le h < r$, $b_{i_h,p} = b_{q,j_h} = 0$ for $s \le h \le r$, $b_{j_h,i_{h+1}} = -2b_{p,q}$ for $s \le h < r$, we can finally conclude that $b_{i_r,i_s} = -b_{p,q}$.

This completes the proof that F is a facet of P^*. □

Just as in the case of the out-star and in-star inequalities, the following variant of the path inequality is also valid and facet-defining for the cycle selection polytope:

$$\sum_{l=1}^{t} \beta_{i_l,j_l} + \sum_{l=1}^{t-1} \beta_{i_l,j_l+1} + \beta_{p,q} \le \sum_{k \in V} \sum_{i \in I} \beta_{k,i} + \sum_{j \in J} \sum_{k \in V} \beta_{j,k} + \sum_{k \in V \setminus (I \cup J)} \beta_{q,k}. \tag{2.53}$$

We omit the proof of this result.

2.4.3 Additional valid inequalities

In this section, we describe one last class of valid inequalities.

Theorem 21. *Let i, j, p, q be four distinct vertices in V. Then, the following inequality is valid for the cycle selection polytope P^*:*

$$\beta_{i,j}+\beta_{i,q}+\beta_{p,j} \leq \sum_{k\in V} \beta_{k,i}+\sum_{k\in V} \beta_{j,k}+ \sum_{k\notin\{i,j\}} \beta_{q,k}+ \sum_{k\notin\{i,j\}} \beta_{k,q}+ \sum_{k\notin\{i,j,q\}} \beta_{k,p}+\beta_{q,j}+\beta_{i,p}. \tag{2.54}$$

Proof. In order to establish this result, we rely on the Chvátal-Gomory procedure. Consider the following valid inequalities for P^* (all of them, except the last one are special instances of the return inequalities (2.2)):

- $\beta_{i,q} \leq \sum_{k\in V} \beta_{q,k}$,
- $\beta_{i,q} \leq \sum_{k\in V} \beta_{k,i}$,
- $\beta_{p,j} \leq \sum_{k\in V} \beta_{j,k}$,
- $\beta_{p,j} \leq \sum_{k\in V} \beta_{k,p}$,
- $\beta_{i,j} \leq \sum_{l\notin\{i,q\}} \sum_{k\in\{i,q\}} \beta_{l,k}$ (i.e., the return inequality with $S = \{i, q\}$),
- $\beta_{i,j} \leq 1$.

By adding all these inequalities, dividing the result by 2, and rounding each coefficient according to the Chvátal-Gomory procedure, we obtain the inequality (2.54). □

Remark 3. We have verified numerically that, when $n = |V| = 4$, the inequalities (2.54) define facets of the cycle selection polytope P^* and that, together with the star inequalities (or equivalently, with the path inequalities), they completely describe P^*. □

2.5 Constrained cycle selections

In Smeulders et al. (2022), the authors consider cycle selections which contain at most **B** arcs and which are unions of directed cycles of length at most **K**, where **B** and **K** are two given constants. (See Section 2.1.2: the cycle length restriction is customary in kidney exchange models; the bound on the cardinality of the selections represents a budget constraint on the cost of crossmatch tests.)

The cardinality constraint on the number of selected arcs is easily incorpo-

rated in the arc formulation: it simply requires that

$$\sum_{(i,j)\in A} \beta_{i,j} \leq \mathbf{B}. \tag{2.55}$$

The cycle length constraint, however, is less natural in this formulation. (Smeulders et al. (2022) rely on the PI formulation to express it.) Nevertheless, we can define $P(\mathbf{B}, \mathbf{K})$ to be the set of $\beta \in \{0,1\}^{|A|}$ associated with $(\mathbf{B}, \mathbf{K})$-constrained cycle selections in complete digraphs, and $P(\mathbf{B}, \mathbf{K})^*$ to be its convex hull.

By simple inspection of the polyhedral results established in Section 2.4, we can observe that these results remain valid for $P(\mathbf{B}, \mathbf{K})^*$ when $\mathbf{B}$ and $\mathbf{K}$ are large enough. For example, the proof of Theorem 11 does not involve any selection containing more than 4 arcs, nor any cycle of length larger than 3. It follows that $P(\mathbf{B}, \mathbf{K})^*$ is full-dimensional when $\mathbf{B} \geq 4$ and $\mathbf{K} \geq 3$.

These observations are summarized in the table below. One should read that each theorem remains valid for $P(\mathbf{B}, \mathbf{K})^*$ as long as $\mathbf{B} \geq \mathbf{B}_0$ and $\mathbf{K} \geq \mathbf{K}_0$.

Valid result for $P(\mathbf{B}, \mathbf{K})^*$	$\mathbf{B}_0$	$\mathbf{K}_0$
Theorem 11 (Dimension)	4	3
Theorems 12–13 (Bound inequalities)	5	3
Theorem 14 (Return inequalities)	5	4
Theorems 16–18 (Star inequalities)	7	6
Theorem 20 (Path inequalities)	$5t-1$	7

2.6 Cycle selections with cycles of length at most 3

In the previous section, we have observed that the theorems identifying facets of $P(\mathbf{B}, \mathbf{K})^*$ remain valid as long as $\mathbf{B} \geq \mathbf{B}_0$ and $\mathbf{K} \geq \mathbf{K}_0$ for fixed values of $\mathbf{B}_0$ and $\mathbf{K}_0$ depending on the statement under consideration. However, these results do not provide information about a complete formulation of $P(\mathbf{B}, \mathbf{K})$. As already mentioned, the budget constraint is easily incorporated in the arc formulation, but it is not so for the cycle length constraint. This section focuses on a formulation of the set of cycle selections with cycles of length at most 3.

2.6.1 Formulation

By a small abuse of language, we say that (S, T) is a partition of a set W if $S \cup T = W$ and $S \cap T = \emptyset$, where either S or T may be empty. We denote by $Part(W)$ the set of all partitions of W.

The arc formulation for the cycle selections of a graph $G = (V, A)$ when the

cycles are restricted to length at most 3 is the following:

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A \quad (2.56)$$

$$\beta_{i,j} \leq \sum_{k \in S:(j,k) \in A} \beta_{j,k} + \sum_{k \in T:(k,i) \in A} \beta_{k,i} \qquad \forall (i,j) \in A : (j,i) \notin A, \forall (S,T) \in Part(V \setminus \{i,j\}) \quad (2.57)$$

$$\beta_{i,j} \leq \beta_{j,i} + \sum_{k \in S:(j,k) \in A} \beta_{j,k} + \sum_{k \in T:(k,i) \in A} \beta_{k,i} \qquad \forall (i,j) \in A : (j,i) \in A, \forall (S,T) \in Part(V \setminus \{i,j\}) \quad (2.58)$$

For short, we simply denote by P_3 the set of β-vectors defined by (2.56)-(2.58), without explicit reference to the graph G. When G is complete, P_3 is exactly the set $P(n,3)$ introduced in the previous section. Note that:

- for $(i,j) \in A$, either (2.57) or (2.58) is present depending on whether $(j,i) \in A$ or not;
- when either S or T is empty, (2.57) or (2.58) boils down to the so-called *predecessor* or *successor return inequalities*, respectively.

Theorem 22. *The constraints* (2.56)-(2.58) *provide a correct formulation of the cycle selection problem when the maximum cycle length is restricted to* 3.

Proof. Consider a cycle selection β with cycles of length at most 3. For each arc (i,j) such that $\beta_{i,j} = 1$, either $(j,i) \in A$ and $\beta_{j,i} = 1$ (2-cycle), or there is a vertex $k \neq i,j$ such that $\beta_{j,k} = \beta_{k,i} = 1$ (3-cycle). For each partition (S,T) of $V \setminus \{i,j\}$, k is either in S or in T. Hence, β is in P_3.

Conversely, assume that $\beta \in P_3$, let B be the corresponding set of arcs, and consider an arc $(i,j) \in B$ ($\beta_{i,j} = 1$). We must show that (i,j) is in a 2-cycle or in a 3-cycle of G_B. Define $S = \{k \in V : (k,i) \in A \text{ and } \beta_{k,i} = 1\}$.

- If $j \in S$, then $\beta_{j,i} = 1$ and (i,j) is in a 2-cycle.
- If $j \notin S$, then $S \subseteq V \setminus \{i,j\}$. Let $T = (V \setminus \{i,j\}) \setminus S$. Consider constraints (2.57) and (2.58) associated with (S,T). Note that they are identical in this case, because $j \notin S$ means that either $(j,i) \notin A$ or $\beta_{j,i} = 0$. Furthermore, by definition of S, $\beta_{k,i} = 0$ for all $(k,i) \in A$, $k \in T$. So, by (2.57)-(2.58), there must exist $k \in S$ such that $(j,k) \in A$ and $\beta_{j,k} = 1$. Then, (i,j,k) forms a 3-cycle in B.

□

Theorem 23. *The separation problem for the linear relaxation of P_3 is solvable in polynomial time.*

Proof. The separation problem is the following: given a vector $\beta \in [0,1]^{|A|}$, is there $(i,j) \in A$ and a partition (S,T) of $V \setminus \{i,j\}$ such that $\beta_{i,j} > \beta_{j,i} + \sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i}$ if $(j,i) \in A$ or $\beta_{i,j} > \sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i}$ if $(j,i) \notin A$?

We can ask the question for each arc (i,j) successively. When (i,j) is fixed, we know whether $(j,i) \in A$ or not. We just need to identify the partition (S,T) which minimizes $\sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i}$ and check whether the relevant strict inequality mentioned above is satisfied. In order to identify this partition, we compare $\beta_{j,k}$ and $\beta_{k,i}$ for each $k \in V \setminus \{i,j\}$. If $\beta_{j,k} > \beta_{k,i}$, then we assign k to T, and otherwise we assign k to S.

This proves that the separation problem is solvable in polynomial time. □

2.6.2 Polyhedral study

We now restrict our attention to the case of a complete digraph G where $|V| = n$ and A contains $n(n-1)$ arcs. Then P_3 is defined as

$$P_3 = \left\{ \beta \in \{0,1\}^{|A|} : \beta_{i,j} \leq \beta_{j,i} + \sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i} \; \forall (i,j) \in A, \forall (S,T) \in Part(V \setminus (i,j)) \right\}.$$

The next results provide information about the structure of the polytope $P_3^* = conv(P_3)$.

Theorem 24. *When $|V| = 2$, the dimension of P_3^* is 1. When $|V| \geq 3$, P_3^* is full-dimensional, that is, $dim(P_3^*) = n(n-1)$.*

Theorem 25. *For all $(i,j) \in A$, the inequality $\beta_{i,j} \geq 0$ defines a facet of P_3^*.*

Theorem 26. *For all $(i,j) \in A$, the inequality $\beta_{i,j} \leq 1$ defines a facet of P_3^*.*

The validity of Theorems 24-26 has already been observed in Section 2.5.

Theorem 27. *For $|V| \geq 3$, for all $(i,j) \in A$, for all partitions (S,T) of $V \setminus \{i,j\}$, the inequality*

$$\beta_{i,j} \leq \beta_{j,i} + \sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i}$$

defines a facet of P_3^*

Proof. Fix $(i,j) \in A$ and (S,T) a partition of $V \setminus \{i,j\}$. Let F be the face of P_3^* defined as

$$F = \left\{ \beta \in P_3^* : \beta_{i,j} = \beta_{j,i} + \sum_{k \in S} \beta_{j,k} + \sum_{k \in T} \beta_{k,i} \right\}.$$

Suppose that F is included in a hyperplane defined by the equation

$$\sum_{(u,v)\in A} b_{u,v}\beta_{u,v} = b_0 \tag{2.59}$$

and consider the following points $\beta^1, \ldots, \beta^{10}$ which are all in F (we only specify the nonzero components):

1. $\beta^1 = 0$.
2. $\beta^2_{i,j} = \beta^2_{j,i} = 1$.
3. If $|S| \geq 1$, fix $k \in S$, let $\beta^3_{i,j} = \beta^3_{j,k} = \beta^3_{k,i} = 1$, let $\beta^{3'}_{i,j} = \beta^{3'}_{j,k} = \beta^{3'}_{k,i} = \beta^{3'}_{i,k} = 1$, and let $\beta^{3''}_{i,j} = \beta^{3''}_{j,k} = \beta^{3''}_{k,i} = \beta^{3''}_{k,j} = 1$.
4. If $|S| \geq 1$, fix $k \in S$, and let $\beta^4_{i,k} = \beta^4_{k,i} = 1$.
5. If $|T| \geq 1$, let β^5, $\beta^{5'}$ and $\beta^{5''}$ be defined like β^3, $\beta^{3'}$ and $\beta^{3''}$, but with $k \in T$.
6. If $|T| \geq 1$, fix $k \in T$, and let $\beta^6_{j,k} = \beta^6_{k,j} = 1$.
7. If $|S| \geq 2$, fix $h, k \in S$, and let $\beta^7_{i,k} = \beta^7_{k,h} = \beta^7_{h,i} = 1$.
8. If $|T| \geq 2$, fix $h, k \in T$, and let $\beta^8_{i,k} = \beta^8_{k,h} = \beta^8_{h,i} = 1$.
9. If $|S| \geq 1$ and $|T| \geq 1$, fix $k \in S$, $h \in T$, and let $\beta^9_{k,j} = \beta^9_{j,h} = \beta^9_{h,k}$.
10. If $|S| \geq 1$ and $|T| \geq 1$, fix $k \in S$, $h \in T$, and let $\beta^{10}_{k,h} = \beta^{10}_{h,k} = 1$.

By successively substituting the points $\beta^1, \ldots, \beta^{10}$ in 2.59, one can easily conclude that, up to a multiplicative constant, (2.59) is equivalent to the equation defining F. This proves that F is a facet of P^*_3. □

2.7 Conclusions and perspectives

In this chapter, we have introduced the definition of cycle selections and of the associated maximum weigthed cycle selection (**MWCS**) problem. To the best of our knowledge, these concepts had not been explicitly identified earlier, in spite of their rather natural definition and of their relation with fundamental graph theoretic concepts like directed cycles and circulations. We have investigated several properties of cycle selections and of **MWCS**, including their computational complexity, the relation between various integer programming formulations, and the polyhedral structure of the cycle selection polytope.

As explained in Section 2.1.2, Smeulders et al. (2022) have (implicitly) considered cycle selections in order to handle a probabilistic variant of the kidney

exchange problem formulated as a two-stage stochastic integer programming problem. In their experiments, the latter problem turned out to be very difficult to solve. We hope to be able to rely on our improved understanding of cycle selections in order to facilitate the solution of **MWCS** and of the stochastic kidney exchange problem. Some attempts in this direction are provided in the next chapter.

Chapter 3

Cycle selections: numerical experiments

The content of this chapter is based on joint work with Yves Crama.

Contents

3.1 Introduction

In Chapter 2, we have introduced the notion of a *cycle selection* and a related optimization problem, namely, the *maximum weighted cycle selection problem* (**MWCS**). As a reminder, given a (loopless) directed graph $G = (V, A)$, where V is the set of vertices and A is the set of arcs of G, we say that a subset of arcs $B \subseteq A$ is a *cycle selection* in G if the arcs of B form a union (possibly empty) of directed cycles. When a weight $w_{i,j} \in \mathbb{R}$ is assigned to each arc $(i, j) \in A$, **MWCS** aims to identify a cycle selection B which maximizes $w(B) = \sum_{(i,j)\in B} w_{i,j}$.

The goal of this chapter is to test our findings about the cycle selection problem through numerical experiments, especially focusing on different formulations and their respective efficiency. First, we compute in Section 3.2 a maximum weighted cycle selection on randomly generated weighted digraphs. Then, in Section 3.3, we apply our knowledge about cycle selection formulations to the two-stage stochastic KEP problem introduced in Smeulders et al. (2022). In particular, we propose alternative ways to solve the problem using the Benders decomposition resolution procedure.

All numerical experiments presented in this chapter are implemented using C++ 14 as programming language and CPLEX 12.10.0 as generic MILP solver. The tests are performed on a Dell Latitude 7490 running Windows 10 64Bit with an Intel Core i5-7300U CPU and two cores at 2.60GHz and 16 GB of RAM.

3.2 Maximum weighted cycle selections

3.2.1 Formulations and instances

Six formulations are introduced in Chapter 2 to describe cycle selections in a given digraph:

1. the arc (**ARC**) formulation;
2. the simple extended arc (**SEA**) formulation;
3. the extended arc (**EA**) formulation;
4. the modified extended arc (**MEA**) formulation;
5. the position-indexed (**PI**) formulation;
6. the cycle (**CY**) formulation.

They have been previously compared from a theoretical perspective regarding the strength of their linear relaxations. The purpose of this section is now to compare the formulations numerically in terms of resolution efficiency.

We first compare the IP formulations proposed to describe cycle selections by solving **MWCS** on various random weighted digraphs. As observed in

Chapter 2, when all arc weights are non-negative, a maximum cycle selection of a digraph simply consists of all arcs contained in strong components. Therefore, in this case, **MWCS** is solvable in linear time by simply using Tarjan's strong component algorithm (Tarjan, 1972). Hence, we focus on solving **MWCS** on digraphs with positive and negative arc weights. Namely, given three parameters $n \in \mathbb{N}$, $p \in [0, 100]$, $d \in [0, 100]$, we generate random digraphs $G = (V, A)$ with weights $w_{i,j}$, $(i, j) \in A$, such that:

- $n = |V|$;
- for each ordered pair of distinct vertices (i, j), $(i, j) \in A$ with probability $\frac{p}{100}$ independently of the other arcs;
- for each arc $(i, j) \in A$, $w_{i,j}$ is uniformly distributed in $[0, 1]$ with probability $\frac{d}{100}$, and $w_{i,j}$ is uniformly distributed in $[-1, 0]$ with probability $1 - \frac{d}{100}$, independently of the other arcs.

We have conducted computational experiments on randomly generated digraphs for various values of $n \in \{50, 100, 150, 200, 250, 300\}$, $p \in \{10, 20, 40, 90\}$, and $d \in \{10, 80, 90\}$ as illustrated in several figures and tables throughout Section 3.2. For each configuration of n, p and d, we have generated 30 random instances. For a given (n, p, d) configuration, the same instances are used across all the tests performed.

Furthermore, we have conducted additional computational experiments on digraphs generated based on Steiner triple instances as explained in greater detail in Section 3.2.4.

While the formulations SEA, EA, MEA, and PI are compact, the ARC formulation has an exponential number of constraints and the CY formulation has an exponential number of variables. To use the CY formulation, either a branch-and-price process must be implemented to generate the variables gradually, or all the cycles of the digraph under consideration must be initially generated. We implemented the formulation by generating all cycles of the digraph. However, given that the maximum cycle length is not limited in **MWCS**, generating the models took over an hour even for the smallest instances. Since our focus in Chapter 2 is mostly on the ARC formulation, we decided not to pursue the study of the CY formulation. Note however that if the maximum length of cycles is limited, the formulation could become more efficient as illustrated in Subsection 3.2.6. Indeed, despite its potentially exponential number of variables, it remains numerically efficient when the maximum cycle length is limited to 2 or 3, see, e.g., Smeulders et al. (2022).

In order to implement the ARC formulation, a branch and cut procedure has been implemented, as explained in the following section.

3.2.2 Implementation of the ARC formulation

Initial model

The ARC formulation of the maximum weighted cycle selection problem is the following IP model:

$$\max \sum_{(i,j)\in A} w_{i,j}\beta_{i,j} \tag{3.1}$$

$$\text{subject to } \beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.2}$$

$$\beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k} \quad \forall (i,j) \in A, \forall S \subseteq V : i \in S, j \in V \setminus S. \tag{3.3}$$

Because of the exponential number of return inequalities (3.3), it is not advisable to include them all in the model given to the solver. The idea is instead to include part of these inequalities in the initial model and to gradually add return inequalities to the model through a separation procedure. We chose to only include the *predecessor inequalities* (3.4) and the *successor inequalities* (3.5) in the initial formulation. These inequalities are the special cases of the return inequalities (3.3) obtained by setting $S = \{i\}$ or $S = V \setminus \{j\}$, respectively, for each fixed arc (i,j):

$$\beta_{i,j} \leq \sum_{(k,i)\in A} \beta_{k,i} \qquad \forall (i,j) \in A, \tag{3.4}$$

$$\beta_{i,j} \leq \sum_{(j,k)\in A} \beta_{j,k} \qquad \forall (i,j) \in A. \tag{3.5}$$

We denote by P_0 the resulting initial model, that is, the model:

$$\begin{aligned} & \max \sum_{(i,j)\in A} w_{i,j}\beta_{i,j} \\ \text{subject to} \quad & \beta_{i,j} \leq \sum_{(k,i)\in A} \beta_{k,i} \qquad \forall (i,j) \in A \\ & \beta_{i,j} \leq \sum_{(j,k)\in A} \beta_{j,k} \qquad \forall (i,j) \in A \\ & \beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \end{aligned}$$

When the IP solver performing a branch and bound procedure finds a feasible solution of P_0, it is necessary to check whether this solution indeed defines

a cycle selection. For example, for $n \geq 4$, the solution $\beta_{1,2} = \beta_{2,1} = \beta_{3,4} = \beta_{4,3} = \beta_{1,4} = 1$ satisfies the constraints of P_0 but is not a cycle selection as illustrated in Figure 3.1. A separation procedure must be run in order to identify some return inequalities violated by the current solution (if any) and to add them to the model. As explained later in the section, the separation procedure will be implemented only for integer solutions.

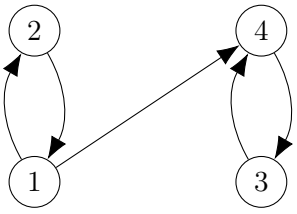


Figure 3.1: Representation of a solution of P_0 which is not a cycle selection

Separation procedure

User-defined separation procedures are called *callbacks* in several popular solvers. For **MWCS**, the callback procedure must identify and add valid *cuts*, that is, all return inequalities violated by the solution associated with the current node. If no violated return inequality is found, the solver continues the classical branch and bound process. On the other hand, if violated return inequalities are identified, the model is solved again by incorporating all previous and newly added inequalities.

Two types of callbacks can be distinguished: USERCUTS callbacks which separate non-integer solutions, and LAZYCUTS callbacks which separate integer solutions.

In our implementation, the separation procedure is implemented through LAZYCUTS callbacks in the following way: each time an integer solution β' is found, Tarjan's algorithm is called on the digraph $G_{B'} = (V, B')$, where $B' = \left\{(i,j) \in A : \beta'_{i,j} = 1\right\}$, in order to identify its strongly connected components (Tarjan, 1972). We say that an arc (i,j) is a *link* of $G_{B'}$ if i and j are in different strongly connected components of $G_{B'}$. Clearly, if B' is a selection, then $G_{B'}$ cannot have any link. So, for each link (i,j), we add the following return inequality to the model:

$$\beta_{i,j} \leq \sum_{(l,k) \in A : l \in V \setminus S, k \in S} \beta_{l,k}, \tag{3.6}$$

where $V \setminus S$ is the set of vertices reachable from j, i.e., the set of vertices $l \in V$ such that there exists a path from j to l in $G_{B'} = (V, B')$. Note that $i \in S$, $j \in V \setminus S$, and that the inequality (3.6) is violated by the solution β' since (i,j) is a link of $G_{B'}$.

3.2.3 Experimental results

In terms of total running time, that is, the time needed to construct the model and to solve it, the ARC formulation and the SEA formulation clearly outperform the other formulations, as illustrated by the performance profiles in Figure 3.2 for the parameter values $n = 50$, $p = 20$, and $d = 80$. Indeed, all of the 30 instances generated with these parameters are solved in less than two seconds when using the formulations ARC and SEA. On the other hand, with the EA formulation, the fastest running time is around 10 seconds, and some instances require up to 22 seconds. With the MEA formulation, the minimum running time is 13 seconds, and some instances require up to 22 seconds. Note that the total time running is divided more or less equally between the time to generate the model and the time to solve it. For example, for the EA formulation, the average generation time is 7.69 seconds, and the average solving time is 4.75 seconds, while for the MEA formulation, the average generation time is 7.55 seconds, and the average solving time is 7.38 seconds. The results for the position-indexed formulation are not displayed, as the running times are too large compared to the others. For example, the fastest running time for the same instances is 156 seconds, and the longest is 667 seconds.

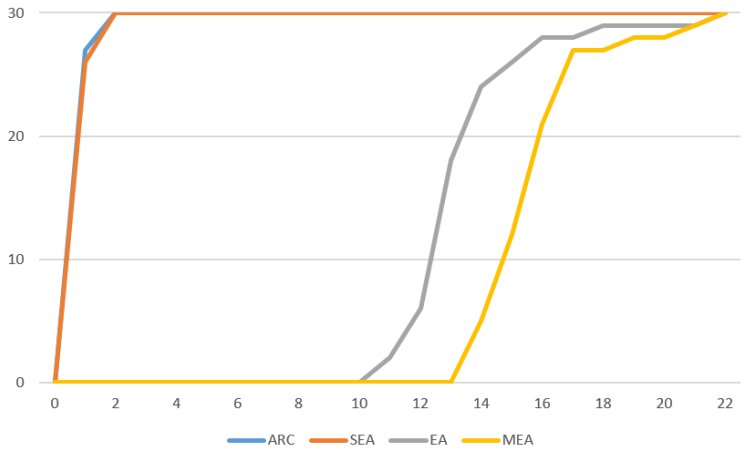


Figure 3.2: Comparison of total running time of the **MWCS** formulations when $n = 50$, $p = 20$ $d = 80$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time, for each formulation.

One of the explanations for these big differences between formulations might be the number of variables and constraints each model has to handle. Table 3.1 displays the potential numbers of variables and constraints for each formulation as well as the average number of variables and constraints for each formulation, given the parameter configuration $n = 50$, $p = 20$, and

$d = 80$. Note that for the ARC formulation, the potential number of constraints is the number of return inequalities, but the average number of constraints reported is based on the initial model P_0, as not all return inequalities are generated.

Table 3.1: Size models with $n = 50$, $p = 20$, and $d = 80$

Method	Variables	Constraints	Average nb of variables	Average nb of constraints
ARC	$\mathcal{O}(m)$	$\mathcal{O}(m\,2^m)$	490.97	981.93 *
SEA	$\mathcal{O}(m)$	$\mathcal{O}(m)$	981.93	1031.93
EA	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	241766.07	266314.40
MEA	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	241766.07	290862.73
PI	$\mathcal{O}(m\,n^2)$	$\mathcal{O}(m\,n^2)$	1227416.67	2382170.27

We also compared the ARC and the SEA formulations on bigger instances and with different p and d parameter values, as shown in Figure 3.3. While the two formulations solve the instances relatively fast, the SEA formulation tends to be faster than the ARC formulation.

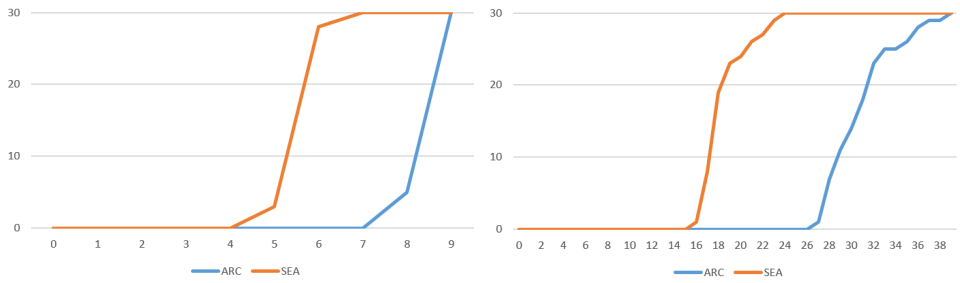


Figure 3.3: Comparison of total running time of the ARC and SEA formulations when $n = 300$, $p = 20$ $d = 80$ (left) and $n = 300$, $p = 40$ $d = 90$ (right). The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time, for each formulation.

Beyond the differences in running times, we observed that for all formulations and for all the instances tested, the solver always immediately finds a maximum weighted selection. By "immediately", we mean that the linear relaxation of each model under consideration has a binary optimal solution which defines a maximum cycle selection. In fact, this observation even holds for the linear relaxation of the incomplete arc formulation P_0, which does not include all return inequalities. More precisely, for all the instances handled with the arc formulation, the solution process goes as follows: At the root node, the solver solves the linear relaxation of P_0. It finds an integer solution and calls the callback procedure to verify that this integer solution defines a selection. This is always the case, so that no cut is ever added. Hence the solution process stops, and no branching is needed.

For all these instances, we further observed that the average optimal value is very close to the expected total weight of the positive arcs, that is,

$$0.5\ n\ (n-1)\ \frac{p}{100}\ \frac{d}{100};$$

see Table 3.2 for an illustration.

Table 3.2: Average optimal value (over 30 instances) and expected weight of the positive arcs for various values of n, $p = 20$, and $d = 80$

n	Average optimal value	Expected total weight of the positive arcs
50	198.56	196
100	789.19	792
150	1789.71	1788
200	3184.24	3184
250	4988.85	4980
350	7171.53	7176

These seemingly intriguing observations can actually be explained by theoretical properties of random graphs. To this effect, consider a random directed graph $G = (V, A)$ where $|V| = n$ and each ordered pair of distinct vertices (i, j) is in A with probability $u(n)$ independently of the other pairs. A result of Graham and Pike (2008), which extends a result of Erdős and Rényi (1959) regarding undirected graphs, implies that when $u(n)$ is "asymptotically larger" than $\frac{\ln n}{n}$, the probability that G is strongly connected tends to 1 as n tends to infinity. (Here, "asymptotically larger" means that $\lim_{n\to\infty} \frac{\ln n}{n\ u(n)} = 0$. We refer to Graham and Pike (2008) for details.)

For our random instances, we can apply this result to the random digraph $G' = (V, A')$, where A' is the subset of arcs with positive weight; so here,

$$u(n) = \frac{p}{100}\ \frac{d}{100}.$$

We obtain that for fixed p, d, and for n large enough, G' is almost surely strongly connected. This means, in particular, that the set A' of positive arcs is (almost surely) a cycle selection, in which case it is necessarily optimal for **MWCS**.

These observations intuitively suggest that in order to generate non trivial instances, we should take values of p and d small enough with respect to the Erdős-Rényi threshold $\frac{\ln n}{n}$. This leads to very sparse digraphs.

With this in mind, random digraphs with parameters $p = 10$ and $d = 10$ have been generated, so that $u = 0.01$ is smaller than $\frac{\ln n}{n}$ for $n \in$

$\{50, 100, 150, 200, 250, 300\}$. **MWCS** has been solved on these digraphs using either the ARC or the SEA formulation. For these sparse instances, the resolution process is not as straightforward as for the earlier ones. Some violated return inequalities are added to the model for the ARC formulation, and some branching occurs for the SEA formulation, especially for the smaller instances with $n \in \{50, 100\}$. Still, the solution process is really fast: all 180 instances are solved in less than 3 seconds with the SEA formulation; as for the ARC formulation, 173 instances are solved in less than 3 seconds, and the seven remaining ones in less than 22 seconds.

3.2.4 Steiner triples

In Chapter 2, the decision version of **MWCS** is proved to be strongly NP-complete by a reduction from the hitting set problem. Based on this proof, we have generated instances of **MWCS** by applying the same reduction to well-known hard instances of the hitting set problem, namely, the so-called Steiner triple instances (Fulkerson et al., 1974).

For the sake of completeness, let us describe the reduction. A Steiner triple instance of the hitting set problem is defined by a finite set X and a collection $T = \{T_1, \ldots, T_r\}$ of subsets of X, with $r = \frac{1}{6}|X|(|X| - 1)$ and $|T_j| = 3$ for $j = 1, \ldots, r$. The set T has a special (Steiner) structure which is not of direct interest here. Given such an instance we construct an associated weighted digraph $G = (V, A)$ as follows: $V = X \cup T \cup \{0\}$, and

- for all $j = 1, \ldots, r$, $(T_j, 0) \in A$ with weight $w(T_j, 0) = r$,
- for all $i \in X$, $(0, i) \in A$ with weight $w(0, i) = -1$,
- for all $j = 1, \ldots, r$ and for all $i \in T_j$, $(i, T_j) \in A$ with weight $w(i, T_j) = 0$.

As follows from the complexity proof of **MWCS** in Section 2.2, the optimal size of a hitting set of T is h^{opt} if and only if the maximum weight of an optimal selection of G is $w^{opt} = r^2 - h^{opt}$. Note that the resulting instances of **MWCS** are quite sparse: they contain $n = |X| + r + 1$ vertices and $|X| + 4r \leq 4n$ arcs.

We tried to solve these instances of **MWCS** using the formulations ARC and SEA for sets X of cardinality ranging from 9 to 243. For values of $|X|$ up to 55, CPLEX was able to terminate in less than three hours of computing time. Table 3.3 provides information about the size of those Steiner instances and the associated digraphs: size of X, number r of triples, number of vertices $|V|$ and number of arcs $|A|$. The middle section of the table displays the size h^{opt} of a minimum hitting set, as well as the optimal value w^{opt} of the maximum weighted cycle selection returned by CPLEX. On the right part of Table 3.3, one can find the total running time in seconds t and the number of nodes explored by the solver for the ARC and SEA formulations. The

instances with $|X|$ equal to 9, 15, 27, 45 are from the online OR Library of J.E. Beasley (http://people.brunel.ac.uk/ mastjjb/jeb/info.html). The remaining instances, with $|X|$ equal to 37, 39, 49, 55, have been additionally generated (https://www.dmgordon.org/cover/).

Table 3.3: Steiner triple instances (part 1)

$\|X\|$	r	$\|V\|$	$\|A\|$	h^{opt}	w^{opt}	t_{ARC}	Nodes_{ARC}	t_{SEA}	Nodes_{SEA}
9	12	22	57	5	139	1	0	0	0
15	35	51	155	9	1216	0	83	1	256
27	117	145	495	18	13671	3	4634	2	3287
37	222	260	925	23	49261	15	8693	111	26465
39	247	287	1027	25	60984	594	528829	133	42215
45	330	376	1365	30	108870	109	68490	627	278419
49	392	442	1617	32	153632	160	74043	1022	220053
51	425	477	1751	34	180591	401	147400	5351	533609
55	495	551	2035	37	244988	823	309300	9513	998079

The ARC formulation generally has a shorter total running time than the SEA formulation, except for a few instances where the opposite is observed. As expected, the instances derived from the Steiner triple instances are more challenging than the random ones, and many nodes are explored in the branch and bound tree. Note that no separation cuts were added during the resolution process for the ARC formulation, for any of the instances.

Three larger instances with $|X|$ in $\{81, 135, 243\}$ have also been tested. The time limit of three hours was reached with both formulations before the solver could find an optimal solution. However, as shown in Table 3.4, the solver was able to produce feasible solutions that are very close to optimality using both formulations. Even more, the ARC formulation quickly found the optimal solution for the instance with $|X| = 81$, although the solver could not prove its optimality within the given time limit.

Table 3.4: Steiner triple instances (part 2)

$\|X\|$	r	$\|V\|$	$\|A\|$	h^{opt}	w^{opt}	w^{best}_{ARC}	w^{best}_{SEA}
81	1080	1162	4401	61	1166339	1166339	1166338
135	3015	3151	12195	103	9090122	9090119	9090118
243	9801	10045	39447	198	96059403	96059392	96059392

3.2.5 MWCS with budget constraint

As we will see in Section 3.3, when solving stochastic kidney exchange problems, it makes sense to impose a budget constraint on the cardinality of a

cycle selection, that is, a constraint of the form:

$$\sum_{(i,j)\in A} \beta_{i,j} \leq b$$

where $b \in \mathbb{N}$. In order to test the difficulty of the resulting problem, we have used the same random weighted digraphs as in Section 3.2.1 with various values for the budget b.

From our previous observations in Section 3.2.3, we know that if b is too large, then the set of b arcs with the largest positive weights is almost surely strongly connected and hence defines a cycle selection (that is, the optimal solution is obtained by setting $\beta_{i,j} = 1$ for those b arcs with the highest positive weight and $\beta_{i,j} = 0$ otherwise). Such instances are excessively easy to solve. Therefore, as a rule of thumb, we choose budget values such that $\dfrac{b}{n(n-1)}$ is small with respect to the Erdős-Rényi threshold $\dfrac{\ln n}{n}$.

Table 3.5: MWCS with budget constraint: results of computational experiments

ARC	time			nodes		cuts		I
n=200	av	min	max	av	max	av	max	
$(20, 80, 100)$	73.06	2	TL	559.00	7115	3398.90	44358	3
$(40, 90, 100)$	138.10	6	TL	285.53	2304	1524.77	15105	3
$(20, 80, 50)$	39.93	2	592	371.07	7322	417.23	6397	4
$(40, 90, 50)$	73.03	5	TL	739.53	18457	744.47	16392	8
SEA	time			nodes				I
n=200	av	min	max	av	max			
$(20, 80, 100)$	27.87	6	62	1327.53	4752			3
$(40, 90, 100)$	84.43	25	303	1620.37	6387			7
$(20, 80, 50)$	39.63	13	85	2342.83	11521			0
$(40, 90, 50)$	59.63	30	131	1321.63	4994			9

Table 3.5 synthesizes some data regarding our computational experiments on **MWCS** with a budget constraint. The top part of the table refers to the ARC formulation and the bottom part refers to SEA formulation. Each line refers to a fixed configuration of parameter values (p, d, b), and to the solution of 30 instances generated with these parameters. Each line displays:

- The average, minimum and maximum total running time (in seconds) over 30 instances. A time limit of 10 minutes was given to the solver. If the maximum total running time is indicated as TL, it means that a least one instance reached the time limit.
- The average and maximum number of nodes explored during the solution process (the minimum is 0 for all configurations).
- For the ARC formulation only, the average and maximum number of separation cuts added during the solution process (the minimum is 0 for all configurations).

- The number I of instances for which the optimal solution of the linear relaxation is binary, and hence is optimal for **MWCS**.

One can observe that, whatever formulation is used, **MWCS** becomes harder to solve when a budget constraint is imposed on the number of selected arcs. With the ARC formulation, during the solution process cuts are added, and branching occurs for most of the instances. With the SEA formulation, branching also occurs for most instances. The average total running time is higher for the ARC formulation. However, Figure 3.4 shows that the ARC formulation is faster for at least two-thirds of the instances but takes more time for the last third of the instances. Some of the instances are not solved in 10 minutes. By way of comparison, for **MWCS** *without* a budget constraint:

- the instances with parameter values $n = 200$, $p = 20$, $d = 80$ are all solved in less than 3 seconds with the ARC formulation and in less than 2 seconds with the SEA formulation;
- the instances with parameter values $n = 200$, $p = 40$, $d = 90$ are all solved in less than 7 seconds with the ARC formulation and in less than 4 seconds with the SEA formulation.

.

3.2.6 MWCS with maximum cycle length constraint

Kidney exchange programs usually impose a maximum length, say K, on the cyclic exchanges that they allow. Accordingly, in this section, we consider a constrained version of cycle selections, and we say that a subset $B \subseteq A$ is a K-cycle selection if the arcs of B form a union of directed cycles, each of length at most K. We won't go very deep into investigating the corresponding constrained version of **MWCS**, which may be a topic for future work, but we report here a few observations about it.

Whereas it was easy to incorporate a budget constraint into all the **MWCS** formulations, imposing a constraint on the maximum cycle length requires more work. The PI and CY formulations of **MWCS** can be easily adapted to account for such a constraint. Indeed, the original formulation proposed in Smeulders et al. (2022), and from which the PI formulation is deduced, imposes a maximum cycle length. For the CY formulation, it suffices to generate variables for cycles of length at most K only.

On the other hand, arc-based formulations are more difficult to deal with when K is arbitrary see Section 2.5. In Section 2.6, we proposed an arc formulation, denoted ARC3, which describes cycle selections with maximum cycle length at most 3. To test this formulation, we compared it numerically with the PI and the CY formulations with maximum cycle length equal to

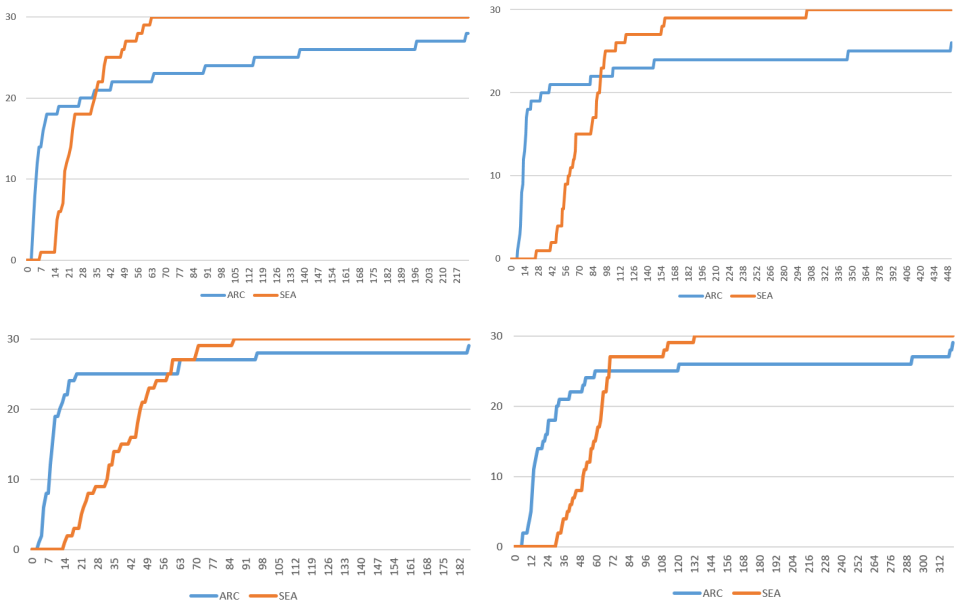


Figure 3.4: Comparison of total running time of the ARC and SEA formulations when $n = 200$, $p = 20$, $d = 80$, $b = 100$ (top left), $n = 200$, $p = 20$, $d = 80$, $b = 50$ (bottom left), $n = 200$, $p = 40$, $d = 90$, $b = 100$ (top right), and $n = 200$, $p = 40$, $d = 90$, $b = 50$ (bottom right). The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

3 as well, denoted as PI3 and CY3, respectively.

Figure 3.5 displays the performance profile of the three formulations for the instances with parameter values $n = 100$, $p = 20$, $d = 80$. It is clear that ARC3 and CY3 are more computationally efficient than PI3: the CY3 formulation solves all 30 instances in less than a second, whereas the ARC3 formulation takes 8 seconds for some instances, and the PI3 formulation requires 95 to 130 seconds, depending on the instance. Regarding the optimization process, the optimal solution of the linear relaxation of the PI3 and CY3 formulations is binary for all instances, so that no branching is performed. For the ARC3 formulation, no separation cuts are added, but an average of 116.53 nodes are explored before obtaining an optimal solution.

A budget constraint can be incorporated into the formulations in addition to the maximum cycle length constraint. For the same parameter values as above, that is, $n = 100$, $p = 20$, $d = 80$, $K = 3$, and $b = 50$, Figure 3.6 displays the performance profiles of the three formulations with the budget constraint and Figure 3.7 compares the formulations with and without the budget constraint. Surprisingly, a budget constraint does not make the problem significantly harder in terms of running time, especially when using

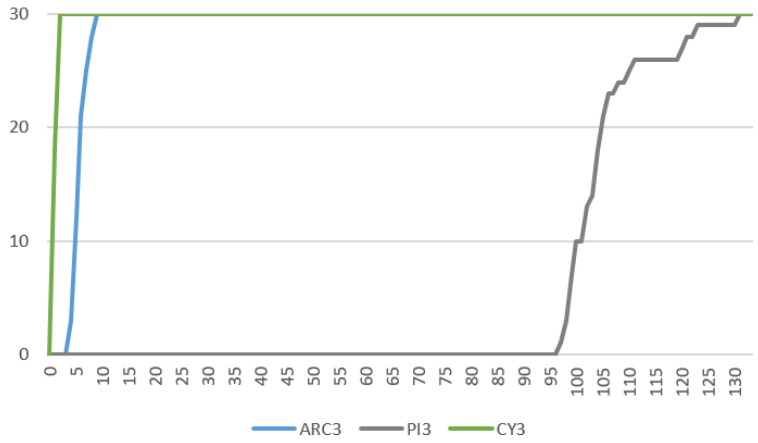


Figure 3.5: Comparison of total running time of the **MWCS** formulations when $n = 50$, $p = 20$ $d = 80$, $K = 3$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

the ARC3 and CY3 formulations. The optimal binary solution is found by solving the linear relaxation of the PI3 formulation for all the instances and the linear relaxation of the CY3 formulation for 27 instances. For the three remaining instances, a few nodes are explored to find the optimal binary solution. For the ARC3 formulation, an average of 142.56 separation cuts are added to the model, and an average of 335.8 nodes are explored before obtaining an optimal solution. Similar observations are made when $b = 25$.

Finally, since the digraphs associated with the Steiner triple instances have cycles of length 3 only, we also evaluated the ARC3 and CY3 formulations on these digraphs. Table 3.6 presents a comparison of the total running times of the ARC, SEA, ARC3, and CY3 formulations for the instances that were solved within three hours. Although the ARC and ARC3 formulations have similar total running times, the most efficient formulation in terms of total running time is CY3.

3.3 Cycle selections in stochastic kidney exchange models

We were initially motivated to study cycle selections due to optimization challenges that emerge in connection with kidney exchange programs. In particular, Smeulders et al. (2022) identify the problem of selecting a subset of arcs of the compatibility digraph $G = (V, A)$ that should undergo further complex compatibility tests (so-called crossmatch tests), so as to maximize the excepted number of transplants. They formulate this prob-

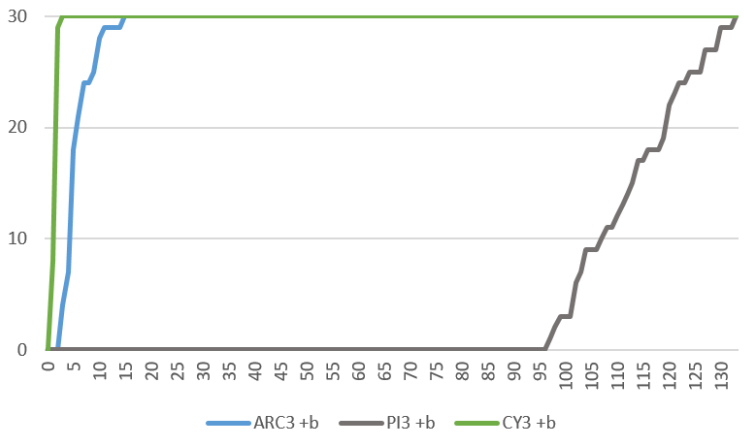


Figure 3.6: Comparison of total running time of the **MWCS** formulations when $n = 50$, $p = 20$ $d = 80$, $K = 3$ and $b = 50$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

lem as a two-stage stochastic optimization problem which, given a *testing budget* b, identifies (in stage 1) a subset of arcs $B \subseteq A$ with $|B| \leq b$ such that the expected number of transplants in the digraph (V, B) (in stage 2) is maximized. (Note that in Smeulders et al. (2022), b is not viewed as expressing a financial limitation but rather a practical one linked to the amount of medical resources required to carry out the crossmatch tests.) Smeulders et al. (2022) propose several integer programming formulations for this problem. They show, among other solution approaches, how to apply Benders decomposition to this *two-stage KEP problem*. Furthermore, they tighten the formulation of the problem by imposing valid constraints which ensure that the set B is a cycle selection. Indeed, arcs that are not contained in

Table 3.6: Comparison of total running time on the digraphs associated with the Steiner triple instances, in seconds

|X|	ARC	SEA	ARC3	CY3
9	1	0	0	0
15	0	1	0	1
27	3	2	8	7
37	15	111	24	6
39	594	133	28	12
45	109	627	256	156
49	160	1022	175	113
51	401	5351	322	185
55	823	9513	812	486

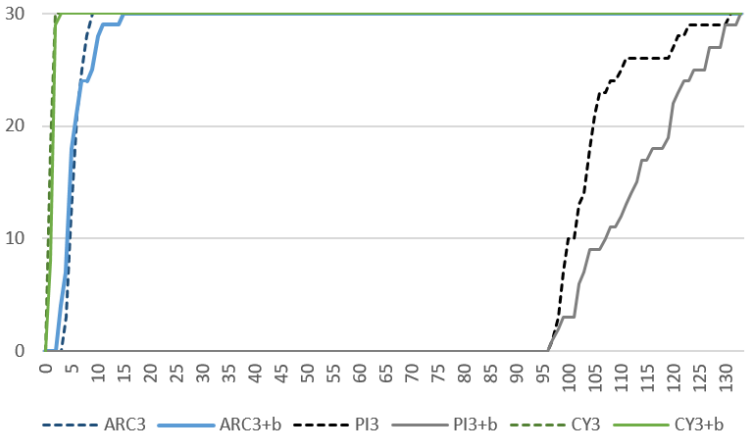


Figure 3.7: Comparison of total running time of the **MWCS** formulations when $n = 50$, $p = 20$ $d = 80$, $K = 3$ without a budget constraint and with a budget constraint (+b) when $b = 50$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

any cycle cannot be used in an exchange and hence, should not be selected in the first stage. The formulation of these constraints is inspired by the position-indexed edge formulation of the kidney exchange problem proposed by Dickerson et al. (2016).

The work of Smeulders et al. (2022) sparked our interest to develop a better understanding of the cycle selection problem and to explore its ILP formulations. In this section, we experiment with solution approaches for the two-stage stochastic kidney exchange problem based again on Benders decomposition, but relying now on the ARC or SEA formulations of the cycle selection problem to tighten the original model

Remark 4. Note that Smeulders et al. (2022) refer to their two-stage stochastic optimization problem as the *selection* problem, which should not be confused with the *cycle selection* problem studied in this thesis. To avoid possible confusion, we call *two-stage KEP problem* the problem introduced in Smeulders et al. (2022).

3.3.1 Models

Let us now present some integer programming formulations of the two-stage KEP problem.

Two-stage KEP problem

Given a directed graph $G = (V, A)$, we denote by $\mathcal{S} = \{A_1, .., A_m\}$ the collection of all subsets of arcs (where $m = 2^{|A|}$). Each subset $A_s \in \mathcal{S}$ is

a possible scenario, and we denote by q_s the probability of occurrence of scenario A_s (we will later denote scenario A_s simply by the index s). To model the two-stage KEP problem, Smeulders et al. (2022) introduce:

- first-stage binary variables $\beta_{i,j}$ for each arc $(i,j) \in A$ with the interpretation that $\beta_{i,j} = 1$ if and only if arc (i,j) is selected to undergo a crossmatch test;
- second-stage binary variables $x_s \in \{0,1\}^{N_s}$ for all $s \in \mathcal{S}$, called *scenario variables* (where N_s is a vector of variables of appropriate length). The scenario variables will describe the kidney exchange solution to be adopted under scenario s.

Let $G_{s,\beta} = (V, A_s \cap B(\beta))$, where $B(\beta) = \{(i,j) \in A : \beta_{i,j} = 1\}$. (We sometimes write B instead of $B(\beta)$, for simplicity.) That is, $G_{s,\beta}$ is the graph associated with the arcs that have been selected in stage 1 and that pass the crossmatch tests in scenario s. The scenario variables define a solution of the kidney exchange problem on the graph $G_{s,\beta}$. The generic IP formulation of the two-stage KEP problem is the following:

$$\max \sum_{s \in \mathcal{S}} q_s \sum_{l=1}^{N_s} a_{s,l} x_{s,l} \tag{3.7}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq b \tag{3.8}$$

$$x_s \in P(G_{s,\beta}) \qquad \forall s \in \mathcal{S} \tag{3.9}$$

$$x_s \in \{0,1\}^{N_s} \qquad \forall s \in \mathcal{S} \tag{3.10}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.11}$$

where $P(G_{s,\beta})$ is a polyhedron described by a finite list of inequalities which defines the feasible solutions of the KEP problem on $G_{s,\beta}$. Any formulation of the KEP problem can be used in this generic two-stage KEP problem.

The objective function expresses the expected value of a solution, where the coefficients $a_{s,l}$ are used to measure the quality of the exchange $x_{s,l}$ (e.g., the number of transplants associated with the exchange). Constraint (3.8) ensures that the budget constraint is respected, while constraints (3.9) ensure that a solution x_s defines a feasible exchange in $G_{s,\beta}$. The exact formulation of these constraints depends on the formulation used for the deterministic KEP problem (for example, the position indexed edge formulation (Dickerson et al., 2016), or the cycle formulation (Roth et al., 2004)).

Relaxed restricted two-stage KEP problem

Because of the large number of possible scenarios, Smeulders et al. (2022) restrict their attention to a subset of scenarios $S \subset \mathcal{S}$ and define $q_s = \frac{1}{|S|}$ for

each scenario $s \in S$. Furthermore, motivated by previous observations by Dickerson et al. (2016), they also relax the scenario variables to continuous values. They confirmed experimentally that this relaxation has a negligible impact on the optimal value of the $\beta_{i,j}$ variables. Moreover, from a practical point of view, the value of the β variables is the only one that really matters in stage 1 (the optimal set of exchanges can be recomputed after crossmatching). Thus, the generic *relaxed restricted two-stage KEP problem*, or RR-2-KEP problem, is the following :

$$\max \sum_{s \in S} \frac{1}{|S|} \sum_{l=1}^{N_s} a_{s,l} x_{s,l} \tag{3.12}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq b \tag{3.13}$$

$$x_s \in P(G_{s,\beta}) \qquad \forall s \in S \tag{3.14}$$

$$x_s \in [0,1]^{N_s} \qquad \forall s \in S \tag{3.15}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{3.16}$$

We will focus on this mixed-integer programming problem for the remainder of the section.

3.3.2 Optimization methods

The RR-2-KEP problem can be solved by several classical optimization methods such as branch-and-cut for example. One of the approaches proposed in Smeulders et al. (2022) is to use Benders decomposition, which is rather natural since the scenario variables are relaxed. When applying Benders decomposition to the RR-2-KEP problem, the following master and slave problems are obtained.

Master problem:

$$\max \frac{1}{|S|} \sum_{s \in S} z_s(\beta) \tag{3.17}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq b \tag{3.18}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{3.19}$$

Slave problem for each $s \in S$:

$$z_s(\beta) = \max \sum_{l=1}^{N_s} a_{s,l} x_{s,l} \tag{3.20}$$

$$\text{subject to } x_s \in P(G_{s,\beta}) \tag{3.21}$$

$$0 \leq x_{s,l} \leq 1 \qquad \text{for } l = 1, \ldots, N_s. \tag{3.22}$$

Smeulders et al. (2022) observed that, in an optimal solution of the RR-2-KEP problem, each selected arc can be assumed to be part of a directed cycle: in other words, one can assume that B is a cycle selection. In the formulation (3.17)-(3.19) of the master problem, however, this information is lost. In order to strengthen the master problem, Smeulders et al. (2022) add constraints that enforce that each selected arc is part of a cycle. These constraints are inspired by the position-indexed edge formulation of the kidney exchange problem proposed by Dickerson et al. (2016). We refer to Smeulders et al. (2022) for details.

Another way to strengthen the master problem is to add to its formulation any of the valid inequalities introduced in Chapter 2, such as the *return inequalities* of the ARC formulation or the constraints of the SEA formulation.

Accordingly, we compare in this section four different methods for the solution of the RR-2-KEP problem, where each method consists of a formulation and an associated optimization algorithm. Namely, we consider:

- **Method 1**: original MILP formulation of the RR-2-KEP problem (3.12)-(3.16) solved by branch-and-cut.
- **Method 2**: MILP formulation of the RR-2-KEP problem (3.12)-(3.16) together with the additional constraints proposed in Smeulders et al. (2022), solved by Benders decomposition.
- **Method 3**: MILP formulation of the RR-2-KEP problem (3.12)-(3.16) together with the additional variables $x \in \mathbb{R}_+^{|A|}$ and constraints (3.23)-(3.25) of the SEA formulation:

$$x_{i,j} \leq m\, \beta_{i,j} \qquad \forall (i,j) \in A \tag{3.23}$$

$$\beta_{i,j} \leq x_{i,j} \qquad \forall (i,j) \in A \tag{3.24}$$

$$\sum_{h:(h,k)\in A} x_{h,k} = \sum_{h:(k,h)\in A} x_{k,h} \qquad \forall k \in V, \tag{3.25}$$

  solved by Benders decomposition.
- **Method 4**: MILP formulation of the RR-2-KEP problem (3.12)-(3.16) together with the return inequalities (3.26):

$$\beta_{i,j} \leq \sum_{(l,k)\in A: l\in V\setminus S, k\in S} \beta_{l,k} \quad \forall (i,j) \in A, \forall S \subseteq V : i \in S, j \in V \setminus S, \tag{3.26}$$

  solved by Benders decomposition.

In our implementations, we use the cycle formulation of Roth et al. (2004) to describe $P(G_{s,\beta})$ in constraints (3.9) and (3.21). In the first three methods, the complete MILP formulations are given to the CPLEX solver (they are compact for fixed K), and the Benders procedure functionality of CPLEX is

activated for Method 2 and Method 3. It cannot be activated with Method 4 due to the exponential number of return inequalities. Instead, these valid inequalities must be gradually added to the model through a separation procedure, as explained in Section 3.2.2. To implement this procedure, callbacks are necessary. However, callbacks are incompatible with the Benders functionality in CPLEX (or Gurobi). Therefore, in order to test Method 4, we had to implement the Benders decomposition procedure "manually" while utilizing callbacks which is possible thanks to the use of callbacks; The next Section 3.3.3 clarifies the steps of the Benders optimization procedure of Method 4.

3.3.3 Implementation of Benders decomposition in Method 4

In our implementation of Method 4, the initial model (P) given to CPLEX is the master problem of the Benders decomposition, that is:

$$\max \frac{1}{|S|} \sum_{s \in S} z_s \tag{3.27}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq b \tag{3.28}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.29}$$

$$z_s \in \mathbb{R}^+ \qquad \forall s \in S. \tag{3.30}$$

Then, CPLEX starts a branch-and-cut procedure. At each node:

1. CPLEX solves the linear relaxation of the model associated with the node. It outputs a solution (z', β'). If the solution β' is fractional, the solver follows the classical branch-and-bound process. If β' is integer, it goes to the next step. (Note that the objective function is initially unbounded and that the value of the β variables is very loosely constrained in (P). We return to these weaknesses in Section 3.3.5 hereunder.)

2. **Cycle selection separation procedure.** The solver checks if β' defines a cycle selection (that is, if all return inequalities are satisfied) as described in Section 3.2.2. If the solution does not satisfy all return inequalities, the violated inequalities (3.6) are added to the model associated with the current node, and CPLEX solves the modified model (back to step 1). If all return inequalities are satisfied, the next step is taken.

3. **Benders procedure.** Given the optimal solution (z', β') of the model associated with the current node, CPLEX solves the following slave problem for all $s \in S$, where C_s denotes the set of eligible cycles in the

digraph $G_s = (V, A_s)$ and w_c is the length of cycle c:

$$\max \sum_{c \in C_s} w_c x_{c,s} \tag{3.31}$$

$$\text{subject to} \sum_{c \in C_s : (i,j) \in A(c)} x_{c,s} \leq \beta'_{i,j} \qquad \forall (i,j) \in A_s \tag{3.32}$$

$$\sum_{c \in C_s : v \in V(c)} x_{c,s} \leq 1 \qquad \forall v \in V \tag{3.33}$$

$$x_{c,s} \geq 0 \qquad \forall c \in C_s. \tag{3.34}$$

If $x^*_{c,s}$ ($c \in C_s$) is the optimal solution of the slave problem associated with scenario s, and $z'_s \geq \sum_{c \in C_s} w_c x^*_{c,s}$, then the following Benders cut is added to the master problem:

$$z_s \leq \sum_{(i,j) \in A_s} \beta_{i,j} r^*_{i,j} + \sum_{v \in V} y^*_v, \tag{3.35}$$

where $r^*_{i,j}$, $(i,j) \in A_s$, and y^*_v, $v \in V$, is the optimal solution of the dual of the s-th slave problem. Once all the slave problems have been solved, if at least one Benders cut (3.35) has been added to the master problem, the solver solves again the problem associated with the current node (back to step 1). Note that all inequalities added during the course of steps 2 and 3 remain part of the master problem formulation until the end of the branch-and-cut process. If no inequality (3.35) has been added, (z', β') is a feasible solution of the RR-2-KEP problem, and the solver closes the node and continues with the classical branching process.

In the solution method described above, the cycle selection separation procedure and the Benders procedure are only applied to integer solutions of the problem (3.27)-(3.30). That is, we only use LAZYCUT callbacks. As explained later in Section 3.3.5, the Benders procedure could also be applied to fractional solutions.

3.3.4 Initial experimental results

We have adapted the original code of Smeulders et al. (2022) to implement the four methods discussed in the previous sections, and we have tested them on the same instances as these authors. More precisely, the compatibility graphs have been generated by the random generator described in Saidman et al. (2006). Three different graph sizes are considered in Smeulders et al. (2022), with respectively $n = 25$, 50 and 100 patient-donor pairs, but we will almost exclusively focus here on the small instances (n=25) as very few of the larger ones can be solved to optimality. For each value of n, there

are 40 graphs, divided into four groups of 10 graphs, with vertex success rate p respectively equal to 0.8, 0.6, 0.4, and 0.2 in these four groups (p is the probability that a patient-donor pair successfully passes the crossmatch test). The maximum cycle length is either $K = 3$ or $K = 4$, depending on the experiment.

In this section, we briefly discuss the results of our initial experiments before we turn to some enhancements of Method 4 in the next section. These initial experiments were conducted on the 40 instances with parameter values $n = 25$, $|S| = 100$, $b = 10$, $K = 3$ and $p \in \{0.2, 0.4, 0.6, 0.8\}$. Over all of our experiments, for parameter values $n, |S|, b$ and K fixed, we tested all possible values of vertex success p. Moreover, for each instance solved, the scenarios are exactly the same for all four methods.

Figure 3.8 displays the performance profiles of the running time for the four methods; here, the value on the vertical axis represents the number of instances solved as a function of the running time (in seconds) indicated on the horizontal axis. We observe that the 40 instances are solved in less than 23 seconds with Method 1, less than 6 seconds with Method 2 and less than 48 seconds with Method 3. Only 36 instances are solved under 273 seconds with Method 4, one in 633 seconds, and three instances were not solved at optimality after 20 minutes, which was the time limit given to the solver. These first results tend to show that Method 2 performs just a bit faster than Method 1, which is in line with the observations in Smeulders et al. (2022). Method 3 performs slower than the first two methods, but the difference is quite small. On the other hand, Method 4 is much worse than the first three methods. In view of these observations, several attempts were conducted to enhance the efficiency of Method 4, as reported in the next section.

3.3.5 Enhancements of the implementation of Method 4

Additional constraints for the master problem.

The valid constraints (3.36), (3.37) and (3.38) hereunder can be added to the master model in order to tighten its initial formulation:

$$z_s \leq \sum_{(i,j)\in A_s} \beta_{i,j} \qquad \forall s \in S, \tag{3.36}$$

$$\beta_{i,j} \leq \sum_{(k,i)\in A} \beta_{k,i} \qquad \forall (i,j) \in A, \tag{3.37}$$

$$\beta_{i,j} \leq \sum_{(j,k)\in A} \beta_{j,k} \qquad \forall (i,j) \in A. \tag{3.38}$$

Constraints (3.36) enforce that for all $s \in S$, z_s, which represents the objective value of the slave problem, cannot exceed the number of selected arcs

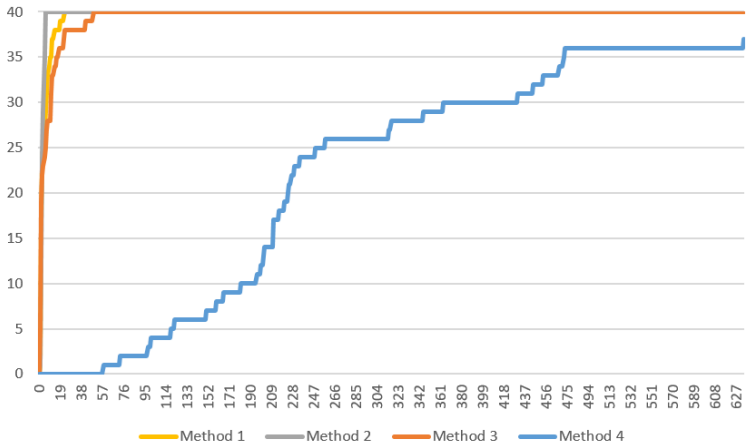


Figure 3.8: Comparison of four methods when $n = 25$, $|S| = 100$, $b = 10$, $K = 3$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

in the graph $G = (V, A_s)$. Constraints (3.37) and (3.38) are the predecessor and successor inequalities already singled out in Section 3.2.2.

These additional constraints made Method 4 perform faster than initially. Nevertheless, as shown in Figure 3.9, the difference in solution time between the methods is still considerable. The performance profile of the initial formulation is in blue, and the profile of the formulation with the additional constraints (3.36)-(3.38) in the master problem (Method 4 v2) is in green. In all subsequent computational experiments, we kept the additional constraints (3.36)-(3.38) in the master problem (P) of the Benders Method 4; henceforth, we will simply refer to this new method as Method 4, for short.

Initial heuristic solution.

The first results showed that when using Method 4, the solver sometimes takes a significant amount of time to find a first feasible solution and hence, a lower bound on the optimal value. To palliate this difficulty, we have injected a first feasible solution (β^{init}, z^{init}) into the model before the start of the Benders procedure. We have tried different heuristics to generate a feasible solution, but most of them were too time-consuming and did not have a significant impact on the total running time of the method. Ultimately, we chose to stick with the heuristic procedure that requires the smallest amount of time. It works as follows:

1. For each scenario s such that $|A_s| \geq 1.05\, p\, |A|$, that is, for each scenario with a number of arcs slightly (5%) higher than the expected number

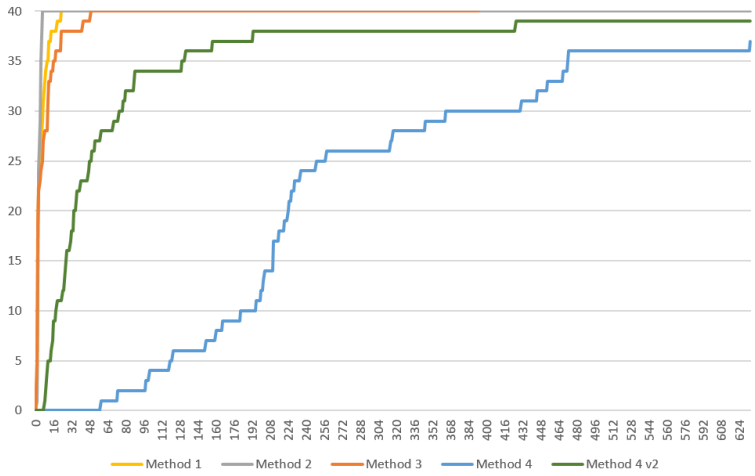


Figure 3.9: Comparison of five methods when $n = 25$, $|S| = 100$, $b = 10$, $K = 3$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

of arcs in $G_s = (V, A_s)$, we solve the following problem:

$$\max \sum_{c \in C_s} w_c x_{c,s} \tag{3.39}$$

$$\text{subject to} \sum_{c \in C_s : (i,j) \in A(c)} x_{c,s} \leq \beta^s_{i,j} \qquad \forall (i,j) \in A \tag{3.40}$$

$$\sum_{c \in C_s : v \in V(c)} x_{c,s} \leq 1 \qquad \forall v \in V \tag{3.41}$$

$$\sum_{(i,j) \in A} \beta^s_{i,j} \leq b \tag{3.42}$$

$$x_{c,s} \geq 0 \qquad \forall c \in C_s \tag{3.43}$$

$$\beta^s_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A_s \tag{3.44}$$

$$\beta^s_{i,j} = 0 \qquad \forall (i,j) \in A \setminus A_s. \tag{3.45}$$

In words, a solution of (3.39)-(3.45) identifies a subset of at most b arcs of A_s which yields the largest possible number of transplants. Then, we retain the scenario s with the highest optimal value of (3.39) and we denote the associated optimal solution $\beta^s \in \{0,1\}^{|A|}$ as $\beta^{init} \in \{0,1\}^{|A|}$.

2. For each scenario $s \in S$, we denote as z^{init}_s the optimal value of the

following problem (similar to the slave problem (3.31)-(3.34)):

$$\begin{aligned} \max \sum_{c \in C_s} & w_c x_{c,s} & & \\ \text{such that : } \sum_{c \in C_s : (i,j) \in A(c)} & x_{c,s} \leq \beta_{i,j}^{init} & & \forall (i,j) \in A \\ \sum_{c \in C_s : v \in V(c)} & x_{c,s} \leq 1 & & \forall v \in V \\ & x_{c,s} \geq 0 & & \forall c \in C_s. \end{aligned}$$

The heuristic feasible solution of (P) given to CPLEX is then $\beta_{i,j}^{init}$ for all $(i,j) \in A$ and z_s^{init} for all $s \in S$.

On the instances with parameter values $n = 25$, $|S| = 100$, $b = 10$, and $K = 3$, the gap between the objective value of the heuristic solution and the optimal value lies between 2% and 82% with a mean of 38%. The running time of Method 4, with or without the initial heuristic solution, is more or less the same as shown in Figure 3.10. (The performance profile of Method 4 is in green without the initial solution, and in purple with the initial solution.). Hence, the efficiency in terms of total running time of injecting an initial solution into the solver is unclear and might need to be investigated on larger instances. However, for the 40 instances with $n = 50$, $S = 50$, $b = 10$, $K = 3$, the time limit of 20 minutes is reached in all cases with Method 4 and by more than half of the instances with the first three methods.

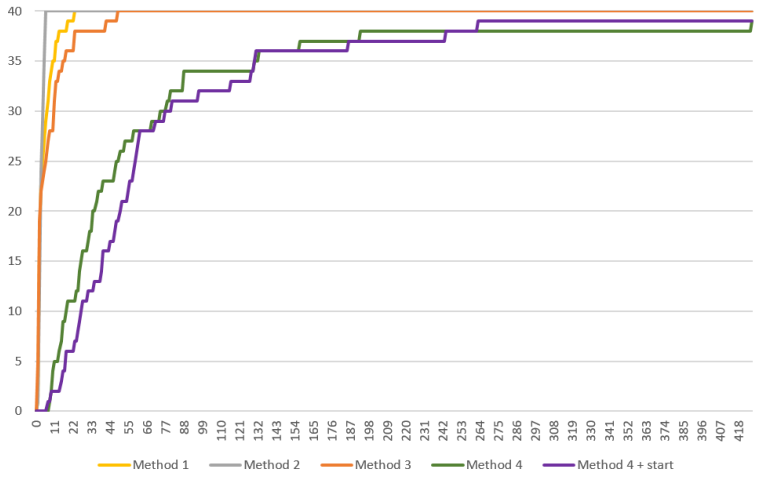


Figure 3.10: Comparison of five methods when $n = 25$, $|S| = 100$, $b = 10$, $K = 3$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

Sample of scenarios for the Benders procedure.

In the Benders procedure, instead of solving the slave problem for each scenario in S, we can first solve it for a sample of scenarios $S_1 \subset S$. If a violated cut is generated for at least one scenario in the sample, then it is added to the master problem and the model is solved again without considering the scenarios outside of the sample. If no violated cut is identified for the sample S_1, then we proceed with another sample S_2 of scenarios, until either a cut is added to the master problem, or all slave problems have been solved. This approach has been implemented, but it did not improve the running time of Method 4.

Benders procedure at each node.

While so far, the separation procedure and the Benders procedure have been applied solely to integer solutions of the master problem during the branch-and-bound procedure (as explained in detail in Section 3.3.3), another possible approach is to apply the Benders procedure at each node. That is, each time the solution of the linear relaxation of the master problem is fractional (in Step 1 of Section 3.3.3), we can apply the Benders procedure (Step 3 in Section 3.3.3) to this solution instead of following the classical branch-and-bound process. We have implemented this procedure by using callbacks on fractional solutions, that is, USERCUTS callbacks, in addition to the LAZYCUTS callbacks on integer solutions. In our experiments, however, this modification does not improve the total running time, as demonstrated by Figure 3.11 where Method 4* refers to Method 4 with activated USERCUTS callbacks.

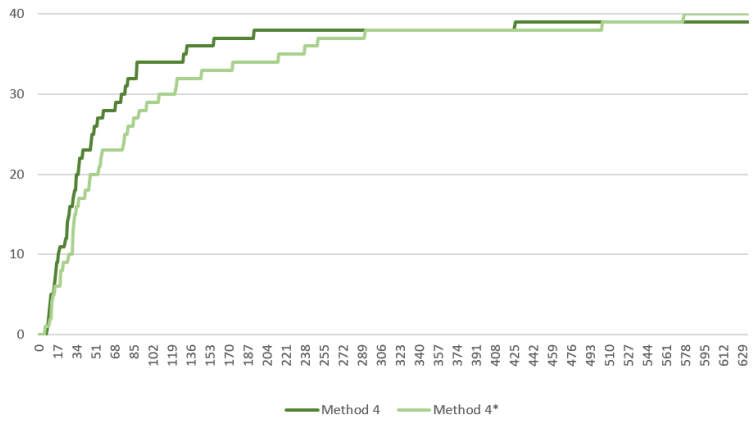


Figure 3.11: Comparison of running time when $n = 25$, $|S| = 100$, $b = 10$, $K = 3$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

IN-OUT procedure.

As a final adaptation of Method 4, an IN-OUT procedure based on a proposal of Ben-Ameur and Neto (2007) was implemented. It is applied to fractional solutions only. In the Benders procedure, instead of using the optimal solution (z', β') of the master problem as parameters of the slave problems and of the Benders cuts (Step 3 in Section 3.3.3), we use the parameters (z^{sep}, β^{sep}) defined as

$$\begin{aligned} \beta_{i,j}^{sep} &= \alpha\beta_{i,j}^{out} + (1-\alpha)\beta_{i,j}^{in} && \forall (i,j) \in A \\ z_s^{sep} &= \alpha z_s^{out} + (1-\alpha) z_s^{in} && \forall s \in S \end{aligned}$$

where

- $(z^{out}, \beta^{out}) = (z', \beta')$ is the optimal solution of the master problem of the current node,
- (z^{in}, β^{in}) is a feasible solution of the RR-2-KEP problem, and
- $\alpha \in [0, 1]$.

Figure 3.12 illustrates the idea, with R (represented by a rectangle) the relaxed feasible region of the master problem of the current node and D (represented by an oval) the relaxed feasible region of the relaxed restricted two-stage KEP problem.

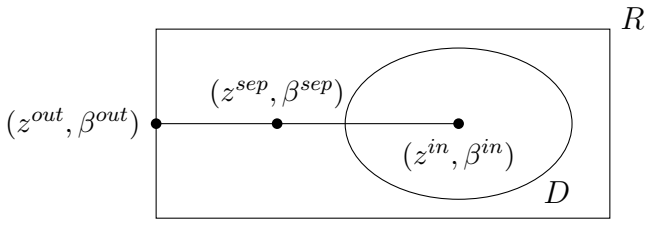


Figure 3.12: Illustration of the IN-OUT procedure

If cuts are found through the slave problems with parameters (z^{sep}, β^{sep}) (i.e., if (z^{sep}, β^{sep}) is in $R \setminus D$) then they are added to the master problem; otherwise (i.e., if (z^{sep}, β^{sep}) is in D), we replace (z^{sep}, β^{sep}) by (z^{out}, β^{out}) and solve again the slave problems as in the classical method.

We tested different options to define (z^{in}, β^{in}) such as:

- best integer solution found by the solver so far,
- random feasible solution,
- linear combination of the best incumbent and a random feasible solution,

and we used different values for α as well. However, quite surprisingly, the use of this method considerably increased the total running time of Method 4: it multiplied this time by a factor 3 on average.

Synthesis of the enhancement attempts.

Overall, in spite of the attempts reported in the previous subsections, no big improvement could be brought to Method 4. Table 3.7 summarizes the information regarding the solution process for five methods on 40 instances with $n = 25$, $|S| = 100$, $b = 10$ and $K = 3$. The table shows:

- The average, minimum and maximum total running time (in seconds). A time limit of 20 minutes was given to the solver. If the maximum total running time is indicated as TL, it means that a least one instance reached the time limit.
- The average number of variables and the average number of constraints for each method. Note that for Method 4 and for Method 4*, we display the average number of constraints in the initial master problem, which does not include all return inequalities (3.26).
- The average, minimum and maximum number of nodes explored during the optimization process.
- For Method 4 and Method 4* only, the average, minimum and maximum number of separation cuts added during the optimization process.

Table 3.7: Comparison of different methods for the RR-2-KEP problem

	time			var.	const.	nodes		
	av	min	max	av	av	av	min	max
M1	6.83	0	42	1533.75	18101.00	15.33	0	139
M2	3.88	1	11	1660.25	18560.83	184.45	0	1536
M3	12.40	1	85	1689.75	18438.00	13.33	0	159
M4	83.1	4	TL	256.00	413.00	3933.60	0	43423
M4*	97.53	5	576	256.00	413.00	17.70	0	111

	cuts		
	av	min	max
M4	901.18	102	3050
M4*	738.05	113	

As observed earlier, Methods 1, 2, and 3 are faster than Method 4, despite their larger number of variables and constraints. This statement remains valid if we account for the number of cuts generated in Method 4 or Method 4*. It can be noticed that the use of USERCUT callbacks additionally to LAZYCUT callbacks in Method 4 leads to the exploration of fewer nodes and the addition of fewer cuts compared to the use of callbacks solely for integer solutions. However, even so, Method 4* is not faster than Method 4,

as evidenced by Figure 3.11. The difference in running times is actually not very significant, as we noticed that different runs may lead to slightly different results.

As observed in the first part of this chapter, when the task is to solve variants of **MWCS**, the PI formulation is not efficient compared to the ARC or the SEA formulations in terms of running time, and PI3 is less efficient than the ARC3 and CY3 formulations. It was pretty surprising, therefore, that Method 2 is more efficient than Methods 3 and 4 for the RR-2-KEP problem.

To confirm our observations, we also compared the four different methods on the same instances but with the maximum cycle length set to $K = 4$. As shown in Figure 3.13, here again, Methods 1, 2, and 3 have similar performance profiles, and Method 4 is the least efficient one.

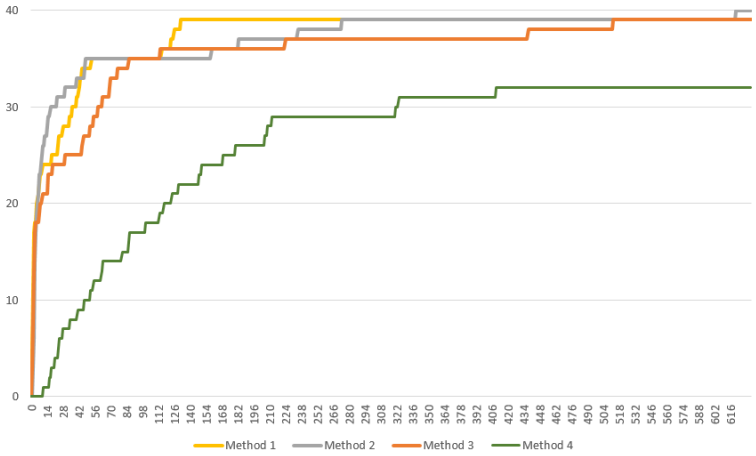


Figure 3.13: Comparison of running time when $n = 25$, $|S| = 100$, $b = 10$, $K = 4$. The horizontal axis displays running times (in seconds), and the vertical axis displays the number of instances solved within a given time.

3.4 Conclusion

In this chapter, we have examined the computational aspects of various formulations that describe cycle selections in directed graphs, with a focus on the maximum weighted cycle selection problem. In Chapter 2, six IP formulations are presented, but the majority of the theoretical study is centered on the ARC formulation. Our experiments have demonstrated that the ARC formulation and the SEA formulation are most efficient among the six formulations. While **MWCS** is computationally easy to solve on random graphs, we have also investigated variants that arise by adding a budget constraint and/or a maximum cycle length constraint to the model. These variants proved more challenging, especially when there is a budget constraint.

Our study was strongly motivated by the work of Smeulders et al. (2022) and their stochastic two-stage KEP problem. We have examined two of their formulations and their respective optimization methods, and we have proposed two new formulations. Although we were unable to improve the running time of the original methods, one of our new methods based on the SEA formulation has a similar running time to that of Smeulders et al. (2022). The performance of the Benders decomposition procedure on the ARC formulation, however, turned out to be quite disappointing in spite of several enhancements which are known to improve the efficiency of the optimization process in other settings.

Chapter 4

Local stability in kidney exchange programs

The content of this chapter is based on the working paper Baratto et al. (2023) with Yves Crama, João Pedro Pedroso and Ana Viana submitted for publication in the European Journal of Operational Research.

Contents

4.1 Introduction

Nowadays, the preferred treatment option offered to patients with an end-stage renal disease is to receive a kidney transplant from a living donor. This option is primarily used when the patient has a relative who is willing to donate a healthy kidney. However, in many situations, the transplantation cannot take place due to immunological incompatibility (based, say, on blood and tissue type) between the patient and the healthy donor.

Kidney exchange programs (KEPs) try to alleviate this limitation by enlisting a (hopefully) large number of incompatible patient-donor pairs, say, pairs (P_i, D_i) made up of patient P_i and donor D_i, for $i = 1, \ldots, n$. Considering such a pool makes it potentially feasible to identify, for example, three patients P_1, P_2, P_3 and three donors D_1, D_2, D_3 such that D_1 is compatible with P_2, D_2 is compatible with P_3, and D_3 is compatible with P_1. Then, three kidneys can be transplanted in cyclic fashion among these six individuals. Kidney exchange programs typically try to maximize the number of transplants at a given time by matching as many compatible individuals as possible. But other objectives may be (sometimes, simultaneously) pursued as well; see, e.g., Biró et al. (2021).

When identifying a potential exchange, it is important to realize that from the point of view of the patients, not all donors' kidneys are equal: indeed, some kidneys may be preferred to others because they are more likely to allow successful transplants or longer survival expectancy. Hence, if an exchange $\mathcal{M}$ is proposed by the program, but another cycle c exists such that all patients of c are better off in c than in $\mathcal{M}$, then the exchange $\mathcal{M}$ may be considered as *unstable*, in the sense that it may be difficult to convince the patients involved in c, and more specifically their medical teams, that $\mathcal{M}$ should be implemented. (A more rigorous definition will be given in Section 4.3.)

The concept of stability has been widely studied in the literature on matching under preferences (Gale and Shapley, 1962) and, to a lesser extent, on kidney exchanges (Roth et al., 2004; Klimentova et al., 2023). This literature will be briefly reviewed in Section 4.2. In Section 4.3, we reexamine the concept and we introduce a weaker definition of *local stability* which appears to be more relevant in the context of kidney exchanges. Section 4.4 introduces the *blocking digraph* G^* associated with a KEP compatibility graph. We observe that stable exchanges correspond to *kernels* of G^*, while locally stable exchanges correspond to *local kernels* of G^*. The section also mentions some basic properties of kernels and local kernels. We prove that it is NP-complete to determine whether a graph has a nonempty local kernel, and hence, to find a local kernel of maximum size. In Section 4.5, we propose integer programming formulations for local stable exchanges. Section

4.6 reports on various numerical tests, including an assessment of the quality of IP formulations for the maximum local stable exchange problem, and a comparison with the results obtained by Klimentova et al. (2023) for the (more restrictive) maximum stable exchange problem. Finally, in Section 4.7, the concept of local stable exchange is extended to the concept of *local strongly stable exchange*. An IP formulation is proposed and numerical tests are conducted for the computation of maximum local strongly stable exchanges.

4.2 Basic concepts and literature review

4.2.1 Stable matching under preferences

The first matching problem involving preferences on possible outcomes has been studied by Gale and Shapley (1962) under the name of *stable marriage problem*. The stable marriage problem involves two disjoint sets of identical size n consisting respectively, say, of men and women, such that each individual has a strict preference order over all the individuals of the opposite sex. The aim is to identify a matching $\mathcal{M}$ of n pairwise disjoint couples (m, w) (where m is a man and w is a woman) which is *stable* in the sense that there is no *blocking* pair $(m_0, w_0) \notin \mathcal{M}$ i.e. a pair (m_0, w_0) such that m_0 prefers w_0 to his partner in $\mathcal{M}$, and w_0 prefers m_0 to her partner in $\mathcal{M}$. Gale and Shapley showed, in particular, that a stable matching always exists and can be found by a polynomial algorithm.

Various extensions of this classical problem have been investigated in the operations research and economic literature, such as bipartite matching problems with two-sided preferences (e.g., the hospitals-residents assignment problem), bipartite matching problems with one-sided preferences (e.g., the house allocation problem), or non-bipartite matching problems with preferences (e.g., the stable roommates problem), as well as many variants that consider complete or incomplete preference lists, with or without ties. Such extensions have been extensively studied from an algorithmic perspective, and polynomial algorithms or hardness results are available for many of them; see, e.g., Gale and Shapley (1962), Irving (1985), Ng and Hirschberg (1991), Manlove et al. (2002), Biró and McDermid (2010), Huang (2010), Manlove (2013).

4.2.2 Optimal kidney exchanges

The optimization of kidney exchanges is a more recent topic but has generated an abundant literature over the past 20 years, in the footprints of a seminal paper by Roth et al. (2004).

A classical model is described as follows. A *compatibility digraph* $G = (V, A)$

is associated with the pool of patient-donor pairs (P_i, D_i), $i = 1, \ldots, n$: the vertex set of G is the set $V = \{1, \ldots, n\}$, and the arc set A contains the arc (i, j) if and only if donor D_i is compatible with patient P_j. A (feasible) *exchange* is a collection of *vertex-disjoint* directed cycles of G. Maximizing the number of feasible transplants amounts therefore to finding in G an exchange which contains as many vertices as possible.

In practice, kidney transplants associated with a cycle are usually carried out simultaneously in order to prevent situations where a donor would drop out once its intended recipient has received a transplant, without the donor itself donating a kidney. In view of the medical and logistical complexity of the resulting procedure, the cycles included in an exchange are usually restricted in size, say, cycles of size at most two, three, or four. We accordingly speak of K-way exchanges, with $K \in \mathbb{N}$.

There is a large amount of literature documenting formulations and algorithms for kidney exchange optimization problems; see, e.g., Abraham et al. (2007), Roth et al. (2007), Constantino et al. (2013), Dickerson et al. (2016), Biró et al. (2021), Delorme et al. (2023a), Delorme et al. (2023b). When $K = 2$ or when $K = n$, maximizing the number of transplants reduces to a weighted matching problem and hence, can be done in polynomial-time. But the problem is NP-hard for any fixed $K \geq 3$.

Besides cycles, some programs also involve non-directed donors (NDD), i.e., donors with no associated patient. When this is the case, directed paths (called *chains*) starting with an NDD are also allowed to be part of an exchange: the NDD can initiate a sequence of transplants by donating a kidney to a patient in a (patient, donor) pair, the donor of that pair donates a kidney to another patient, and so forth until the last donor of the chain donates a kidney to the deceased donor waiting list or becomes available to initiate another chain on the next run of the program. Here again, a limit on the maximum chain length is usually imposed. Chains can be taken into account in the compatibility digraph model by adding dummy arcs between each pair and each non-directed donor: in this way, chains are transformed into cycles in the augmented digraph. For the remainder of this chapter, chains will not be explicitly mentioned and will be handled in the same way as cycles.

4.2.3 Stable kidney exchanges

The concept of *stable kidney exchange* extends the concept of stable matching. It will be defined more precisely in subsequent sections. For now, we can already mention that it was introduced as a natural solution concept in the early work by Roth et al. (2004). When the cycle length is not bounded ($K = n$), these authors observed that stable exchanges are equivalent to

core solutions of a model of the housing market previously studied by Shapley and Scarf (1974). It follows that a stable exchange always exists and can be efficiently computed. At the other end of the spectrum, when $K = 2$, stable exchanges correspond to stable solutions of the roommates problem with incomplete preference lists. Manlove (2013) surveys some of the main results on the stable roommate problem. Let us just mention here that when no ties are allowed in the preference lists, then the existence of complete stable solutions can be checked in polynomial time (Irving, 1985) and a maximum stable solution can also be found in polynomial time (Tan, 1990). On the other hand, the existence question becomes NP-complete when ties are allowed (Ronn (1990), Manlove et al. (2002)).

When the maximum cycle length is greater than or equal to 3 ($K \geq 3$), it is NP-complete to decide if a stable kidney exchange exists. This follows from a result of Biró and McDermid (2010) and Huang (2010) for three-sided stable matchings with cyclic preferences, which is a special case of the stable exchange problem; see also Mészáros-Karkus (2017) for extensions.

The papers cited above focus on the theoretical complexity of stable exchange problems. More recently, Klimentova et al. (2023) turned to the challenge of actually *computing* stable kidney exchanges for large size, realistic compatibility digraphs. They defined different optimization variants of the problem, proposed several integer programming formulations, and carried out extensive numerical experiments with these formulations.

As mentioned in the Introduction, the main contribution of the present chapter is to propose an alternative, weaker concept of *local stability* for kidney exchanges, to investigate some of its theoretical properties, and to compare it experimentally with the classical concept handled in Klimentova et al. (2023). The next section introduces this new concept.

4.3 Stability and local stability

4.3.1 Stability: definitions

Let $G = (V, A)$ be an arbitrary digraph. For a vertex $i \in V$, we denote as $N_G^-(i)$ the set of *in-neighbors* of i, that is, $N_G^-(i) = \{j \in V : (j, i) \in A\}$.

When G is a compatibility digraph for kidney exchanges, we assume that each patient has expressed *preferences* over its set of compatible donors. More precisely, for each $i \in V$, the preferences of patient P_i are described by a *rank* function $r_i : N_G^-(i) \to \mathbb{R}$, with the interpretation that, for all $j, k \in N_G^-(i)$, P_i *prefers* donor D_j to donor D_k (or for short, i prefers j to k) if and only if $r_i(j) < r_i(k)$. We say that P_i is *indifferent* between D_j and D_k (or that i is indifferent between j and k) if $r_i(j) = r_i(k)$.

For example in Figure 4.1 hereunder, $r_2(4) = 1 < r_2(1) = 2$, meaning that the patient of pair 2 prefers the donor of pair 4 to the donor of pair 1.

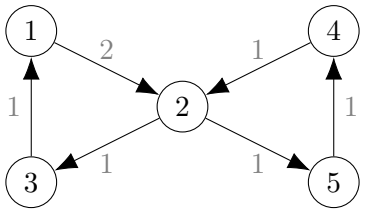


Figure 4.1: A small digraph with arcs labeled by values of the rank functions

Given an integer parameter K, let $\mathcal{C}_K(G)$ be the set of *K-cycles* of G, that is, the set of directed cycles of G with length at most K. In the sequel, when we speak of a cycle, we always mean a directed cycle in $\mathcal{C}_K(G)$, where K is assumed to be fixed. We use letters like $u, v, w, \ldots$ to denote cycles of G (this is admittedly unusual, but will become natural in Section 4.4). For any cycle u, we let $V(u)$ be the set of vertices of u, and we let $A(u)$ be its set of arcs.

Definition 9. An *exchange* of G is a collection $\mathcal{M} \subseteq \mathcal{C}_K(G)$ of pairwise vertex-disjoint K-cycles. A vertex i *is matched* in $\mathcal{M}$ or simply, i *is in* $\mathcal{M}$, if i is contained in one of the cycles of $\mathcal{M}$. We denote by $V(\mathcal{M}) = \bigcup_{u \in \mathcal{M}} V(u)$ the set of vertices matched in $\mathcal{M}$ and by $A(\mathcal{M}) = \bigcup_{u \in \mathcal{M}} A(u)$ the set of arcs included in $\mathcal{M}$.

Definition 10. Let $\mathcal{M}$ be an exchange, let $u \in \mathcal{C}_K(G) \setminus \mathcal{M}$ be a cycle not contained in $\mathcal{M}$, and let $i \in V(u)$. We say that vertex i *prefers* the cycle u to the exchange $\mathcal{M}$ if either

- $i \notin V(\mathcal{M})$, or
- $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i prefers k to k'.

In the context of kidney exchanges, the first condition in this definition expresses the assumption that any vertex i prefers being matched (in cycle u) over being unmatched (in $\mathcal{M}$). The second condition states that i prefers its donor in the cycle u to its donor in the exchange $\mathcal{M}$. When $\mathcal{M}$ consists of a single cycle, say, $\mathcal{M} = \{v\}$, we simply say that i prefers u to v.

Definition 11. A *blocking cycle* for an exchange $\mathcal{M}$ is a cycle $u \in \mathcal{C}_K(G) \setminus \mathcal{M}$ (not contained in $\mathcal{M}$) such that each vertex in $V(u)$ prefers u to $\mathcal{M}$. When $\mathcal{M} = \{v\}$, we say that u is blocking for v.

So, each vertex i in a blocking cycle u would prefer being contained in the transplantation cycle u rather than in the exchange $\mathcal{M}$ (either because i is not matched in $\mathcal{M}$, or because i prefers its donor in u to its donor in $\mathcal{M}$). This naturally leads to the definition of a stable exchange (see Gale

and Shapley (1962), Roth et al. (2004), Biró and McDermid (2010), Huang (2010), Klimentova et al. (2023)).

Definition 12. An exchange $\mathcal{M}$ is *stable* if there is no blocking cycle for $\mathcal{M}$ in $\mathcal{C}_K(G)$.

Example 1. In Figure 4.1, the exchange $\mathcal{M} = \{u\}$, where $u := (1, 2, 3, 1)$, is not stable. Indeed, $v := (2, 5, 4, 2)$ is a blocking cycle for $\mathcal{M}$, since the patient of pair 2 prefers the donor of pair 4 to the donor of pair 1.

Note that when $K = 2$, Definition 12 mimicks the definition of stable matchings given in Section 4.2.

4.3.2 Local stability: definitions

Example 2. Consider the digraph G in Figure 4.2. For $K = 2$, there are four cycles of interest, namely, $u_1 := (1, 2, 1)$, $u_2 := (2, 3, 2)$, $u_3 := (3, 1, 3)$ and $u_4 := (4, 5, 4)$. There is no stable exchange in G. Indeed, at most one of the cycles u_1, u_2, u_3 can be selected in an exchange, but u_1 is blocking for u_2, u_2 is blocking for u_3, and u_3 is blocking for u_1. Moreover, since u_1, u_2, u_3 are disjoint from u_4, they all block the exchange $\mathcal{M} = \{u_4\}$.

Note however that, from the point of view of the patients of a kidney exchange program, it does not make sense to reject $\mathcal{M} = \{u_4\}$ since this cycle could be implemented without affecting any of the options available to the remaining patient-donor pairs and hence, without opposition from any of them.

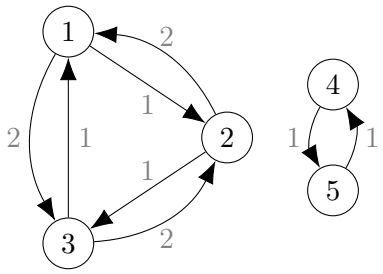


Figure 4.2: A digraph without stable exchange

The anomaly underlined in Example 2 arises because Definition 11 does not impose that a blocking cycle u for $\mathcal{M}$ should intersect $\mathcal{M}$ (in the sense that $V(\mathcal{M}) \cap V(u) \neq \emptyset$). As a result, an exchange (like $\mathcal{M} = \{u_4\}$) can be blocked by a cycle (say, u_1) which is disjoint from it and which, intuitively, is therefore unrelated. These observations motivate the consideration of a weaker and seemingly new notion of stability, that we now proceed to introduce.

Definition 13. A *locally blocking cycle*, or *L-blocking cycle*, for an exchange $\mathcal{M}$ is a blocking cycle for $\mathcal{M}$ that intersects $\mathcal{M}$. In other words, it is a cycle

u that is not contained in $\mathcal{M}$, that intersects $\mathcal{M}$, and such that each vertex in $V(u)$ prefers u to $\mathcal{M}$. When $\mathcal{M} = \{v\}$, we say that u is blocking for v.

Definition 14. An exchange $\mathcal{M}$ is *locally stable*, or *L-stable*, if there is no L-blocking cycle for $\mathcal{M}$ in $\mathcal{C}_K(G)$.

Comparing Definition 11 and Definition 13 makes it clear that every L-blocking cycle is also blocking, but the converse is not necessarily true. As a consequence, every stable exchange is locally stable, but the reverse implication does not hold in general.

Example 3. In Figure 4.3, the cycle $u := (2, 4, 2)$ is L-blocking for the cycle $v := (1, 2, 3, 1)$. Indeed, these cycles have vertex 2 in common, which prefers u to v because it prefers the donor of pair 4 to the donor of pair 1, and because vertex 4 is unmatched in v.

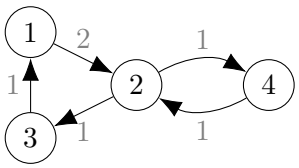


Figure 4.3: $u := (2, 4, 2)$ is L-blocking for $v := (1, 2, 3, 1)$

Example 4. In Example 2 and Figure 4.2, the exchange $\mathcal{M} = \{u_4\}$ is not stable, but it is locally stable. Indeed, the cycles u_1, u_2, u_3 block the cycle u_4, but they do not intersect it. Hence, they are not locally blocking for u_4.

Our proposal in this chapter is that local stability is a pertinent property to be considered by decision-makers in a kidney exchange program. Indeed, just like classical stability, it expresses the property that no subset of patient-donor pairs has a common incentive to block an exchange in order to participate in a more attractive cyclic sequence of transplants. Contrary to stability, however, local stability is not affected by the consideration of irrelevant alternatives, in the sense that a cycle cannot block an exchange on which it has no immediate bearing, as illustrated by Example 4. In subsequent sections, we will demonstrate that this meaningful generalization of the stability requirement sometimes allows the identification of large local stable exchanges in situations where no stable exchanges exist.

Remark 1. A reviewer of the initial submission to EJOR pointed out the concept of *internal stability*, which was introduced by Liu et al. (2014). Let us say that a cycle $u \in \mathcal{C}_K(G)$ is *internally blocking* for an exchange $\mathcal{M}$ if u is blocking for $\mathcal{M}$ and $V(u) \subseteq V(\mathcal{M})$. An exchange $\mathcal{M}$ is internally stable if there is no internally blocking cycle for $\mathcal{M}$. Since an internally blocking cycle is locally blocking, the following implications hold for every exchange $\mathcal{M}$:

$$\mathcal{M} \text{ stable } \Rightarrow \mathcal{M} \text{ locally stable } \Rightarrow \mathcal{M} \text{ internally stable.}$$

To see that the second implication cannot be reversed, it is enough to consider the digraph induced by vertices 1, 2, 3 in Example 2. For $K = 2$, this digraph has an internally stable exchange (e.g., the 2-cycle $u_1 = (1, 2, 1)$), but it has no nonempty L-stable exchange. The relation between L-stability and internal stability deserves to be further examined in future work.

4.3.3 Stability and local stability: characterizations

In this section, we provide alternative characterizations of stable and L-stable exchanges which will be used in Section 4.5 to derive integer programming formulations.

Definition 15. For a cycle $v \in \mathcal{C}_K(G)$, we denote by $\mathcal{B}(v)$ the set of all L-blocking cycles for v.

Definition 16. Two cycles u, v are *friends* if they have a nonempty intersection, if u does not L-block v, and if v does not L-block u. We denote by $\mathcal{F}(v)$ the set of cycles that are friends with v.

Clearly, $u \in \mathcal{F}(v)$ if and only if $v \in \mathcal{F}(u)$. Cycles u and v are friends when some vertex i in $V(u) \cap V(v)$ has no preference between the two cycles (for example if the cycles share an arc (j, i) as illustrated in Figure 4.4), or if the two cycles share at least two vertices and one prefers u while the other one prefers v.

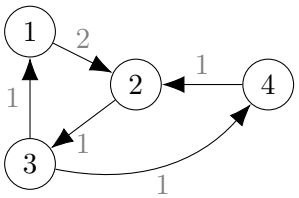


Figure 4.4: $v = (1, 2, 3, 1)$ and $u = (2, 3, 4, 2)$ are friends when $K = 3$

We note the following property for future reference.

Lemma 1. Two cycles u, v intersect each other if and only if $u \in \mathcal{B}(v) \cup \mathcal{F}(v)$ or $v \in \mathcal{B}(u) \cup \mathcal{F}(u)$.

Proof. This trivially follows from the definitions. □

The following result will be crucial for the subsequent developments.

Lemma 2. Let $\mathcal{M}$ be an exchange and let v be a cycle not contained in $\mathcal{M}$. The following statements are equivalent:
(i) there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$;

(ii) v is not blocking for $\mathcal{M}$;
(iii) v intersects $\mathcal{M}$ and v is not L-blocking for $\mathcal{M}$.

Proof. (i) $\Rightarrow$ (ii). Let $w \in \mathcal{M}$. If $w \in \mathcal{B}(v)$, then by definition $V(v) \cap V(w)$ is not empty, and any vertex in the intersection prefers w to v. It follows that v is not blocking for $\mathcal{M}$.

If $w \in \mathcal{F}(v)$, then again $V(v) \cap V(w)$ is not empty and $v \notin \mathcal{B}(w)$. Hence, there must be a vertex $i \in V(v) \cap V(w)$ such that i does not prefer v to w. So, once again, v is not blocking for $\mathcal{M}$.

(ii) $\Leftrightarrow$ (iii). This equivalence is just a restatement of Definition 13.

(iii) $\Rightarrow$ (i). If v intersects $\mathcal{M}$, but v is not L-blocking for $\mathcal{M}$, it means that there exists a cycle $w \in \mathcal{M}$ and a vertex $i \in V(v) \cap V(w)$ such that either i prefers w to v, or i is indifferent between v and w. In particular, $v \notin \mathcal{B}(w)$. But then, Lemma 1 implies that $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$. □

We are now ready for the characterization theorems.

Theorem 28. *For an exchange $\mathcal{M}$, the following conditions are equivalent:*
(a) *$\mathcal{M}$ is stable;*
(b) *for each cycle $v \notin \mathcal{M}$, there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$.*

Proof. This immediately follows from Definition 12 and from the equivalence of (i)-(ii) in Lemma 2. □

Theorem 29. *For an exchange $\mathcal{M}$, the following conditions are equivalent:*
(a) *$\mathcal{M}$ is L-stable;*
(b) *for each cycle $v \notin \mathcal{M}$, if v intersects $\mathcal{M}$, then there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$.*

Proof. (a) $\Rightarrow$ (b). If $\mathcal{M}$ is L-stable and $v \notin \mathcal{M}$, then v cannot be L-blocking for $\mathcal{M}$. So, if v intersects $\mathcal{M}$, then condition (iii) of Lemma 2 holds. This implies that condition (i), and hence (b), also hold.

(b) $\Rightarrow$ (a). Conversely, if (b) holds, then every cycle $v \notin \mathcal{M}$ is either disjoint from $\mathcal{M}$ (in which case it is not L-blocking) or satisfies condition (i) of Lemma 2 (in which case it is also not L-blocking, in view of condition (iii)). Hence, $\mathcal{M}$ is L-stable. □

4.4 Blocking digraph, kernels and local kernels

The aim of this section is to provide alternative interpretations of stable and L-stable exchanges in terms of a digraph $G^* = (V^*, A^*)$, to be called the

blocking digraph of G, that we define as follows:

- $V^* = \mathcal{C}_K(G)$: there is a vertex v in V^* for each cycle v in $\mathcal{C}_K(G)$;
- $A^* = \{(v, w) : w \in \mathcal{B}(v) \cup \mathcal{F}(v)\}$.

Remark 2. In view of Lemma 1, when two cycles u, v intersect, then at least one of the arcs (u, v) or (v, u) is in A^* (both arcs are in A^* exactly when u and v are friends). And conversely, if (u, v) is an arc in A^*, then u and v intersect. So, G^* can be viewed as an orientation of the intersection graph of K-cycles of G. When $K = 2$, G^* is an orientation of a line graph (see Boros and Gurvich (2006), Maffray (1992), Ratier (1996) for related constructions when G is bipartite).

The following concept is classical in game theory and graph theory; see, e.g., von Neumann and Morgenstern (1953), and Boros and Gurvich (2006) for related literature.

Definition 17. A *kernel* in a digraph $D = (W, E)$ is a subset $S \subseteq W$ which is both *independent* and *absorbing*:

- *independent*: for all $(u, v) \in E$ at most one of u, v is in S;
- *absorbing*: for every vertex $v \notin S$, there exists a vertex $w \in S$ such that $(v, w) \in E$ (see Figure 4.5).

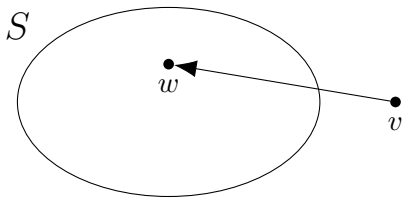


Figure 4.5: An absorbing set S

From Theorem 28 and the definition of kernels, we immediately obtain:

Theorem 30. *For a digraph $G = (V, A)$ and its blocking digraph $G^* = (V^*, A^*)$, the stable exchanges of G are exactly the kernels of G^*.*

The relation expressed in Theorem 30 does not come as a complete surprise, as similar observations have been formulated in the literature, for example, for the stable marriage problem (see, e.g., Manlove (2013), Ratier (1996)), or in a different context, for certain types of hedonic games (see, e.g., Deineko and Woeginger (2013), Igarashi (2017)). We are not aware of any reference where the relation is explicitly stated for stable kidney exchanges, therefore we state Theorem 30 for the record and to prepare the statement of Theorem 31 hereunder.

Let us now turn to our new notion of local stability. Galeana-Sánchez and

Neumann-Lara (1984) define local kernels as follows (the terminology is due to Duchet and Meyniel (1993)).

Definition 18. A *local kernel*, or *L-kernel*, of a digraph $D = (W, E)$ is a subset S of vertices which is both independent and *locally absorbing*:

- *locally absorbing*: for all $u \in S$ and $v \notin S$ such that $(u, v) \in E$, there exists $w \in S$ such that $(v, w) \in E$.

The second condition in this definition means that every out-neighbor of S is "absorbed" by S. Figure 4.6 provides an illustration. Clearly, every kernel is a local kernel.

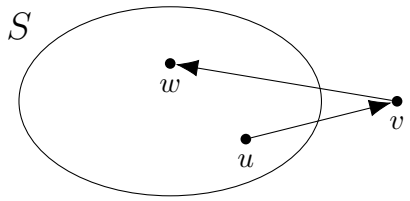


Figure 4.6: A locally absorbing set S

The relation between L-stable exchanges and L-kernels is akin to the relation between stable exchanges and kernels (Theorem 30), namely:

Theorem 31. *For a digraph $G = (V, A)$ and its blocking digraph $G^* = (V^*, A^*)$, the L-stable exchanges of G are exactly the L-kernels of G^*.*

Proof. Assume that $\mathcal{M}$ is an L-stable exchange in G. Then, $\mathcal{M}$ is independent in G^*. If $u \in \mathcal{M}$, $v \notin \mathcal{M}$ and $(u, v) \in A^*$, then u intersects v (by Lemma 1). By statement (b) in Theorem 29 and by definition of the blocking digraph, there exists $w \in \mathcal{M}$ such that $(v, w) \in A^*$, and hence $\mathcal{M}$ is a local kernel in G^*.

Conversely, if $\mathcal{M}$ is an L-kernel in G^*, then $\mathcal{M}$ is an exchange in G. To verify statement (b) in Theorem 29, suppose that $v \notin \mathcal{M}$ and that v intersects $\mathcal{M}$, i.e., there is a cycle $u \in \mathcal{M}$ such that v intersects u. In view of Remark 2, then, $(u, v) \in A^*$ or $(v, u) \in A^*$ (or both). If $(v, u) \in A^*$, then $u \in \mathcal{B}(v) \cup \mathcal{F}(v)$ by definition of A^*, and hence condition (b) of Theorem 29 holds. If $(u, v) \in A^*$, then by definition of L-kernels there exists $w \in \mathcal{M}$ such that $(v, w) \in A^*$, and condition (b) is satisfied again. □

There only seems to be a handful of publications about local kernels. We collect here some simple observations of interest, in particular about the relation between kernels and local kernels.

Fact 1. Not every digraph has a kernel, but every digraph has an L-kernel, since the empty set always is an L-kernel.

Fact 2. A directed cycle of odd length, say $(u_1, u_2, \ldots, u_{2\ell+1}, u_1)$, has no L-kernel other than the empty set. Indeed, in any nonempty independent set S of this odd cycle, there is a vertex $u_k \in S$ such that u_{k+1}, u_{k+2} are not in S. Then, u_k and u_{k+1} violate the definition of local absorption.

For a digraph G, let us say that a kernel (respectively, an L-kernel) of G is *maximal* if it is not properly included in another kernel (respectively, L-kernel) of G.

Fact 3. It is easy to see that a kernel can never be included in another one. In other words, every kernel is maximal. Every kernel also is a maximal L-kernel. On the other hand, in view of Fact 1, an L-kernel is not necessarily maximal, and a maximal L-kernel is not necessarily a kernel.

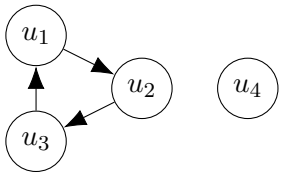


Figure 4.7: A blocking digraph without kernel but with a nonempty L-kernel

Example 5. Figure 4.7 illustrates the previous facts. It displays the blocking digraph G^* of the digraph G in Figure 4.2. In line with the discussion in Section 4.3.2, G^* has no kernel (essentially, because of the 3-cycle (u_1, u_2, u_3, u_1)), but $S = \{u_4\}$ is a maximal L-kernel of G^*.

Later in the chapter, we will be interested in computing *maximum* kernels and L-kernels, that is, kernels and L-kernels of maximum size.

Fact 4. Even when a kernel exists, the maximum size of an L-kernel can be strictly larger than the maximum size of a kernel.

Example 6. The digraph G^* in Figure 4.8 illustrates this fact. Indeed, $\{u_3\}$ is the unique kernel of G^*, while $\{u_1, u_2\}$ is its largest L-kernel. The size of the maximum L-kernel could actually be made arbitrarily large by creating multiple copies of vertices u_1 and u_2.

It is interesting to observe that G^* in Figure 4.8 is the blocking digraph of the KEP compatibility graph in Figure 4.9 for $K = 4$, where $u_1 = (1, 2, 3, 4, 1)$, $u_2 = (5, 6, 7, 8, 5)$, $u_3 = (3, 5, 14, 9, 3)$, $u_4 = (12, 13, 14, 15, 12)$, $u_5 = (9, 10, 11, 12, 9)$, and $u_6 = (9, 16, 17, 13, 9)$. As a consequence, the maximum stable exchange in G is $\mathcal{M}^s = \{u_3\}$ and the maximum L-stable exchange is $\mathcal{M}^{ls} = \{u_1, u_2\}$.

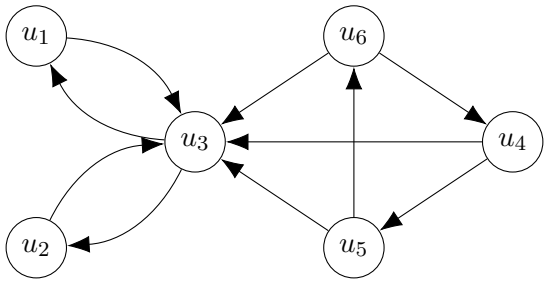


Figure 4.8: A digraph G^* with an L-kernel larger than the unique kernel

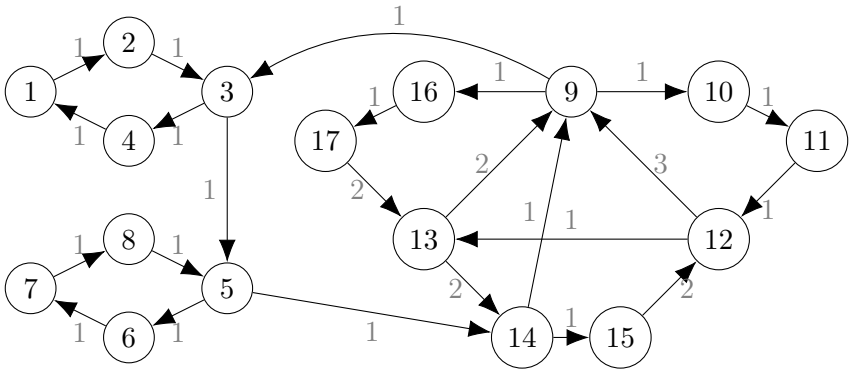


Figure 4.9: A compatibility digraph G

Fact 4 and Example 6 confirm that a maximum locally stable exchange might be larger than a maximum stable exchange. In the context of kidney exchanges, it means that an L-stable exchange may increase the number of transplants. This observation underlines the potential relevance of locally stable exchange.

In Section 4.6, we will turn to the computation of maximum L-kernels and L-stable exchanges. Chvátal (1973) proved that deciding whether a digraph has a kernel is an NP-complete problem. The NP-hardness results cited in Section 4.2 for stable exchanges strengthen this statement, and polynomial algorithms exist for some classes of digraphs; see, e.g., Pass-Lanneau et al. (2020). On the other hand, the complexity of computing L-kernels has apparently not been investigated in the literature, but we can establish:

Theorem 32. *Given a digraph $G = (V, A)$, deciding whether G has a nonempty local kernel is NP-complete.*

Proof. The problem is clearly in NP. The completeness proof is inspired from Chvátal (1973). We provide a reduction from the satisfiability problem, namely: given a Boolean conjunctive normal form F on n variables $x_1, \ldots, x_n$, say, $F = C_1 \vee \ldots \vee C_m$, we construct a digraph $D = (W, E)$ such

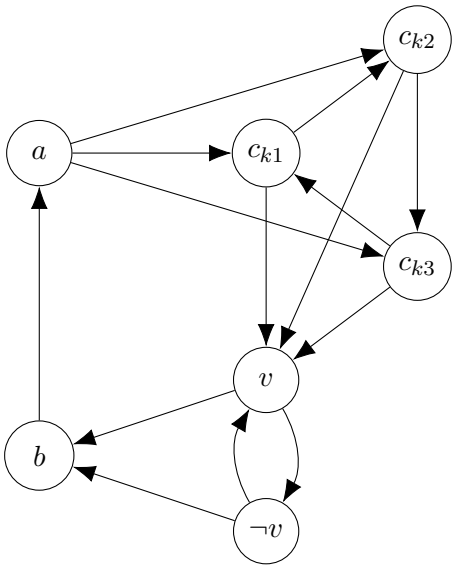


Figure 4.10: Construction for a clause C_k containing a literal v

that D has a nonempty local kernel if and only if F is satisfiable.

- For each variable x_i of F, we create two vertices x_i and $\neg x_i$ in W and join them by the arcs $(x_i, \neg x_i)$, $(\neg x_i, x_i)$ in E.
- For each clause C_k of F, we introduce three vertices c_{k1}, c_{k2}, c_{k3} in W, and join them in the cyclic triangle $(c_{k1}, c_{k2}), (c_{k2}, c_{k3}), (c_{k3}, c_{k1})$.
- For each pair (c_{kj}, v) such that literal v appears in clause C_k, we add the three arcs $(c_{k1}, v), (c_{k2}, v), (c_{k3}, v)$ in E.
- We add a new vertex a and all the arcs of the form (a, c_{kj}), for all clauses C_k and for all $j \in \{1, 2, 3\}$.
- We add a new vertex b, the arc (b, a), and all the arcs of the form $(x_i, b), (\neg x_i, b)$ in E.

The construction is illustrated in Figure 4.10 for a clause C_k containing a literal v (which may be a variable x_i or its negation).

Suppose now that $x^* = (x_1^*, \ldots, x_n^*) \in \{0,1\}^n$ is a satisfying assignment for F. Then $S^* = \{a\} \cup S$, with $S = \{x_i | x_i^* = 1\} \cup \{\neg x_i | x_i^* = 0\}$, is a nonempty local kernel (and even, a kernel) of D. Indeed, S^* is an independent set, and all vertices $x_i \notin S$, $\neg x_i \notin S$, b, c_{kj} are absorbed by some vertex of S^* as follows from the following observations:

- if $x_i \notin S$, then $\neg x_i \in S$ and $\neg x_i$ absorbs x_i;
- if $\neg x_i \notin S$, then $x_i \in S$ and x_i absorbs $\neg x_i$;

- b is absorbed by a;
- since x^* satisfies F, at least one literal v in C_k is true, for each k; hence, c_{k1}, c_{k2} and c_{k3} are absorbed by v.

Conversely, assume that S^* is a nonempty local kernel of D. Let us see what vertices can be in S^*.

- Assume that $c_{k1} \in S^*$. Then, due to independence, $c_{k2} \notin S^*$, $c_{k3} \notin S^*$, and $v \notin S^*$ for any literal v that appears in C_k. Hence, the local absorption condition of Definition 4.9 is violated, since $c_{k1} \in S^*$, (c_{k1}, c_{k2}) is an arc, but there is no vertex $w \in S^*$ such that $(c_{k2}, w) \in E$. This contradiction shows that $c_{k1} \notin S^*$. The same conclusion holds by symmetry for c_{k2} and c_{k3}.

- Assume that $b \in S^*$. Since (b, a) is an arc, local absorption requires that (a, w) must be an arc for some vertex $w \in S^*$. But this is not possible, since the only arcs leaving a are of the form (a, c_{kj}), and c_{kj} is not in S^* by the previous bullet point: contradiction.

So, at this point, we know that the only vertices that can potentially be in S^* are a, x_i, $\neg x_i$ for $i \in \{1, \cdots, n\}$.

- If $a \notin S^*$, then at least one vertex of the form x_i, $\neg x_i$ must be in S^*, for $i \in \{1, \cdots, n\}$ (since S^* is not empty), say $x_1 \in S^*$. Then, (x_1, b) being an arc, local absorption implies that there is an arc of the form (b, w) with $w \in S^*$. But the only arc leaving b is (b, a), which contradicts the assumption that $a \notin S^*$.

- So, $a \in S^*$. Now, for all k, (a, c_{k1}) is an arc of D. Hence, there must be an arc $(c_{k1}, w) \in E$ for some vertex $w \in S^*$. This vertex w can only be a literal x_i or $\neg x_i$ that appears in clause C_k, for some $i \in \{1, \ldots, n\}$. This shows that, for each clause C_k, at least one literal of C_k must be in S^*.

We conclude that S^* is of the form $\{a\} \cup S$ where S contains at most one of x_i, $\neg x_i$ for each variable x_i, and S contains at least one literal of C_k for each k. Hence, the literals in S define a satisfying assignment for F. (If for some i neither x_i nor $\neg x_i$ appears in S, then the corresponding variable can be assigned an arbitrary value; alternatively, either x_i or $\neg x_i$ can be added to S^*, which remains a local kernel.) □

Note that, as a corollary of Theorem 32, computing a local kernel of maximum size is also NP-hard.

4.5 Integer programming formulations

We are now ready to provide integer programming formulations of stable and L-stable exchanges. For this purpose, we introduce the natural binary variables y_v, for all $v \in \mathcal{C}_K(G)$, with the interpretation that $y_v = 1$ if cycle v is in the exchange.

Consider the following constraints:

$$y_u + y_v \leq 1 \qquad \forall u, v \in \mathcal{C}_K(G) : V(u) \cap V(v) \neq \emptyset \qquad (4.1)$$

$$1 \leq y_v + \sum_{w \in \mathcal{B}(v) \cup \mathcal{F}(v)} y_w \qquad \forall v \in \mathcal{C}_K(G) \qquad (4.2)$$

$$y_v \in \{0, 1\} \qquad \forall v \in \mathcal{C}_K(G). \qquad (4.3)$$

Theorem 33. *The solutions of* (4.1), (4.2), (4.3) *describe all stable exchanges of* G.

Proof. Suppose that y satisfies (4.1), (4.2), (4.3), and let $\mathcal{M}$ be the associated set of cycles. Constraints (4.1) express that $\mathcal{M}$ is an exchange, and constraints (4.2) express condition (b) in Theorem 28. □

Formulation (4.1)-(4.3) can also be viewed as the natural formulation for the kernels of G^*, as found for example in Aharoni and Holzman (1998), Chen et al. (2016). The packing constraints (4.1) can be replaced by the stronger constraints

$$\sum_{v \in \mathcal{C}_K(G) : i \in V(v)} y_v \leq 1 \qquad \forall i \in V \qquad (4.4)$$

since (4.4) expresses that at most one cycle containing a given vertex i can be included in an exchange. Note that the linear relaxation of (4.4) is stronger than the relaxation of (4.1). The resulting strengthened formulation (4.2)-(4.4) is exactly the so-called "cycle formulation" of stable exchanges in Klimentova et al. (2023). (The notations in Klimentova et al. (2023) are slightly different, as these authors define additional subsets $B(i, v)$, for $v \in \mathcal{C}_K(G)$ and $i \in V(v)$, such that $\mathcal{B}(v) \cup \mathcal{F}(v) = \cup_{i \in V(v)} B(i, v)$ for all $v \in \mathcal{C}_K(G)$. Except for this notational difference, the formulations are identical.)

The collection of cycles $\{v \in \mathcal{C}_K(G) : i \in V(v)\}$ is a clique in G^* and hence, (4.4) is one of the well-known clique inequalities

$$\sum_{v \in C} y_v \leq 1 \qquad \text{if } C \text{ is a clique in } G^*.$$

However, (4.4) does not necessarily include all clique inequalities and not even all maximal clique inequalities for G^*. As an example, in Figure 4.8,

$C = \{u_3, u_4, u_5, u_6\}$ is a maximal clique of G^*, but none of the constraints (4.4) is associated with C since no vertex in Figure 4.9 is included in all four cycles u_3, u_4, u_5, u_6.

Let us turn next to a formulation of L-stability. Define the constraints:

$$y_u + y_v \leq 1 \qquad \forall u, v \in \mathcal{C}_K(G) : V(u) \cap V(v) \neq \emptyset \tag{4.5}$$

$$y_u \leq \sum_{w \in \mathcal{B}(v) \cup \mathcal{F}(v)} y_w \qquad \forall u \in \mathcal{C}_K(G), \forall v \in \mathcal{B}(u) \cup \mathcal{F}(u) \tag{4.6}$$

$$y_v \in \{0, 1\} \qquad \forall v \in \mathcal{C}_K(G). \tag{4.7}$$

Theorem 34. *The solutions of* (4.5), (4.6), (4.7) *describe all L-stable exchanges of* G.

Proof. When y satisfies (4.5), (4.6), (4.7), let $\mathcal{M}$ be the associated set of vertices in G^*. Then, $\mathcal{M}$ is independent in G^*. To verify that $\mathcal{M}$ is locally absorbing in G^*, assume that $u \in \mathcal{M}$, $v \notin \mathcal{M}$ and $(u, v) \in A^*$ (that is, $v \in \mathcal{B}(u) \cup \mathcal{F}(u)$). Then, the inequalities (4.6) imply the existence of $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$ such that $y_w = 1$, i.e., $(v, w) \in A^*$ and $w \in \mathcal{M}$ as required for local absorption. Hence, (4.5)-(4.7) exactly describes the local kernels of G^*. □

The formulation (4.5)-(4.7) can be rewritten as the following natural formulation of local kernels in the digraph $G^* = (V^*, A^*)$:

$$y_u + y_v \leq 1 \qquad \forall (u, v) \in A^* \tag{4.8}$$

$$y_u \leq \sum_{w:(v,w) \in A^*} y_w \qquad \forall (u, v) \in A^* \tag{4.9}$$

$$y_v \in \{0, 1\} \qquad \forall v \in V^*. \tag{4.10}$$

Different formulations can be derived as follows. First, here again, the constraints (4.5) can be replaced by the tighter clique inequalities (4.4). Next, note that when $v \in \mathcal{F}(u)$, the local absorption inequalities (4.6) are redundant and can be removed: indeed, if $v \in \mathcal{F}(u)$ then $u \in \mathcal{F}(v)$, hence the right-hand side of (4.6) contains y_u and the inequality is satisfied since the variables y_w are nonnegative. Equivalently, constraints (4.9) are redundant when $(v, u) \in A^*$. The remaining non-redundant constraints (4.9) can be aggregated by fixing v and by summing for all u such that $(u, v) \in A^*$ and $(v, u) \notin A^*$ (that is, for all u such that $v \in \mathcal{B}(u)$). This yields the inequalities

$$\sum_{u:(u,v) \in A^*, (v,u) \notin A^*} y_u \leq \delta^-(v) \sum_{w:(v,w) \in A^*} y_w \qquad \forall v \in V^* \tag{4.11}$$

where $\delta^-(v) = |\{u : (u, v) \in A^*, (v, u) \notin A^*\}|$.

Interestingly, one easily verifies that the inequalities (4.11) correctly express the local absorption condition in any digraph $G^* = (V^*, A^*)$ (be it a blocking digraph or not): in other words, inequalities (4.9) and (4.11) have the same 0–1 solutions. There are only $|V^*|$ aggregated constraints of type (4.11), while there are $|A^*|$ constraints of type (4.9). However, the linear relaxation of (4.11) is weaker than that of (4.9). In Section 4.6, we will experimentally compare different formulations based on the above observations.

4.6 Numerical tests for L-stable exchanges

The aim of this section is, first, to assess the practical difficulty of computing maximum L-stable exchanges by solving the IP formulations presented in Section 4.5 and second, to compare optimal stable exchanges against optimal L-stable ones. All formulations were implemented using Python 3.10 as programming language and were tested using Gurobi 9.5.0. The tests were executed on a Dell Latitude 7490 running Windows 10 64Bit in an Intel Core i5-7300U CPU with 2 Cores at 2.60GHz and 16 GB of RAM.

4.6.1 Instances

We have performed numerical tests on a set of instances created by Klimentova et al. (2023) using the random generator developed by Santos et al. (2017). The generator simulates the features of compatibility digraphs arising in real-world KEPs. The instances are described in detail in Klimentova et al. (2023). Each instance is defined by a compatibility KEP digraph $G = (V, A)$, by preferences on the potential donors of each patient, and by a value of K. The number n of incompatible pairs can take 22 distinct values, namely,

$$n \in \{20, 30, \cdots, 170, 180, 200, 250, 300, 350, 400\},$$

and $K \in \{2, 3, 4\}$. Each instance also contains $\lceil 0.05\, n \rceil$ non-directed donors. The chains originating from an NDD are viewed as cycles in an augmented digraph, as explained in Section 4.2.2. Fifty different digraphs with $|V| = n + \lceil 0.05\, n \rceil$ vertices are available for each value of n. So, in total, we have 3300 instances ($22 \times 3 \times 50$) in this dataset.

For each digraph, the preferences on the arcs are strict, that is, a patient is never indifferent between two distinct donors. We have also experimented with instances featuring weak preferences (as in Klimentova et al. (2023)), and with a third data set from Smeulders et al. (2022). Since the results were similar in all cases, we only report here on the first type of instances.

Remark 3. When chains originating from non-directed donors are transformed into cycles by adding "dummy" arcs, the question arises of defining

the preferences of each NDD over its in-neighbors. These preferences could all be set to a same value (say, "most preferred") to translate the fact that an NDD does not actually receive any transplant and hence, is indifferent between its in-neighbors. Alternatively, for each dummy arc (i, k), where k is an NDD, we can consider that the donor D_i donates a kidney to the deceased donor waiting list or acts as an NDD in a future run of the KEP, as explained in Section 4.2.2. When this is the case, it makes sense to introduce preferences on the dummy arcs so as to express the features of the kidneys donated by the in-neighbors of the NDD k. The instances generated by Klimentova et al. (2023) accordingly include preferences among the in-neighbors of each NDD.

By way of illustration, Table 4.1 displays some of the size parameters of the graphs G and G^* for the 50 instances with $n = 40$ and $K = 3$ (see also Table 4.5 further down for instances of different size, with $K = 2$). In this and subsequent tables, with a slight abuse of notations for $|A|$, $|V^*|$ and $|A^*|$:

- n is the number of patient/donor pairs in each digraph G;
- $|V| = n + \lceil 0.05n \rceil$ is the number of vertices of G;
- $|A|$ is the average number of arcs of G in 50 instances with the same value of n;
- $|V^*|$ is the average number of cycles in 50 instances with the same value of n, that is, the average number of vertices in the corresponding blocking digraphs; the next two columns ($\min|V^*|$ and $\max|V^*|$) display the minimum and the maximum number of cycles in the 50 instances;
- $|A^*|$ is the average number of arcs in the corresponding blocking digraphs; the next two columns ($\min|A^*|$ and $\max|A^*|$) show the minimum and the maximum number of arcs in 50 blocking digraphs, for the same value of n.

Table 4.1: Size parameters of instances with $n = 40$, $K = 3$

n	$\|V\|$	$\|A\|$	$\|V^*\|$	$\min\|V^*\|$	$\max\|V^*\|$	$\|A^*\|$	$\min\|A^*\|$	$\max\|A^*\|$
40	42	471	452	27	1052	58533	156	199874

4.6.2 Comparison of formulations for maximum L-stable exchanges

In this section, we first compare the IP formulations proposed to describe locally stable exchanges by solving the *maximum locally stable exchange problem* with the objective function

$$\max \sum_{u \in \mathcal{C}_K(G)} |V(u)|\, y_u \tag{4.12}$$

where $|V(u)|$ is the length of cycle u. Four different IP formulations have been tested: beside the integrality constraints (4.7), they contain the following L-stability constraints.

- Formulation 1: constraints (4.5) and (4.6); total: $2|A^*|$ constraints.
- Formulation 2: constraints (4.5) and (4.11); total: $|A^*| + |V^*|$ constraints.
- Formulation 3: constraints (4.4) and (4.6); total: $|V|+|A^*|$ constraints.
- Formulation 4: constraints (4.4) and (4.11); total: $|V| + |V^*|$ constraints.

Recall from Section 4.5 that constraints (4.4) and (4.5) have the same 0–1 solutions, but the linear relaxation of (4.4) is stronger than the relaxation of (4.5). Similarly, constraints (4.6) and (4.11) have the same 0–1 solutions, but the relaxation of (4.6) is stronger than the relaxation of (4.11). So, Formulation 2 is in principle the weakest and Formulation 3 is the tightest among these four formulations, whereas Formulations 1 and 4 are incomparable with each other, and are intermediate between 2 and 3. However, the number of constraints also differs significantly and as a result, it becomes hard to predict the total running time of different formulations, in particular when A^* grows large (see Table 4.1 and Table 4.5).

In order to assess numerically the quality of the linear relaxations, we computed the integrality gap $Gap^k_{LP} = 100 \times \frac{z^k_{LP} - z^*}{z^*}$, where z^k_{LP} is the optimal value of the linear relaxation of Formulation k and z^* is the optimal value of the problem. For all instances with $n \in \{20, 30, 40\}$ and $K = 2, 3$, Formulations 1 and 2 appear to have the same integrality gap, and this gap is extremely large. For example, for 50 instances with $n = 40$ and $K = 3$, the integrality gap is in $[112; 6094]$ with a mean value of 2802. The gaps for Formulations 3 and 4 are much smaller. This is illustrated in Figure 4.11 which displays the performance profiles of Gap^3_{LP} and Gap^4_{LP} (it shows the number of instances for which the gap is smaller than the abscissa on the horizontal axis; the gap for Formulations 1 and 2 is too large to be meaningfully displayed in this figure). Both gaps are smaller than 35 for all 50 instances. As expected, Formulation 3 is tighter than Formulation 4, but only slightly so. These results confirm that the clique inequalities (4.4) considerably tighten the formulations, whereas the aggregation of constraints (4.6) into (4.11) does not deteriorate very much the upper bounds.

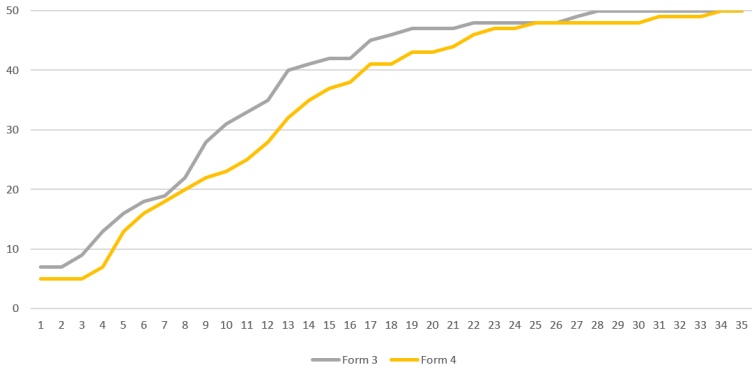


Figure 4.11: Comparison of Gap_{LP} for Formulations 3 and 4 when $n = 40$, $K = 3$. The horizontal axis displays gaps and the vertical axis displays the number of instances with a gap smaller than a given value.

Table 4.2: Mean running time (in seconds) for Formulations 1–4 when $n = 40$, $K = 3$

Formulation 1	36.58
Formulation 2	3.78
Formulation 3	14.67
Formulation 4	0.56

Let us now consider the total running time of the IP solver on the different formulations. Table 4.2 displays the mean running time (in seconds) for each formulation, computed over 50 instances of size $n = 40$ and $K = 3$. Figure 4.12 displays the performance profiles of the running time for the four formulations on the same instances; here, the value on the vertical axis represents the number of instances solved as a function of the running time (in seconds) indicated on the horizontal axis.

With Formulation 4, 44 instances are solved in less than 1 second and all 50 instances are solved within 3 seconds. On the other hand, with Formulation 1, only 11 instances are solved under 3 seconds, 47 instances under 120 seconds, and all the instances under 230 seconds. As the running time varies significantly for the different formulations, even for small instances, a time limit of 2 minutes was set in order to test additional instances. Table 4.3 displays the number of instances that were solved within the time limit among 50 instances with $K = 3$ and $n \in \{40, 60, 80, 100\}$.

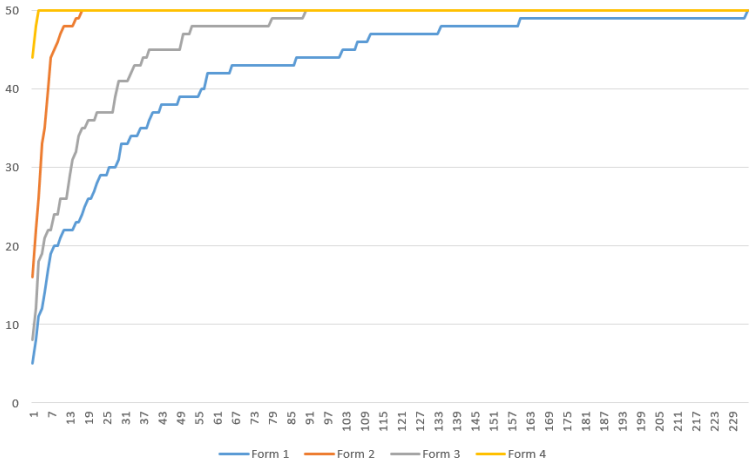


Figure 4.12: Comparison of running time for Formulations 1–4 when $n = 40$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

Table 4.3: Comparison of formulations: number of instances solved within 2 minutes

n	40	60	80	100
Formulation 1	47	8	0	0
Formulation 2	50	50	25	0
Formulation 3	50	18	0	0
Formulation 4	50	50	49	47

Surprisingly, in spite of its weaker relaxation and of its larger size, Formulation 2 often turns out to be more efficient than Formulation 3. This seems to be at least partially due to the way Gurobi handles different types of constraints. Indeed, when the preprocessing steps and the cut generation parameters of the solver are disabled, the running times of Formulations 2 and 3 turn out to be worse, but very close to each other.

All in all, however, the results clearly suggest that, under our experimental setting, Formulation 4 is the most efficient one, certainly because it is compact and has a relatively good LP relaxation. In fact, interestingly, the quality of the relaxation does not seem to deteriorate when the size of the instances increases. This claim is supported by the values of the integrality gap Gap^4_{LP} reported in Table 4.4, where each line refers to 50 instances of a given size (all these instances are solved to optimality within 10 minutes).

Table 4.4: Integrality gap for Formulation 4 ($K = 3$)

n	mean Gap^4_{LP}	min Gap^4_{LP}	max Gap^4_{LP}
40	11.18	0.00	33.15
60	12.43	2.16	27.49
80	12.25	3.84	23.79
100	10.35	0.00	21.39

In view of these observations, we restrict our attention to Formulation 4 in the sequel. When $K = 2$, we will see in the next section that large instances of the L-exchange problem can be solved rather easily. When $K = 3$, however, the problem may become much harder. For example, when $n = 120$, Gurobi solves Formulation 4 in 463 seconds on average and can solve 41 of 50 instances in less than 10 minutes. When $n = 130$, the average running time doubles (990 seconds) and only 19 instances are solved in less than 10 minutes. Clearly, more work may be needed in the future to solve large instances efficiently. But for now, we prefer to turn to a comparison between stable and L-stable exchanges.

4.6.3 Comparison with stable exchanges

As underlined in Section 4.4, the maximum size of an L-stable exchange (or an L-kernel) may potentially be (much) larger than the maximum size of a stable exchange (or a kernel). In particular, nonempty L-stable exchanges may exist even in situations where there is no stable exchange.

Moreover, Klimentova et al. (2023) have observed that, in spite of the theoretical complexity of the problem (see Biró and McDermid (2010) and Huang (2010)), computing maximum stable exchanges is relatively easy in practice. Within a time limit of 1 hour, they solve all instances with $K = 2$, all instances up to $n = 100$ when $K = 3$, and all instances up to $n = 50$ when $K = 4$ (on a relatively fast computer). By contrast, we are not aware of any numerical work regarding the computation of L-stable exchanges.

We have therefore performed an experimental comparison of the solution of instances of the maximum stable exchange problem and of the maximum L-stable exchange problem using formulation (4.2)-(4.4) and formulation (4.4), (4.11), (4.7) (Formulation 4), respectively, with the same objective function (4.12).

Let us first briefly comment on the running time of the IP solver for each problem. Figure 4.13 and Figure 4.14 display the performance profiles for both problems on two sets of 50 instances with $K = 3$, $n = 80$ and $n = 100$ respectively. We see that the running time never exceeds 480 seconds, and is actually much shorter for most instances. Moreover, there is no clear

dominance pattern regarding the practical difficulty of solving these two models.

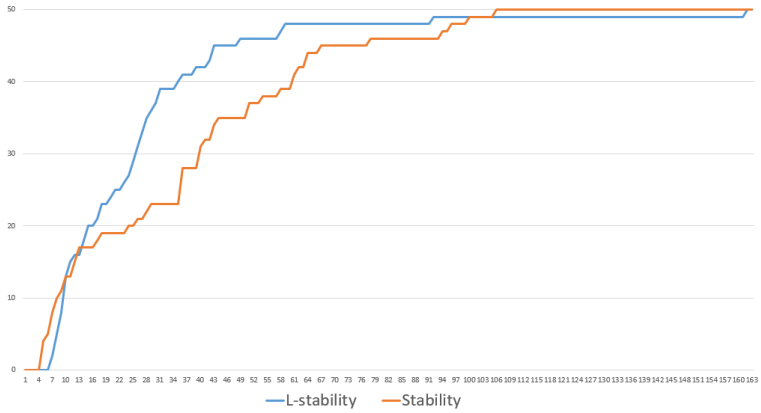


Figure 4.13: Running time for stable exchanges and L-stable exchanges, $n = 80$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

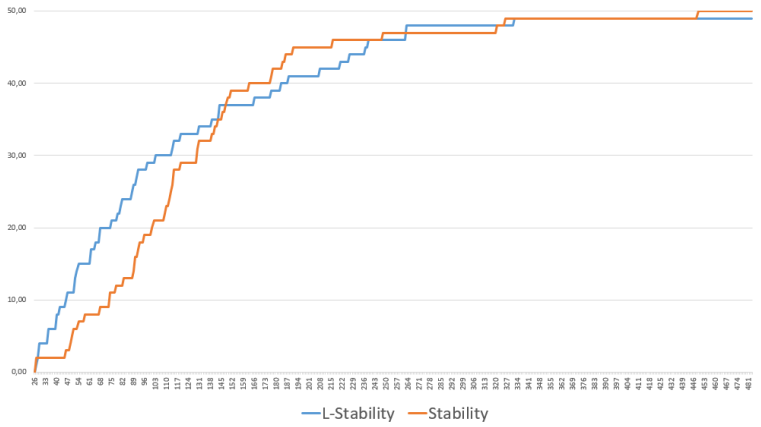


Figure 4.14: Running time for stable exchanges and L-stable exchanges, $n = 100$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

Let us next examine the features of the optimal solutions.

For $K = 3$, all the instances up to $n = 180$ have a stable exchange. Likewise for all the instances up to $n = 80$ when $K = 4$. Moreover, for these instances, the maximum size of a stable exchange and of an L-stable exchange is always the same (in spite of Fact 4 and Example 6, which show that equality does not hold in general).

By contrast, when $K = 2$, many instances in our dataset do not have a stable exchange. Table 4.5 provides information about a collection of 600 instances with various values of n ranging between 50 and 400, with the same notations as in Table 4.1. One can readily observe that for a fixed value of n, the number of cycles in the compatibility digraphs can vary significantly, and this translates into even more variance in the number of arcs in the blocking digraphs.

Table 4.5: Size parameters of instances with $K = 2$

n	$\|V\|$	$\|A\|$	$\|V^*\|$	min$\|V^*\|$	max$\|V^*\|$	$\|A^*\|$	min$\|A^*\|$	max$\|A^*\|$
50	53	782	116	52	193	1394	316	2783
70	74	1522	217	143	348	3636	1693	7851
90	95	2520	365	220	577	7938	3661	16058
110	116	3736	544	337	838	14530	5378	26005
130	137	5183	749	479	1099	23487	10379	37859
150	158	6938	997	640	1337	35967	17765	52686
170	179	8892	1273	863	1676	51791	28275	75557
200	210	12104	1704	1255	2250	80657	49880	128648
250	263	19191	2718	1814	3582	162990	83531	251676
300	315	27554	3924	2686	5152	282393	145333	439438
350	368	37697	5361	4176	6890	451272	296268	677069
400	420	49019	6948	5557	8784	667607	467942	962287

Table 4.6 synthesizes some results of our computational experiments for the 600 instances mentioned above. Note that 72 of these instances do not have a stable exchange. The left part of Table 4.6 refers to the maximum stable exchange problem and the right part refers to the maximum L-stable exchange problem. In detail, for each value of n:

- a and a_L are the average optimal values for each problem; the averages are computed over those instances which have a stable exchange or a nonempty L-stable exchange, respectively;
- *prep* and $prep_L$ are the average times (in seconds) required to construct the models;
- *solve* and $solve_L$ are the average times (in seconds) required to solve the models;
- T and T_L are the average total times (in seconds) required to handle each problem; e.g., $T = prep + solve$;
- ϕ is the number of instances that do not have a stable exchange among 50 instances with the same value of n;
- ϕ_L is the number of instances that do not have a nonempty L-stable exchange among 50 instances with the same value of n.

Table 4.6: Results for instances with $K = 2$

n	a	*prep*	*solve*	T	ϕ	a_L	$prep_L$	$solve_L$	T_L	ϕ_L
50	23.3	0.0	0.0	0.0	2	22.9	0.0	0.0	0.0	0
70	32.9	0.0	0.0	0.0	3	32.1	0.0	0.0	0.0	0
90	43.8	0.1	0.0	0.1	6	42.7	0.1	0.0	0.1	0
110	56.2	0.1	0.0	0.1	3	54.7	0.1	0.0	0.2	0
130	67.1	0.2	0.0	0.2	2	65.8	0.2	0.0	0.3	0
150	78.6	0.2	0.0	0.2	2	77.4	0.4	0.1	0.5	0
170	90.3	0.3	0.0	0.3	8	82.7	0.6	0.1	0.7	0
200	105.8	0.5	0.1	0.6	5	99.6	0.8	0.2	1.0	0
250	137.8	1.0	0.2	1.2	8	122.5	1.4	0.5	1.9	0
300	167.6	1.9	0.5	2.4	4	154.9	2.6	0.9	3.5	0
350	198.9	3.0	0.9	3.9	11	164.6	4.0	1.4	5.4	0
400	230.4	4.6	1.4	6.0	18	167.6	5.9	2.2	8.2	1

A few observations can be made from Table 4.6. First, the average running times are extremely low. They appear to be a bit higher for the L-stable exchange problem, but the performance profiles show that no clear conclusion can be drawn in this respect. (When $K = 2$, the stable exchange problem is equivalent to the stable roommate problem with incomplete preferences, which is polynomially solvable; Irving (1985). However, we have not exploited this property in our experiments.)

More interestingly, just as in the cases $K = 3$ and $K = 4$, none of the random instances we tested for $K = 2$ has an L-stable exchange larger than the maximum stable exchange, *provided that there is a stable exchange* (this, in spite of Fact 4). However, among the 72 instances in Table 4.6 which do not have a stable exchange, 71 have a nonempty L-stable exchange.

The average optimal values of the two problems differ, but one should remember that the averages are computed over those instances which have a stable exchange or a nonempty L-stable exchange, respectively. So, the differences are due solely to the instances that do not have a stable solution but have a nonempty L-stable solution. The magnitude of the differences indicates that for such instances, the size of the maximum L-stable exchange is both significantly larger than zero, and significantly smaller than for the instances which have a stable exchange. For example, when $n = 200$, the average size of a stable exchange (if there is one) is 105.8, whereas the average size of a maximum L-stable exchange is 44.4 for the five remaining instances. More generally, Table 4.7 displays information pertaining only to those instances that have no stable exchange, but do have a nonempty L-stable exchange. One can observe that the running times are close to the ones presented in Table 4.6, which shows that the instances without stable exchange are not particularly hard to solve in terms of running time.

Table 4.7: Results for instances with $K = 2$ having a nonempty L-stable exchange and no stable exchange

n	$\phi - \phi_L$	a_L	$prep_L$	$solve_L$	T_L
50	2	13.0	0.0	0.0	0.0
70	3	19.3	0.0	0.0	0.0
90	6	34.7	0.1	0.0	0.1
110	3	31.3	0.2	0.0	0.2
130	2	34.0	0.2	0.0	0.2
150	2	53.0	0.4	0.1	0.5
170	8	42.8	0.5	0.1	0.7
200	5	44.4	0.8	0.3	1.0
250	8	42.0	1.4	0.5	1.8
300	4	8.5	2.8	0.8	3.6
350	11	42.9	4.1	1.5	5.6
400	17	59.4	6.0	2.2	8.2

4.7 Local strong stability

4.7.1 Definitions

In Klimentova et al. (2023), the authors define another type of stability, namely *strong stability*, that we now proceed to introduce by adapting the definitions of Section 4.3.1.

Definition 19. Let $\mathcal{M}$ be an exchange, let $u \in \mathcal{C}_K(G) \setminus \mathcal{M}$ be a cycle not contained in $\mathcal{M}$, and let $i \in V(u)$.

We say that vertex i *is indifferent between* the cycle u and the exchange $\mathcal{M}$ if $i \in V(\mathcal{M})$, $(k, i) \in A(u)$, $(k', i) \in A(\mathcal{M})$, and i is indifferent between k and k'.

We say that i *weakly prefers* u to $\mathcal{M}$ if either i prefers u to $\mathcal{M}$ or i is indifferent between u and $\mathcal{M}$.

Definition 20. A *weakly blocking cycle* for an exchange $\mathcal{M}$ is a cycle $u \in \mathcal{C}_K(G) \setminus \mathcal{M}$ such that

- each vertex in $V(u)$ weakly prefers u to $\mathcal{M}$, and
- if u intersects $\mathcal{M}$, then at least one vertex $i \in V(u) \cap V(\mathcal{M})$ prefers u to $\mathcal{M}$.

When $\mathcal{M} = \{v\}$, we simply say that u is weakly blocking for v.

Definition 21. An exchange $\mathcal{M}$ is *strongly stable* if there is no weakly blocking cycle for $\mathcal{M}$ in $\mathcal{C}_K(G)$.

Similarly to what we did in Section 4.3, we now propose a seemingly new concept of *locally weakly blocking cycles* and *locally strongly stable exchanges*.

Definition 22. A *locally weakly blocking cycle*, or *LW-blocking cycle*, for an exchange $\mathcal{M}$ is a weakly blocking cycle for $\mathcal{M}$ that intersects $\mathcal{M}$. When $v \in \mathcal{C}_K(G)$, we denote by $\mathcal{B}_W(v)$ the set of all LW-blocking cycles of the exchange $\{v\}$ (or for short, of the cycle v).

Note that for an exchange $\mathcal{M}$, an L-blocking cycle is an LW-blocking cycle, while the converse is not necessarily true. In particular, for a cycle v, $\mathcal{B}(v) \subseteq \mathcal{B}_W(v)$ but in general, $\mathcal{B}(v) \neq \mathcal{B}_W(v)$.

Example 7. Consider again the digraph of Figure 4.4 with $K = 3$ and exactly two cycles in $\mathcal{C}_K(G)$, namely, $v = (1, 2, 3, 1)$ and $u = (2, 3, 4, 2)$. One can check that $\mathcal{B}_W(v) = \{u\}$. Indeed, vertex 2 prefers its predecessor in u to its predecessor in v, vertex 3 has the same predecessor in both cycles, and vertex 4 is not in $V(v)$. On the other hand, $\mathcal{B}(v)$ is empty since vertex 3 does not prefer u to v and hence, u is not L-blocking for v.

Definition 23. An exchange $\mathcal{M}$ is called *locally strongly stable*, or *LS-stable*, if there is no LW-blocking cycle for $\mathcal{M}$ in $\mathcal{C}_K(G)$.

Let us clarify the relations between the concepts introduced so far.

Theorem 35. *Let $\mathcal{M}$ be an exchange and let v be a cycle not contained in $\mathcal{M}$.*
(a) *If v is blocking for $\mathcal{M}$, then v is weakly blocking for $\mathcal{M}$.*
(b) *If v is locally weakly blocking for $\mathcal{M}$, then v is weakly blocking for $\mathcal{M}$.*
(c) *If v is locally blocking for $\mathcal{M}$, then v is both blocking and locally weakly blocking for $\mathcal{M}$.*
(d) *If $\mathcal{M}$ is strongly stable, then $\mathcal{M}$ is both stable and locally strongly stable.*
(e) *If $\mathcal{M}$ is stable, then $\mathcal{M}$ is locally stable.*
(f) *If $\mathcal{M}$ is locally strongly stable, then $\mathcal{M}$ is locally stable.*

Proof. All implications directly follow from the definitions. In particular, implication (d) follows from (a) and (b), implications (e) and (f) follow from (c). □

None of the implications can be reversed in Theorem 35. Moreover, stable exchanges and locally strongly stable exchanges are, in general, unrelated. These points are illustrated by the next example.

Example 8. The exchange $\{u_4\}$ in Example 2 is LS-stable, but not stable. On the other hand, the exchange $\{v\}$ in Example 7 is stable, but not LS-stable (and hence, not strongly stable). The latter observation also clarifies the fact that strong stability differs from stability even when the preference

relation between each vertex and its in-neighbors is strict (no indifference).

Finally, we need one last definition before we turn to alternative characterizations of (local) strong stability.

Definition 24. Two cycles u and v are *strong friends* if they have a nonempty intersection, if u is not LW-blocking for v, and v is not LW-blocking for u, that is, if $u \notin \mathcal{B}_W(v)$ and $v \notin \mathcal{B}_W(u)$. We denote by $\mathcal{F}_S(v)$ the set of strong friends of a cycle v.

Clearly, $u \in \mathcal{F}_S(v)$ if and only if $v \in \mathcal{F}_S(u)$. Two cycles are strong friends either when they share at least two vertices, one of which prefers v while the other one prefers u, or when all vertices in $V(u) \cap V(v)$ are indifferent between their respective predecessors in u and in v.

Example 9. Figure 4.15 illustrates both situations. For $K = 4$, cycles $u_1 = (1, 2, 3, 4, 1)$ and $v_1 = (1, 3, 5, 6, 1)$ are strong friends since they share vertices 1 and 3, and since 1 prefers v_1 whereas 3 prefers u_1. Moreover, cycles $u_2 = (7, 8, 9, 7)$ and $v_2 = (8, 9, 10, 8)$ are strong friends since they share vertices 8 and 9 and both vertices are indifferent between the two cycles.

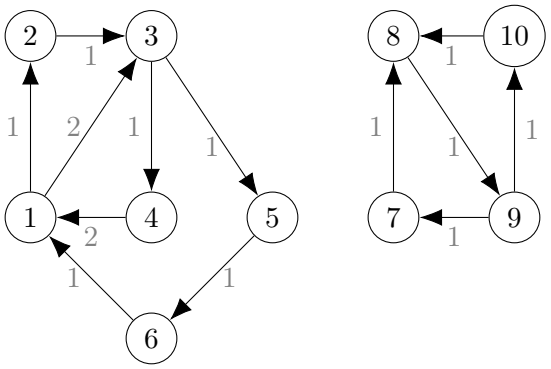


Figure 4.15: Illustration of cycles being strong friends

4.7.2 Characterizations and formulations

With the above definitions, most of the characterizations and formulations obtained in previous sections for stable and L-stable exchanges can be adapted in a rather straightforward way for strongly stable and LS-stable exchanges. In particular, the statements of Lemma 1, Lemma 2, Theorem 28 and Theorem 29 can be modified with $\mathcal{B}(v)$ replaced by $\mathcal{B}_W(v)$ and $\mathcal{F}(v)$ replaced by $\mathcal{F}_S(v)$ for all $v \in \mathcal{C}_K(G)$, as follows.

Lemma 3. Two cycles u, v intersect each other if and only if $u \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)$ or $v \in \mathcal{B}_W(u) \cup \mathcal{F}_S(u)$.

Lemma 4. Let $\mathcal{M}$ be an exchange and let v be a cycle not contained in $\mathcal{M}$. The following statements are equivalent:
(i) there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)$;
(ii) v is not weakly blocking for $\mathcal{M}$;
(iii) v intersects $\mathcal{M}$ and v is not locally weakly blocking for $\mathcal{M}$.

Proof. (i) $\Rightarrow$ (ii). Let $w \in \mathcal{M}$. If $w \in \mathcal{B}_W(v)$, then $V(v) \cap V(w)$ is not empty, and at least one vertex in the intersection prefers w to v. It follows that v is not blocking for $\mathcal{M}$.

If $w \in \mathcal{F}_S(v)$, then by definition, $V(v) \cap V(w)$ is not empty and $v \notin \mathcal{B}_W(w)$. Hence, either there is a vertex in $V(v) \cap V(w)$ which prefers w to v, or every vertex in $V(v) \cap V(w)$ is indifferent between v and w. So, once again, v is not weakly blocking for $\mathcal{M}$.

(ii) $\Rightarrow$ (iii). Indeed, v is weakly blocking for $\mathcal{M}$ if and only if either $V(v) \cap V(\mathcal{M}) = \emptyset$ or v is locally weakly blocking for $\mathcal{M}$.

(iii) $\Rightarrow$ (i). If v intersects $\mathcal{M}$, but v is not LW-blocking for $\mathcal{M}$, then there exists a cycle $w \in \mathcal{M}$ such that either some vertex in $V(v) \cap V(w)$ prefers w to v, or every vertex in $V(v) \cap V(w)$ is indifferent between v and w. In particular, $v \notin \mathcal{B}_W(w)$. But then, by Lemma 3, $w \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)$. □

The next two theorems are now immediate consequences of Lemma 4.

Theorem 36. *For an exchange $\mathcal{M}$, the following conditions are equivalent:*
(a) *$\mathcal{M}$ is strongly stable;*
(b) *for each cycle $v \notin \mathcal{M}$, there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)$.*

Theorem 37. *For an exchange $\mathcal{M}$, the following conditions are equivalent:*
(a) *$\mathcal{M}$ is locally strongly stable;*
(b) *for each cycle $v \notin \mathcal{M}$, if v intersects $\mathcal{M}$, then there exists $w \in \mathcal{M}$ such that $w \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)$.*

By analogy with Section 4.4, we introduce the *weak blocking digraph* $G^{**} = (V^*, A^{**})$ associated with G, where

- $V^* = \mathcal{C}_K(G)$;
- $A^{**} = \{(v, w) : w \in \mathcal{B}_W(v) \cup \mathcal{F}_S(v)\}$.

Theorem 38. *For a digraph G and its weak blocking digraph G^{**}, the strongly stable exchanges of G are exactly the kernels of G^{**}, and the locally strongly stable exchanges of G are exactly the L-kernels of G^{**}.*

Proof. The statement follows from Theorem 36 and Theorem 37 (compare with Theorem 30 and Theorem 31). □

4.7.3 Numerical tests for LS-stable exchanges

This section presents the results of numerical experiments based on the following formulation of maximum LS-stable exchanges:

$$\max \sum_{v \in V^*} |V(v)|\, y_v \tag{4.13}$$

$$\sum_{v \in V^* : i \in V(v)} y_v \leq 1 \qquad \forall i \in V \tag{4.14}$$

$$\sum_{u:(u,v)\in A^{**},(v,u)\notin A^{**}} y_u \leq \Delta^-(v) \sum_{w:(v,w)\in A^{**}} y_w \qquad \forall v \in V^* \tag{4.15}$$

$$y_v \in \{0,1\} \qquad \forall v \in V^*. \tag{4.16}$$

where $\Delta^-(v) := |\{u : (u,v) \in A^{**}, (v,u) \notin A^{**}\}|$ (compare with $\delta^-(v)$ in constraints (4.11)).

Interestingly, and unlike the case of (local) stability, several instances do not have a strongly stable exchange when $K = 3$ and $K = 4$, but many of those instances do have a locally strongly stable exchange! For example, when $K = 3$ and $n \leq 100$, 35 instances (out of 450) do not have a strongly stable exchange. Table 4.8 provides information about the associated graphs. The notations are the same as in Table 4.5, and additionally:

- $|A^{**}|$ is the average number of arcs in the weak blocking digraphs; the next two columns, $\min|A^{**}|$ and $\max|A^{**}|$ show the minimum and the maximum number of arcs in 50 weak blocking digraphs for each value of n.

Table 4.8: Size parameters of instances of with $K = 3$

n	$\lvert V\rvert$	$\lvert A\rvert$	$\lvert V^*\rvert$	**min**$\lvert V^*\rvert$	**max**$\lvert V^*\rvert$	$\lvert A^{**}\rvert$	**min**$\lvert A^{**}\rvert$	**max**$\lvert A^{**}\rvert$
20	21	118	64	5	178	1924	6	9078
30	32	285	233	45	580	16600	371	66787
40	42	472	452	27	1052	53767	109	186104
50	53	782	1013	330	2064	206148	22292	619907
60	63	1081	1484	521	3035	399117	50755	1167418
70	74	1522	2521	1055	4764	961696	208876	2584228
80	84	1930	3495	1790	6404	1686656	469639	4609734
90	95	2520	5409	2554	11527	3477577	1020955	11826362
100	105	3020	6895	3783	13587	5296861	1637136	15697727

Comparing Table 4.8 and Table 4.5, a first observation is that the average number of cycles increases considerably when K goes from 2 to 3. For example, when $n = 90$, $|V^*|$ goes up (on average) from 365 to 5409. But the growth of $|A^{**}|$ with respect to $|A^*|$ is even more spectacular: when $n = 90$

and $K = 2$, the average value of $|A^*|$ is 7938, whereas the average value of $|A^{**}|$ is almost 3.5×10^6 when $K = 3$.

Table 4.9 displays some results of the computational experiments. The left part of the table refers to the maximum strongly stable exchange problem and the right part refers to the maximum LS-stable exchange problem. For each value of n,

- a_S and a_{LS} are the average optimal values for each problem; the averages are computed over those instances which have strongly stable or nonempty LS-stable exchanges, respectively;
- $prep_S$ and $prep_{LS}$ are the average times (in seconds) required to construct the models;
- $solve_S$ and $solve_{LS}$ are the average time (in seconds) required to solve the models;
- ϕ_S is the number of instances that do not have a strongly stable exchange among 50 instances with the same value of n;
- ϕ_{LS} is the number of instances that do not have a nonempty LS-stable exchange among 50 instances with the same value of n.

Table 4.9: Results for instances with $K = 3$

n	a_S	$prep_S$	$solve_S$	ϕ_S	a_{LS}	$prep_{LS}$	$solve_{LS}$	ϕ_{LS}
20	8.4	0.0	0.0	2	8.3	0.0	0.0	1
30	14.4	0.1	0.0	1	14.4	0.1	0.0	0
40	19.2	0.2	0.0	3	18.0	0.3	0.1	0
50	25.1	0.8	0.3	4	23.0	1.2	0.4	0
60	29.9	1.4	0.7	4	28.4	2.5	1.0	1
70	36.4	3.5	3.9	3	35.1	6.1	2.5	1
80	41.1	5.7	14.8	9	36.1	11.4	5.9	1
90	49.1	11.8	41.9	5	43.8	23.5	14.9	3
100	53.9	18.8	84.3	4	51.3	38.4	22.1	1

For the instances that we considered, the average running time turns out to be slightly higher for strongly stable exchanges than for locally strongly stable exchanges. But a more interesting observation is that a significant number of instances (35) do not have a strongly stable exchange, whereas only 8 instances do not have a nonempty LS-stable exchange. For example, when $n = 80$, $\phi_S = 9$ instances (out of 50 tested) do not have any strongly stable exchange, but only one of them does not have a nonempty LS-stable exchange ($\phi_{LS} = 1$). Table 4.10 shows the results obtained for the instances which do not have a strongly stable exchange, but do have a nonempty LS-stable exchange. Note for example that, when a strongly stable exchange exists for the instances with $n = 80$, its average cardinality is $a_S = 41.1$. In contrast, for those 8 instances which have no strongly stable exchange,

but which have a nonempty LS-stable one, the average size of an optimal LS-stable exchange is 14.6. In spite of this relatively small value, the average size of a nonempty maximum LS-stable exchange over all 50 instances with $n = 80$ is $a_{LS} = 36.1$ (Table 4.9), not much smaller than a_S. This suggests once again that locally strongly stable exchanges may provide a meaningful and fruitful alternative when stable exchanges do not exist.

Table 4.10: Results for instances with $K = 3$

n	$\phi_S - \phi_{LS}$	a_{LS}	$prep_{LS}$	$solve_{LS}$
20	1	2.0	0.0	0.0
30	1	7.0	0.3	0.1
40	3	6.6	0.6	0.3
50	4	4.5	1.4	0.5
60	3	7.3	1.6	0.7
70	2	25.0	5.6	2.1
80	8	14.6	10.7	7.4
90	2	6.5	16.7	8.1
100	3	28.3	31.2	11.9

4.8 Kernels and L-kernels of random digraphs

The numerical tests conducted in previous sections consisted in computing maximum kernels or local kernels in (weak) blocking digraphs associated with kidney exchange compatibility digraphs. Since it appears that no numerical results concerning L-kernels have been published in the past, we have completed our experiments by computing maximum kernels and L-kernels of randomly generated digraphs $D = (W, E)$ using the kernel IP formulation

$$\begin{aligned}
&\max \sum_{v \in W} y_v \\
&y_u + y_v \leq 1 && \forall (u,v) \in E \\
&1 \leq y_v + \sum_{w:(v,w)\in E} y_w && \forall v \in W \\
&y_v \in \{0,1\} && \forall v \in W
\end{aligned}$$

and the L-kernel IP formulation

$$\begin{aligned}
&\max \sum_{v \in W} y_v \\
&y_u + y_v \leq 1 && \forall (u,v) \in E \\
&y_u \leq \sum_{w:(v,w)\in E} y_w && \forall (u,v) \in E \\
&y_v \in \{0,1\} && \forall v \in W.
\end{aligned}$$

The digraphs $D = (W, E)$ are characterized by their size $|W| = n$ and their density d; each arc (i, j), $i \neq j$, is present in E with probability d independently of the other arcs. Each line of Table 4.11 gives the following information for 50 random instances with parameter valurs (d, n):

- a is the average size of the maximum kernel among the instances which have a kernel;
- a_L denotes the average size of the maximum L-kernel among the instances which have a nonempty L-kernel;
- ϕ is the number of instances that do not have a kernel;
- ϕ_L is the number of instances that do not have a nonempty L-kernel.

Table 4.11: Kernel vs. local kernel

d	n	a	ϕ	a_L	ϕ_L
0.01	50	35.2	1	35.0	0
	100	57.4	5	57.1	0
	150	73.2	11	71.6	0
	200	86.1	17	81.5	0
	250	97.2	27	88.3	0
0.02	50	28.0	3	27.8	0
	100	43,1	10	41.3	0
	150	53.5	20	45.2	0
	200	59.9	23	41.3	2
	250	66.8	23	44.6	1
0.05	50	19.4	14	17.1	1
	100	27.7	16	23.0	5
	150	31.7	14	30.2	13
0.10	50	14.0	13	12.2	6
	100	17.7	12	17.7	12
	150	20.2	5	20.2	5

A few observations can be made. First, out of 500 instances with small density ($d = 0.01$ or 0.02), only three do not have a nonempty L-kernel, whereas 141 do not have a kernel. As the density increases, more instances fail to have an L-kernel of size greater than zero.

Second, for a fixed number of vertices n, the size of the kernels and L-kernels tends to decrease as the density of the digraphs increases. This may be simply due to the independence constraints. (With regard to this observations, note that the blocking digraphs G^* considered in Section 4.6.3 have a rather small density, whereas the weak blocking digraphs G^{**} in Section 4.7.3 have a higher one.)

Finally, as in previous experiments, the difference between the average optimal values a and a_L is explained by the instances that do not have a kernel but have a nonempty L-kernel. This difference is relatively small, meaning that the L-kernels, when they are not empty, are of comparable size with the kernels, when the latter exist. Actually, in spite of Fact 4 and Example 6, none of the random instances considered in Table 4.11 features a maximum kernel of size, say, κ and a maximum L-kernel of size strictly larger than κ. But a couple of such instances occurred in our experiments when $d = 0.01$ and $n = 300$ or $n = 350$.

4.9 Conclusions and perspectives

In this chapter, we have introduced a new concept of local stability for kidney exchanges. We believe this concept to be quite natural in the KEP setting but surprisingly, it has apparently not been investigated earlier. The concept extends in a similar way to strongly stable exchanges.

We have also made explicit the link between (local) stable exchanges and (local) kernels in an associated digraph. This leads to integer programming formulations which can be optimized by a generic solver for graphs of moderate sizes, in spite of the NP-hardness of (local) kernels. The experimental results show that nonempty L-stable exchanges frequently exist in digraphs which do not have a stable exchange.

Our contributions open various directions for future research. First, it would be interesting to investigate the relevance of local stability and local kernels for different classes of matching problems, beyond kidney exchanges. Next, the complexity of computing maximum L-stable exchanges is currently open (our complexity result only applies to L-kernels in arbitrary digraphs). Finally, even though local kernels have been previously considered in graph theory, their properties have barely been investigated so far. A deeper understanding of these properties may be useful in order to efficiently compute maximum L-kernels in large-size digraphs.

Chapter 5

Generation of delisting dates for the simulation of Eurotransplant's allocation mechanisms

The content of this chapter is based on joint work with Yves Crama and Bart Smeulders. It has not been submitted for publication.

Contents

5.1 Introduction

Nowadays, patients with end-stage renal disease (ESRD) have several treatment options: dialysis, or kidney transplant from a deceased donor, or kidney transplant from a living donor.

The most common treatment for ESRD is dialysis, it is also the least preferred option for the patients and for the medical teams; it is costly, and the time consumption of dialysis or the dietary restrictions for example deeply impact the quality of life of the patient.

If a patient has the opportunity of receiving a transplant from a living donor, such a transplant is the preferred treatment regarding the chance of success and the comfort of life of the patients (Roth et al., 2004). Some problems linked to that possibility of treatment have been studied theoretically in previous chapters of this thesis when the ESRD patients have a willing donor but the pair is medically incompatible.

A deceased donor kidney transplant occurs when a kidney from a person who recently died is removed and transplanted into a patient, also named recipient, whose kidneys are no longer functional. A human body can function normally with only one healthy kidney; hence, a single deceased donor can potentially generate two kidney transplants. This chapter will focus on this kind of treatment and more precisely, on a European program allocating deceased donor kidneys to a waiting list of patients under a specific mechanism, that is, the so-called ETKAS mechanism of Eurotransplant.

In 2019, Eurotranplant initiated a research project with Bart Smeulders and Frits C.R. Spieksma to simulate and to compare numerically the possible outcomes of the ETKAS mechanism under alternative allocation rules. In collaboration with Bart Smeulders, we contributed to one specific modeling aspect of this project. Namely, when given the actual waiting list of patients recorded by Eurotransplant during a period of time, some data required to conduct the simulation was missing: for those patients who had been transplanted under the ETKAS mechanism, the total time during which they would have remained in an appropriate medical condition to be transplanted (had they not been transplanted already) was unknown. Indeed, in the absence of a transplantation, these patients would not have remained forever on the waiting list: at some point, the evolution of their medical condition or their death would have made them leave the waiting list.

We contributed to the simulation of the missing data. The current chapter provides an account of this work, as well as an attempt to validate the choice of the model adopted in the simulation. Section 5.2 briefly introduces Eurotransplant, the 2019 version of ETKAS, the second simulated mechanism, and the outline of the simulation conducted. Section 5.3 describes

the simulation of the missing data using survival analysis and the Kaplan-Meier method, and Section 5.4 provides some insights into the validity of the method.

5.2 Context

Transplantation treatments are often managed by programs, especially deceased donor' transplantation. Each program has its own ethical and legal regulations. In Europe, national and international programs such as Scandiatransplant or Eurotransplant (ET) exist. Eurotransplant is an international non-profit European organization founded in 1967 to coordinate organ transplantations. Currently, 8 countries constitute the Eurotransplant network: Austria, Belgium, Croatia, Germany, Hungary, Luxembourg, the Netherlands, and Slovenia. ET manages transplants of hearts, lungs, kidneys, livers, pancreas, from deceased donors but also from living donors. For example, in 2019, 6060 transplants from a deceased donor were coordinated by Eurotransplant, and 1300 transplants from living donors. Most of the transplants from deceased donors are kidney transplants as shown in Figure 5.1 below (source: https://statistics.eurotransplant.org/ on November, 2nd 2022) :

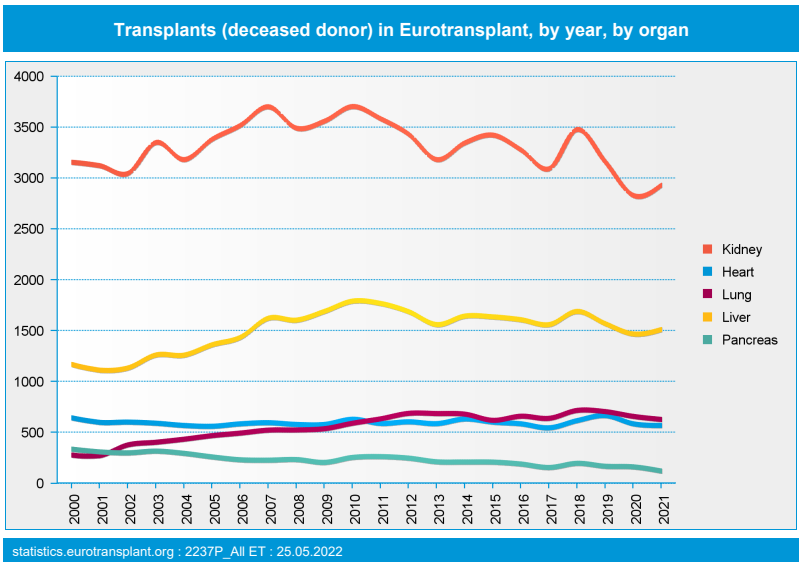


Figure 5.1: Transplants (deceased donor) in Eurotransplant, by year, by organ

Currently, when a kidney donor is available in the ET program, various kidney allocation mechanisms are employed depending on donor characteristics and on the needs of prioritized patient groups:

- The **Eurotransplant Senior Program** (ESP) allocates kidneys from donors over the age of 65 to patients in the same age group.
- The **Acceptable Mismatch** (AM) mechanism serves patients who are incompatible with the vast majority of donors and thus receive priority for the compatible kidneys.
- The **Eurotransplant Kidney Allocation System** (ETKAS) serves all the others.

As explained in the introduction, our focus is on the ETKAS program. ETKAS was introduced in 1996 and is a point-based allocation system. Two-thirds of the kidneys handled by ET are allocated by ETKAS.

Since their introduction in 1996, the ETKAS allocation rules have been modified due to new insights, changes in the availability of organs, and the behavior of patients. In 2019, the ETKAS allocation system, to be called Current ETKAS, was the same as today. It can be described as follows.

When a new donor becomes available, a tiered system prioritizing certain patients based on donor criteria is established. The tiers are as follows: patients perfectly compatible with the donor first (Tier 1); then if the donor age is less than 16, pediatric patients (Tier 2); and finally, the other patients (Tier 3). Within each tier, given the donor characteristics and the patient characteristics, points are allocated to each patient under certain criteria (identical for each tier). Then, the donor's kidneys are allocated to the patients with the largest number of points. The reader can find the detailed criteria in the Eurotransplant Manual available online (Eurotransplant (2023)).

In 2019, an alternative kidney allocation mechanism, to be called New ETKAS, was under consideration by the Eurotransplant kidney advisory committee. This new allocation mechanism would not be point-based but instead would rank recipients based on a number of criteria. The ranking criteria proposed for New ETKAS can be found in de Ferrante et al. (2019).

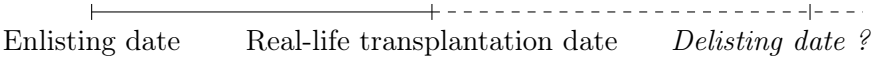
In order to test the effectiveness of the allocation system New ETKAS and to compare the results with those of the current ETKAS allocation system, simulations were conducted as described in de Ferrante et al. (2019).

Each simulation consists of two stages.

1. In the first stage, a waiting list of patients is constructed, and for each patient, a delisting date is randomly generated: it is the date at which the patient will leave the waiting list if he is not transplanted

in the simulated system. Note indeed that for patients who received a transplant in real life, the amount of time during which they would have remained eligible for a transplant is unknown and must be simulated. This is necessary to reflect the fact that at some point in time, their health condition does not allow them any longer to receive a transplant and they must leave the waiting list. If a delisting date is not generated for these patients, the waiting list will grow much bigger than what is observed in reality.

Patient data covers recipients on the waiting list at the start of the simulation, as well as patients arriving during the simulated period. A small number of recipients with incomplete data are removed. In total, 56,172 unique patients are used in the simulation.



The reader will find more details about the simulation of the delisting dates in Section 5.3.

2. In the second stage, the deceased donor kidney program is simulated both under the Current ETKAS allocation scheme and under the New ETKAS allocation scheme. The simulation period starts on November 1, 2004, and runs until June 2, 2016 (4,232 days). The inputs are:

   - The allocation scheme under consideration
   - A list L of all 56,172 recipients in the historical data during the simulation period,
   - A list P of all donors observed since May 27, 2013 (the day Hungary joined Eurotransplant). It contains 9,609 donors.
   - The real-life time points of a donor arrival during the simulation period. In total, there are 23,454 donor arrival time points between November 1, 2004, and June 2, 2016.

   At each time point t corresponding to a real-life arrival of a donor, the simulation performs the following steps (as illustrated in Figure 5.2):

   (a) Let L_t be the waiting list of recipients at the given time point t. For all patients in L, a patient is on L_t if their arrival date is earlier or equal to t and if their delisting date (possibly simulated, see Section 5.3) is after t.

   (b) Draw a donor randomly from the donors list P.

   (c) Allocate the donor's kidney(s) to patient(s) on L_t according to the allocation scheme under consideration.

(d) Simulate the outcome: a transplantation might fail and in this case, the patient is not removed from L. Otherwise, the patient is removed from L.

(e) Update the simulation data.

In total, for the 23,454 donor arrivals, nearly 39,000 kidneys are drawn from the donors pool and are allocated in each simulation.

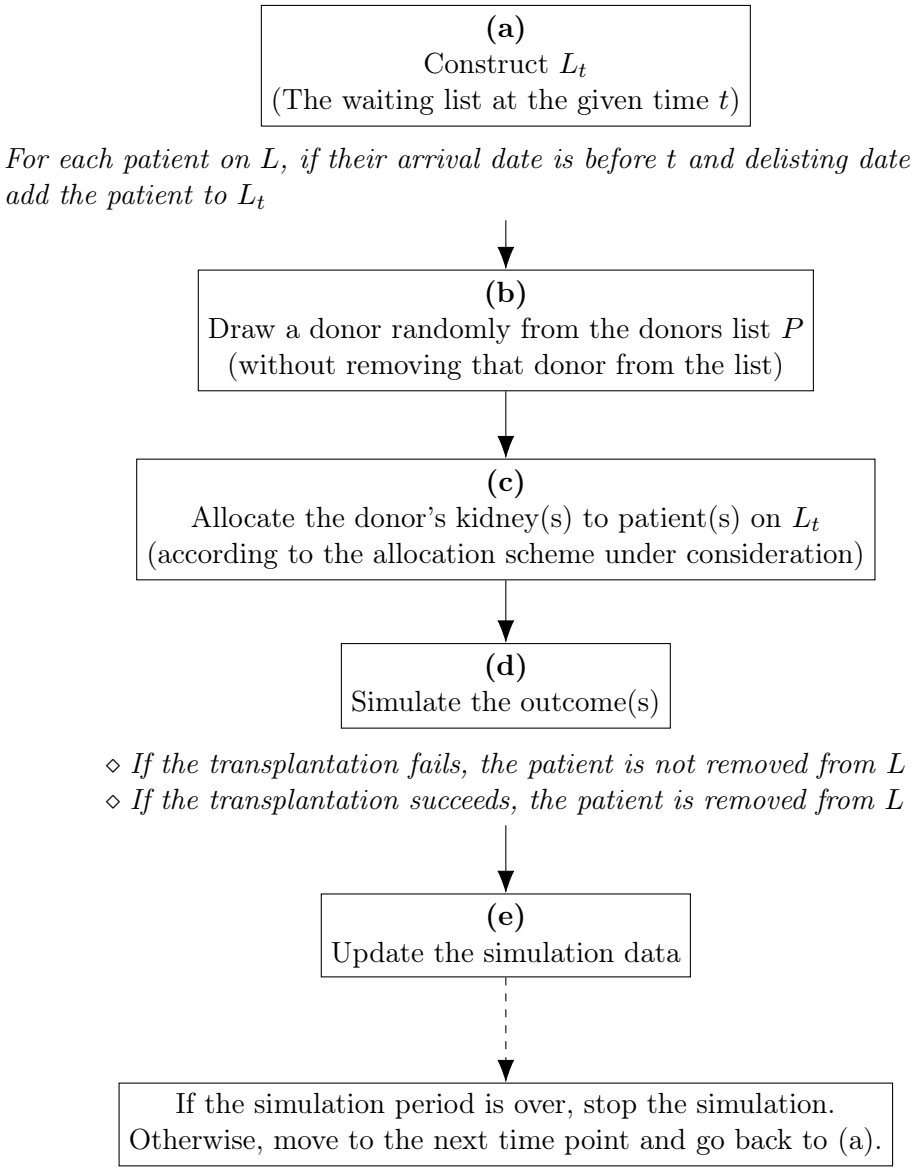


Figure 5.2: Visualization of the second stage of the simulation

Both for Current ETKAS and for New ETKAS, the simulation was run 10

times. Computation times were around 1 hour per run. Note that due to the complexity of the real-life system, not all aspects of the allocation process were simulated. The focus of this chapter is on the first stage. More details about the second stage of the simulation and the results are contained in de Ferrante et al. (2019).

5.3 Generation of delisting dates

5.3.1 Definitions and notations

The question of interest in this chapter concerns the first stage described above, whereby we must construct a waiting list of recipients and generate their delisting dates in case no transplantation takes place during the simulation. For the sake of clarity and of brevity, when we speak of *delisting* in the sequel, we always means *delisting without transplantation.*

In order to generate delisting dates, we relied on estimates of survival functions. By definition, for a subject and a random event of interest (e.g., the removal of the subject from a waiting list), the **survival function** $S^*(t)$ represents the probability that the time elapsed between the start of the observation of the subject until the realization of the event of interest exceeds t units of time. Given a sample of observations collected over a period of time, a main challenge is to compute a good estimate of $S^*(t)$. We generically denote such an estimate by $S(t)$ in the sequel, and with some abuse of terminology, call it again survival function, or survival curve. The task of computing $S(t)$ is complicated by the fact that some of the subjects in the sample might have been lost during the period of observation, or have not experienced the event of interest by the end of the period. Such subjects are called **censored**. Even though there is no information about the exact time of the event of interest for censored subjects, they still bring some partial information regarding the survival probabilities. Indeed, we know that the event of interest did not occur during their individual observation period, and this information can be integrated into the estimation of the survival function. (Note that in the study of the ETKAS system, the number of transplanted patients is much larger than the number of patients delisted without a transplant: 35,625 transplanted patients vs. 4,011 delisted patients, respectively. This means that much of the information about delisting dates is actually censored.)

In the context of our problem:

- ◇ The **observed sample** consists of the patients who have entered the ETKAS real-life waiting list between November 1, 2004 and June 2, 2016.
- ◇ The **unit of time** is one day.

- ◇ The **start of the observation period of each subject** is the day of the first dialysis. Indeed, medical teams have different strategies for deceased donor transplants. Because the wait for a kidney can be very long (it can take years), some teams decide to put a patient on the waiting list early, before the patient's condition actually requires a transplant, while other teams put the patient on the waiting list only when the patient's kidneys are no longer functioning and transplantation is required. Thus the date of the first dialysis is a better point of comparison than the date of entry on the waiting list. Note however that some recipients do not have a recorded first day of dialysis. For this small subset of patients, the first date of observation was taken to be the date of entry on the waiting list.
- ◇ The **event of interest** is the delisting of a patient. That is, the removal of the patient from the ETKAS waiting list without transplantation.
- ◇ **Censored subjects** are the patients who received a transplant in real life or patients still on the ETKAS waiting list at the end of the observation period.

In view of the above definitions, $S^*(t)$ is the probability that a patient will not be delisted within t days after their first date of observation or equivalently, that the patient will still be on the waiting list and eligible for transplantation t days after their first date of observation.

5.3.2 Kaplan-Meier method

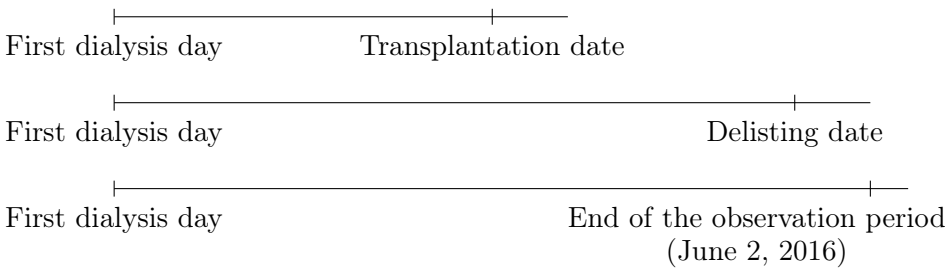
There are several ways to estimate survival functions and more generally to conduct a survival analysis, some methods being parametric, some non-parametric and some semi-parametric. In our case, as the data regarding the patients on the waiting list was not complete, we decided to use the Kaplan-Meier estimate, a non-parametric method (Kaplan and Meier, 1958). The data available for each patient are: identifier number on the waiting list, birth date, date of entry on the waiting list, date of the first dialysis, status under the current allocation mechanisms at the end of the observation period, date of exit (for the transplanted and delisted patients), country, blood type, and HLA. The Kaplan-Meier method is often cited as the simplest way of computing survival probabilities over time in spite of the difficulties associated with subjects or situations (Kaplan and Meier, 1958; Clark et al., 2003). It is also the most used method for survival analysis. However, the method has its drawbacks as explained in Section 5.4 and must be used cautiously. Survival analysis is often used for statistical analysis of medical studies (Jason T. Rich et al., 2010; Jager et al., 2008; Clark et al., 2003), rarely for simulation of missing data. The following section explains how we computed survival curves using the Kaplan-Meier estimate in the context of

our problem. Then Section 5.4 provides some insight into the validity of the method.

In real life, at the end of the observation period under the current ETKAS allocation mechanisms, some of the patients have been transplanted, others have been delisted, and some are still on the waiting list. We denote these three possible statuses as:

- **T**: transplanted;
- **D**: delisted;
- **W**: still on the waiting list at the end of the period.

In order to build survival functions using the Kaplan-Meier method, we compute for all patients the duration (in days) between their first day of dialysis and either the date they left the waiting list (regardless of the reason why they left), or the end of the observation period for those patients who did not exit the waiting list.



Next, for each day t, D_t and N_t are computed, where:

- D_t is the number of patients with status **D** who remained **exactly** t **days** on the waiting list. That is, the number of patients who were on the waiting list for exactly t days and then left the waiting list without a transplant.
- N_t is the number of patients who were on the waiting list t **days or more** regardless of their status.

Then, an estimator $S(t)$ of the survival function is computed in the following way (Kaplan and Meier, 1958): for each t, $S(t)$ is equal to the estimated fraction of patients who are not delisted after $t-1$ days multiplied by the fraction of patients on the waiting list at period t who are not delisted at time t. This leads to the following expression:

$$S(t) = S(t-1) \times \left(1 - \frac{D_t}{N_t}\right).$$

An example of a survival curve looks as in Figure 5.3

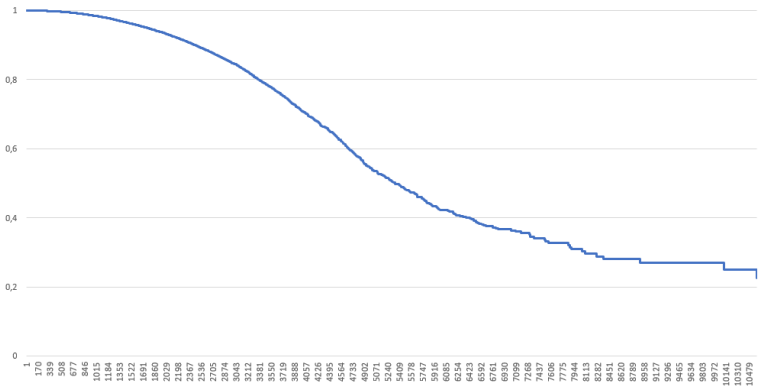


Figure 5.3: Representation of a survival curve

Using the estimated probabilities $S(t)$, delisting dates are then generated based on a draw from the resulting survival curve, under the restriction that all delisting dates must be later than the real-life transplant time.

The idea is the following: for a patient who has been transplanted after T days in the real-life data set, draw a random number u uniformly between 0 and $S(T)$, and determine t such that $u = S(t)$. Then, we will use t as the maximum time spent on the waiting list for the patient in the simulation (starting at the first day of dialysis), i.e., the delisting date of the patient will be the date of their first dialysis plus t days. If the patient does not receive a deceased donor kidney in the simulation during the t days after their first day of dialysis, he will be removed from the waiting list. Note that since the simulated delisting date results from a random draw, the delisting date of a given patient will be different in each simulation run.

Example. If a patient in the data set has been transplanted in real-life 2200 days after their first day of dialysis, a delisting date should be drawn for the simulation. Based on the survival curve, let's assume that the probability of a duration $T = 2200$ *is equal to* 0.90*, i.e.,* $S(2200) = 0.9$*. Then, a number* u *between* 0 *and* 0.90 *is drawn randomly: for example* $u = 0.40$*. Using the survival curve again, the duration* t *corresponding to* $S(t) = u = 0.4$ *can be found, here* $t = 6300$ *days. Then, the delisting date for that patient in the simulation will be the date of first dialysis plus* 6300 *days. See Figure 5.4*

We noticed some differences in the survival curves for different age groups. Hence, we decided to group the patients into four age categories: 0-15 , 16-54 , 55-64 and 65+, based on age at the date of first dialysis (see also Section 5.4.1). Moreover, as explained earlier, some patients do not have a first day of dialysis. So, 8 survival curves have been generated:

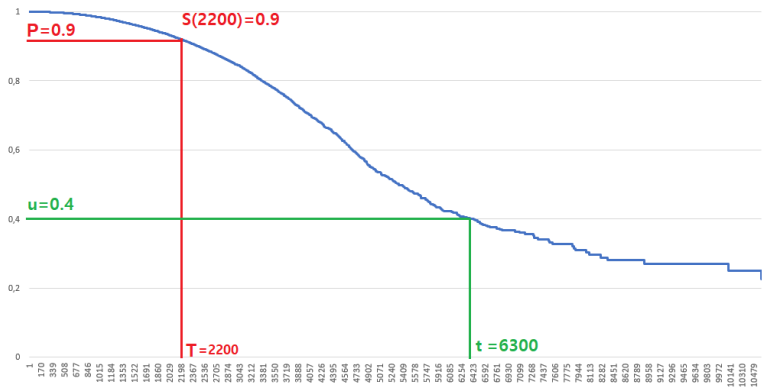


Figure 5.4: Example of simulation of a delisting date

- 4 curves for patients who have a first day of dialysis date, one for each age group,
- 4 curves for patients who do not have a first day of dialysis, one for each age group as well.

At the end of the first step of the simulation, when a delisting date must be simulated for a patient who received a real-life transplant, their simulated delisting date in the absence of any transplant during the simulation will be generated based on the curve corresponding to their age category and on whether he has been on dialysis or not.

5.4 Validation of the method

In de Ferrante et al. (2019), to assess whether the simulation *as a whole* adequately reflects the observed outcomes, the authors compared the proportion of transplantations per category of compatibility donor/patient in the real data and in the Current ETKAS simulation outputs. These comparisons led to the validation of the simulation. However, the simulation of the delisting dates for real-life transplanted patients has not been validated by itself. The goal of this section is to gain some insights into the validity of the Kaplan-Meier method in the context of the ETKAS study.

5.4.1 Independence

In Kaplan and Meier (1958), the authors state that in order to use their estimate, the population studied must satisfy an independence assumption. In our context, the independence assumption would require that the patients who are transplanted must have the same survival probability distribution as those who are delisted. This assumption seems counter-intuitive in the case of allocation mechanisms. Indeed, it would mean that the allocation

mechanism can be compared to allocating the kidneys randomly. However, the independence assumption is hard to verify in practice as pointed out, e.g., in Jager et al. (2008); Jackson et al. (2014). A lack of data prevents testing accurately the dependence or independence between the sets of patients or prevents from performing the same kind of analysis as Ruth et al. (2022). (The latter develop a Kaplan-Meier type estimate which can incorporate time-dependent characteristics of the patients and avoid the bias from having dependent censoring and lifetimes.) As a consequence, in the literature, the independence assumption is rarely tested (see, e.g., Jager et al. (2008)).

Still, some simple statistical tests can be performed to clarify the relationship between the populations of delisted and transplanted patients.

χ^2-tests

In order to have a first insight into the relation between delisted patients (**D**) and transplanted patients (**T**), two χ^2-tests of independence were performed. As a reminder, given two discrete variables X, Y and a contingency table of (X, Y) for a sample of the population, a χ^2-test of independence can be used to verify the hypothesis H_0 that X and Y are independent. In our context, for both tests, the first variable X is defined as the status of exit of the patient (delisted or transplanted). The second variable Y is different in each test.

1. **Test 1:** $Y1$:= age category.
   The age category of a patient is determined on the day of their first dialysis. If the patient has never been on dialysis, their age category is determined on their date of entry in the waiting list.
   H_0 : X and $Y1$ are independent.
   H_1 : X and $Y1$ are not independent.

Table 5.1: Contingency table of X := status of exit and $Y1$:= age category

	0-15	16-54	55-64	65+
T	1207	20763	9026	4629
D	17	2084	1441	469

The χ^2-test clearly rejects the hypothesis H_0 at a significance level of 5% as $\chi^2_{obs} = 285.44 > \chi^2_{((2-1)*(4-1);0.95)} = 7.82$, and $p < 2.2\ 10^{-16}$. In addition to the contingency Table 5.1, we also display in Figure 5.5 the proportion of delisted and transplanted patients for each age category. The proportions of **T**- and **D**-patients in each category clearly vary a lot from one category to the other. These results support the argument for separating survival curves according to age category, as mentioned at the end of Section 5.3.2.

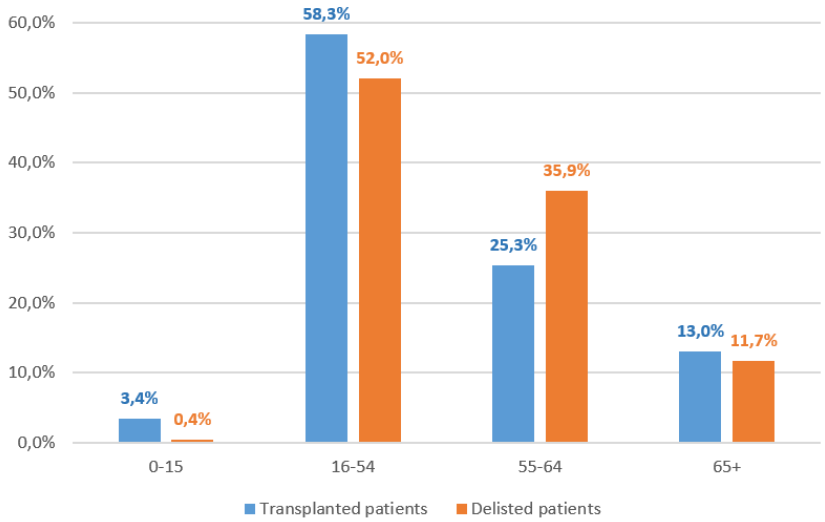


Figure 5.5: Proportion of transplanted (**T**) and delisted patients (**D**) in each age category.

2. **Test 2:** $Y2 :=$ country of origin.
   H_0 : X and $Y2$ are independent.
   H_1 : X and $Y2$ are not independent.

Table 5.2: Contingency table of $X :=$ status of exit and $Y2 :=$ country

	Austria	Belgium	Croatia	Germany	Hungary	Netherlands	Slovenia
T	3547	4584	1562	20268	666	4455	543
D	345	245	73	2830	54	442	22

The χ^2-test also rejects the hypothesis of independence at a significance level of 5% as $\chi^2_{obs} = 350.85 > \chi^2_{((2-1)*(7-1);0.95)} = 12.59$, and $p < 2.2\ 10^{-16}$. Figure 5.6 below displays the proportion of **T**- and **D**-patients in each country. Germany seems to play a big role in the rejection of independence. These results could also allow to distinguish survival curves according to countries (additionally to age categories). However, this would have led to many curves and smaller populations to estimate each of the curves. Hence, it has not been done.

Student t-tests

As the first χ^2-test rejected the hypothesis that the exit status of a patient and their age category are independent, additional tests were performed as follows. Let m_1 be the mean age at which the transplanted patients arrive on the waiting list, and let m_2 be the mean age at which the patients who are

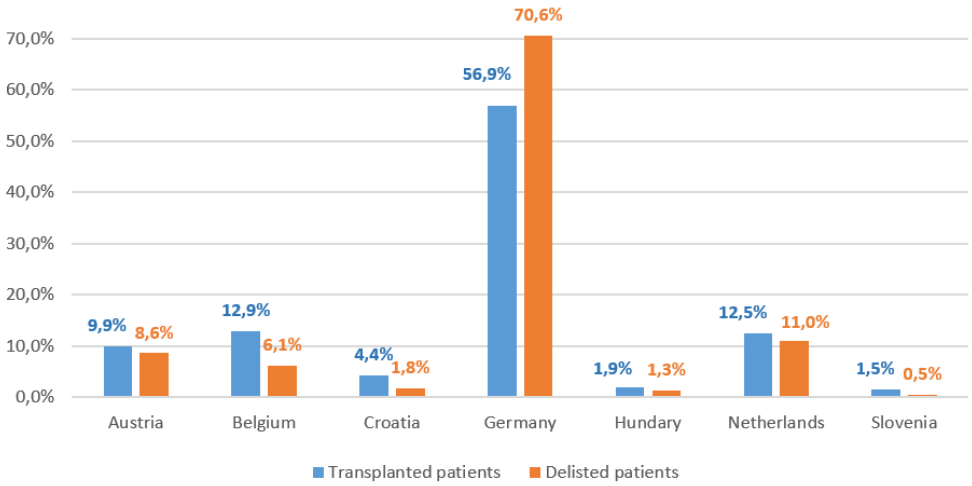


Figure 5.6: Proportion of transplanted (**T**) and delisted patients (**D**) in each country member of ET.

eventually delisted arrive on the waiting list. We used Student t-tests to test in each age category the null hypothesis $H_0 : m_1 = m_2$ vs. $H_1 : m_1 \neq m_2$.

Table 5.3: t_{obs} values in each age category

	t_{obs}	p
0-15	2.1168	0.034
16-54	-19.6013	$< 10^{-5}$
55-64	9.4673	$< 10^{-5}$
65+	0.846	0.397

As shown in Table 5.3, the t-test rejects the hypothesis H_0 at a significance level of 5% as $|t_{obs}| > |t_{0.05}| = 1.96$ and $p < 0.05$ for the age categories 0-15, 16-54, and 55-64, whereas the hypothesis H_0 cannot be rejected for the age category 65+. These results suggest that even when we separate the patients depending on their age category, the group of transplanted patients and the group of delisted patients are structurally different.

As mentioned before, it is very complicated to test rigorously the independence assumption of the Kaplan-Meier method. The simple tests that we performed, however, tend to indicate that some basic assumptions of the method are not satisfied in this study.

5.4.2 Cumulative incidence function

Another known difficulty regarding the Kaplan-Meier method, as raised in Lacny et al. (2018); Ruth et al. (2022), is that it might over- or underestimate the survival curves, especially in a medical context.

While $S^*(t)$, the survival function, expresses the probability that a patient will still be on the waiting list t days after their first day of dialysis, the **failure function** $F^*(t) = 1 - S^*(t)$ expresses the probability that a patient will be delisted (without a transplant) less than t days after their first day of dialysis.

In Lacny et al. (2018), the authors suggest that the Kaplan-Meier method overestimates the failure function of an event *when compared* with the **cumulative incidence function** (CIF), an alternative estimate which considers so-called *competing risks*. A competing risk precludes and alters the probability of the event of interest. However, the Kaplan-Meier method treats competing risks as censored observations and assumes that subjects who experience a competing risk have a similar chance of experiencing the event of interest as those lost to follow-up and, therefore, theoretically overestimates $F^*(t)$ (Lacny et al., 2018) (or equivalently, underestimate the survival curve $S^*(t)$).

In our settings, receiving a kidney transplant clearly precludes experiencing the event of interest, i.e., being delisted, and thus can be considered as a competing risk. Hence, we decided to compare the Kaplan-Meier estimate of the failure function, that is, $F_{KM}(t) = 1 - S(t)$, with the estimate of the cumulative incidence function, where the event of interest still is the delisting of the patient as in the Kaplan-Meier method, but where we consider the event of receiving a transplant as a competing risk.

We computed the following estimate of the cumulative incidence function (CIF) $F_{CIF}(t)$ (Lacny et al., 2018) (Appendix A):

$$F_{CIF}(t) = \sum_{s \leq t} \left(\frac{D_s}{N_s} \, . \, E(s-1) \right)$$

where:

- D_s is the number of subjects that experience the event of interest at time s;
- N_s is the number of patients who were on the waiting list s days or more regardless of their status.
- $E(s)$ is the "event-free" survival probability, that is the probability of experiencing neither the event of interest nor the competing risk before time s. It is estimated as:

$$E(t) = \prod_{s \leq t} \left(1 - \frac{D_s + R_s}{N_s} \right)$$

  where R_s is the number of subjects that experience the competing risk

at time s, i.e., the number of patients who remain on the waiting list for exactly s days and then leave it because they received a transplant.

Surprisingly, unlike the results in Lacny et al. (2018), given the same population, the two curves are almost identical, as illustrated in Figure 5.7.

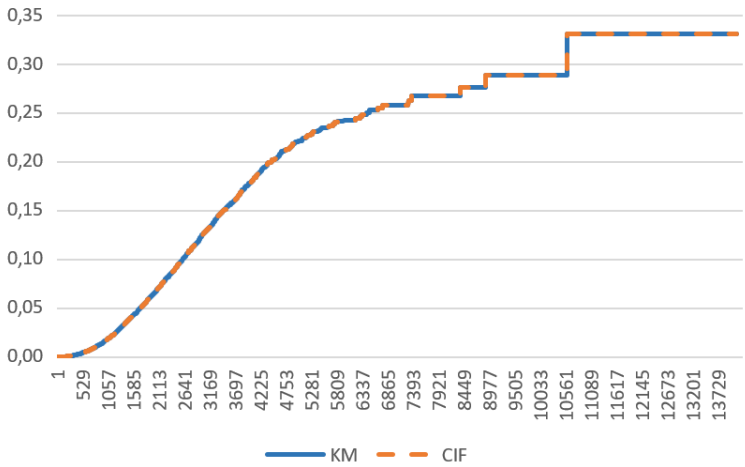


Figure 5.7: Comparison of the failure probabilities using the Kaplan-Meier and cumulative incidence estimates. The horizontal axis displays the time unit (in days) and the vertical axis displays probabilities.

This result means that even though we are clearly in a setting where at least one competing risk is present, and despite the results in Lacny et al. (2018), the Kaplan-Meier method and the CIF method seem to estimate very similar values of the failure probabilities. (Note that the probabilities are not exactly equal, they differ very slightly. However, the difference is too small to be noticeable in the figures.) Since $S(t) = 1 - F(t)$, this also means that both methods evaluate the survival probabilities almost identically; see Figure 5.8.

Although this observation does not validate the Kaplan-Meier method, it does provide reassurance that even with unverified assumptions for the Kaplan-Meier method, a different method produces nearly identical estimates of survival probabilities.

5.5 Conclusion

As mentioned in the literature, the Kaplan-Meier estimate of survival curves has advantages and drawbacks. Its principal advantage is its simplicity and easiness of computation, and the fact that it can be applied even with poor data. However, a main drawback is the assumption of independence between the time to delisting and the time to transplantation. We compared

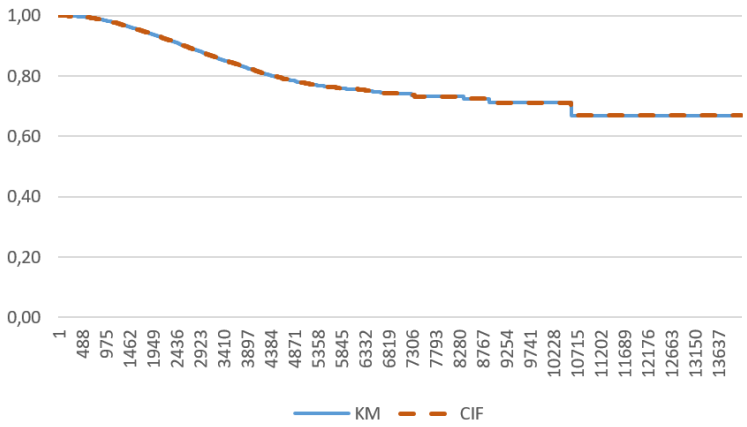


Figure 5.8: Comparison of the survival curve using the Kaplan-Meier estimate and the survival curve using the cumulative incidence estimate. The horizontal axis displays the time unit (in days) and the vertical axis displays probabilities.

the Kaplan-Meier estimate of failure probabilities with the estimate provided by the cumulative incidence function, and they output almost the exact same curves. As this survival analysis was included in a more substantial project, the exactness of the method used to generate delisting dates for transplanted patients under the current ETKAS allocation mechanism during the observation period may not have impacted the conclusions as a whole. Nevertheless, further research would be needed to better understand what method should be used to estimate survival curves in view of the available data.

Chapter 6

Conclusion

This dissertation delves into the topic of kidney transplantation and investigates three subjects related to the efficient operations of transplantation programs. Each of the core chapters includes a concluding section that is directly related to its subject matter. This final chapter recalls some of the earlier conclusions, but also provides a more general perspective on the contents covered in the dissertation.

As explained in Chapter 1, kidney transplantation can take place in two very different settings, depending on whether a kidney is transplanted from a living donor or from a deceased donor. The dissertation covers topics related to both situations, with a main emphasis on living-donor transplantation through exchanges. The opportunities offered by such kidney exchanges have been first underlined in the literature by Rapaport (1986). Roth et al. (2004) were first to investigate the underlying combinatorial optimization problems, and in particular the kidney exchange program (KEP) problem that arises when several patients are associated with willing living donors who are medically incompatible with them. During the following years, many other interesting papers have been published regarding formulations of the KEP problem, but also complexity results and proposals for optimization methods. Moreover, in recent years, the focus of the literature has shifted towards variants of the original problem. This trend is reflected in the dissertation.

Chapter 2 and Chapter 3 study cycle selections and the maximum-weighted cycle selection problem. This work was initially motivated by the connections between the cycle selection problem and the stochastic two-stage KEP (RR-2-KEP) problem of Smeulders et al. (2022), and it turned out to be fruitful in itself. Chapter 2 investigates several properties of the cycle selection problem, such as its computational complexity, the relative strength

of several IP formulations, and a polyhedral study of the cycle polytope associated with one of the formulations. Chapter 3 provides an experimental comparison of the IP formulations and applies the available knowledge about these formulations to the stochastic two-stage KEP problem. The numerical results show that while some formulations of the cycle selection problem can be used to solve it rather easily (at least, for the classes of instances that we considered), the impact of the same formulations on the solution of the RR-2-KEP problem is rather limited. The ARC formulation did not lead to an efficient solution of the stochastic RR-2-KEP problem, in spite of the theoretical study conducted in Chapter 2 and of several algorithmic enhancements of the basic Benders procedure. On the other hand, the SEA formulation emerged as an efficient one, with similar running times as those obtained by the methods tested in Smeulders et al. (2022). Future work may provide a better understanding of the polyhedral structure of the SEA formulation and may further improve the solution of the cycle selection problem and of the RR-2-KEP problem.

Chapter 4 focuses on the seemingly new concept of locally stable exchanges in compatibility digraphs with preferences. It is motivated by the assumption that the medical teams involved in a kidney exchange program may be reluctant to reject a cycle which happens to be blocked by patient-donor pairs not involved in the cycle (as may be the case when they insist that only stable exchanges should be selected). The chapter establishes a one-to-one correspondence between locally stable exchanges and local kernels of an associated digraph. Whereas local kernels have been introduced earlier by graph theorists, local stability has apparently not be studied despite the large amount of literature investigating stability in matching problems with preferences. We provide IP formulations describing locally stable exchanges and numerical results assessing the efficiency of the formulations. We also compare our numerical results regarding locally stable exchanges with those of Klimentova et al. (2023) about stable exchanges. The experiments highlight the observation that some KEP compatibility digraphs have a local stable exchange of cardinality greater than zero even though they have no stable exchange. More generally, they show that the maximum size of a locally stable exchange can be strictly greater than the maximum size of a stable exchange. These results underline the potential relevance of locally stable exchanges for kidney exchange programs. From a more theoretical perspective, the new concept of local stability raises several open questions and topics not treated in the chapter, such as the complexity of computing a maximum locally stable exchange or the structure of the digraphs having a locally stable exchange of cardinality greater than zero but no stable exchange. We also note that the concept of local stability (Chapter 4) may potentially find applications in areas unrelated to kidney exchange programs.

Chapter 5 discusses modeling questions related to the operations of the Eurotransplant deceased donor transplantation program. When a deceased donor's kidney becomes available, the transplantation program must decide, under some regulation and rules, which patient on the waiting list will be selected to receive the kidney. The ETKAS mechanism is the specific set of allocation rules followed by Eurotransplant. In Chapter 5, we question the validity of certain statistical models used to simulate a modified version of the current ETKAS mechanism. The observations formulated in this chapter may be useful for improving future modeling exercises of the same nature. Although the simulated mechanism was ultimately not adopted by Eurotransplant, the simulation model allowed us to gain valuable insights into the allocation process.

From a personal point of view, from day one, I truly loved the general topic of this thesis, that is, kidney transplantation and the complex operational questions faced by the managers of large transplantation programs. I also appreciated the various ways in which the subject could be approached and the different perspectives from which it could be studied. The problems discussed in the first chapters are initially handled from a mathematical perspective, but as the chapters unfold, they get closer to models of real-life situations: Chapter 2, on cycle selections, is purely mathematical, whereas Chapter 3 deals with numerical experiments on stochastic kidney exchange models. Chapter 4 attempts to capture some important features of real-life KEP problems, where not all donors compatible with a given patient are considered equal from the point of view of the medical team. It appears that solving the KEP problem by incorporating preferences can be a very relevant approach when running KEPs. Even though our approach is still mathematical and theoretical, its application to real-world programs seems potentially feasible and could deliver interesting insights. Finally, Chapter 5 dives even deeper into the reality of kidney transplantation programs, as it deals with a commonly used allocation mechanism for deceased donor kidneys and it involves models built on real-world historical data. Working on this project raised my awareness of how many details have to be taken into account to simulate accurately a rather simple list of rules and how real-life data should be managed to be used in a simulation process. While this chapter was not the primary focus of the thesis, it helped deepen my understanding of the operations process of a real-world transplantation program, as it relied on contacts with a medical team, and it made me appreciate how real-life situations can differ from the stylized assumptions of the scientific literature.

Bibliography

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

Aharoni, R. and Holzman, R. (1998). Fractional kernels in digraphs. *Journal of Combinatorial Theory Series B*, 73(1):1–6.

Ashlagi, I. and Roth, A. E. (2021). Kidney exchange: an operations perspective. *Management Science*, 67(9):5455–5478.

Balas, E. and Oosten, M. (2000). On the cycle polytope of a directed graph. *Networks*, 36(1):34,46.

Balas, E. and Rüdiger, S. (2009). On the cycle polytope of a directed graph and its relaxations. *Networks*, 54(1):47,55.

Bang-Jensen, J. and Gutin, G. Z. (2009). *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer, London, second edition.

Baratto, M. and Crama, Y. (2023). Cycle selections. *Discrete Applied Mathematics*, 335:4–24.

Baratto, M., Crama, Y., Pedroso, J. P., and Viana, A. (15 May 2023). Local stability in kidney exchange programs. Working paper, University of Liège.

Bauer, P. (1997). The circuit polytope: Facets. *Mathematics of Operations Research*, 22(1):110–145.

Bauer, P., Linderoth, J. T., and Savelsbergh, M. W. P. (2002). A branch and cut approach to the cardinality constrained circuit problem. *Mathematical Programming*, 91:307–348.

Ben-Ameur, W. and Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17.

Biró, P., Haase-Kromwijk, B., Andersson, T., Ásgeirsson, E. I., Baltesová, T., Boletis, I., Bolotinha, C., Bond, G., Böhmig, G., Burnapp, L., et al. (2019). Building kidney exchange programmes in Europe: an overview of exchange practice and activities. *Transplantation*, 103(7):1514.

Biró, P., Van de Klundert, J., Manlove, D., Pettersson, W., Andersson, T., Burnapp, L., Chromy, P., Delgado, P., Dworczak, P., Haase, B., et al. (2021). Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research*, 291(2):447–456.

Biró, P. and McDermid, E. (2010). Three-sided stable matchings with cyclic preferences. *Algorithmica*, 58:5–18.

Boros, E. and Gurvich, V. (2006). Perfect graphs, kernels, and cores of cooperative games. *Discrete Mathematics*, 306(19):2336–2354.

Chen, Q., Chen, X., and Zang, W. (2016). A polyhedral description of kernels. *Mathematics of Operations Research*, 41(3):969–990.

Chvátal, V. (1973). On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de recherches mathématiques, Université de Montréal.

Clark, T., Bradburn, M., Love, S., and Altman, D. (2003). Survival analysis part I: Basic concepts and first analyses. *British Journal of Cancer*, 89(2):232–238.

Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer Programming*. Springer.

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

Coullard, C. R. and Pulleyblank, W. R. (1989). On cycle cones and polyhedra. *Linear Algebra and Its Applications*, 114(C):613,640.

de Ferrante, H., Smeulders, B., Spieksma, F., Baratto, M., and Tieken, I. (2019). Simulating kidney allocation schemes for eurotransplant. Technical report. Unpublished.

Deineko, V. G. and Woeginger, G. J. (2013). Two hardness results for core stability in hedonic coalition formation games. *Discrete Applied Mathematics*, 161:1837–1842.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., and Pettersson, W. (2023a). New algorithms for hierarchical optimization in kidney exchange programs. *Operations Research*.

Delorme, M., Manlove, D., and Smeets, T. (2023b). Half-cycle: A new formulation for modelling kidney exchange problems. *Operations Research Letters*, 51(3):234–241.

Dickerson, J., Manlove, D., Plaut, B., Sandholm, T., and Trimble, J. (2016). Position-indexed formulations for kidney exchange. *arXiv.org*.

Dickerson, J. P., Procaccia, A. D., and Sandholm, T. (2018). Failure-aware kidney exchange. *Management Science*, 65(4):1768–1791.

Duchet, P. and Meyniel, H. (1993). Kernels in directed graphs: a poison game. *Discrete Mathematics*, 115(1):273–276.

Erdős, P. and Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297.

Eurotransplant (2023). *Eurotransplant Manual (version 2023.4): Chapter 4 Kidney (ETKAS and ESP)*.

Fulkerson, D. R., Nemhauser, G. L., and Trotter, L. E. (1974). *Two computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems*, pages 72–81. Mathematical Programming Studies: Approaches to Integer Programming. Springer, Berlin, Heidelberg.

Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.

Galeana-Sánchez, H. and Neumann-Lara, V. (1984). On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76.

Graham, A. J. and Pike, D. A. (2008). A note on thresholds and connectivity in random directed graphs. *Atlantic Electronic Journal of Mathematics*, 3(1):1–5.

Grötschel, M., Lovász, L., and Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 6:169–197.

Hartmann, M. and Özlük, O. (2001). Facets of the p-cycle polytope. *Discrete Applied Mathematics*, 112(1):147–178.

Held, P. J., McCormick, F., Ojo, A., and Roberts, J. P. (2016). A cost-benefit analysis of government compensation of kidney donors. *American Journal of Transplantation*, 16(3):877–885.

Hoffman, A. J. (1960). Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. In Bellman, R. and Hall, M., editors, *Combinatorial Analysis: Proceedings of Symposia in Applied Mathematics*, volume 10, pages 113–127, Providence. American Mathe-

matical Society.

Huang, C.-C. (2010). Circular stable matching and 3-way kidney transplant. *Algorithmica*, 58:137–150.

Igarashi, A. (2017). Coalition formation in structured environments. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, pages 1836–1837.

Irving, R. W. (1985). An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, 6(4):577–595.

Jackson, D., White, I. R., Seaman, S., Evans, H., Baisley, K., and Carpenter, J. (2014). Relaxing the independent censoring assumption in the Cox proportional hazards model using multiple imputation. *Statistics in Medicine*, 33(27):4681–4694.

Jager, K. J., van Dijk, P. C., Zoccali, C., and Dekker, F. W. (2008). The analysis of survival data: the Kaplan–Meier method. *Kidney International*, 74(5):560–565.

Jason T. Rich, M., J. Gail Neely, M., Randal C. Paniello, M., Courtney C. J. Voelker, M. D., Brian Nussenbaum, M., and Eric W. Wang, M. (2010). A practical guide to understanding Kaplan-Meier curves. *Otolaryngology–Head and Neck Surgery*, 143(3):331–336. PMID: 20723767.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer.

Klimentova, X., Alvelos, F., and Viana, A. (2014). A new branch-and-price approach for the kidney exchange problem. In Murgante, B., Misra, S., Rocha, A. M. A. C., Torre, C., Rocha, J. G., Falcão, M. I., Taniar, D., Apduhan, B. O., and Gervasi, O., editors, *Computational Science and Its Applications – ICCSA 2014*, pages 237–252, Cham. Springer International Publishing.

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

Klimentova, X., Pedroso, J. P., and Viana, A. (2016). Maximising expectation of the number of transplants in kidney exchange programmes. *Computers & Operations Research*, 73:1–11.

Lacny, S., Wilson, T., Clement, F., Roberts, D. J., Faris, P., Ghali, W. A., and Marshall, D. A. (2018). Kaplan–Meier survival analysis overestimates cumulative incidence of health-related events in competing risk settings: a meta-analysis. *Journal of Clinical Epidemiology*, 93:25–35.

Lam, E. and Mak-Hau, V. (2020). Branch-and-cut-and-price for the cardinality-constrained multi-cycle problem in kidney exchange. *Computers & Operations Research*, 115:104852.

Liu, Y., Tang, P., and Fang, W. (2014). Internally stable matchings and exchanges. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1433–1439.

Maffray, F. (1992). Kernels in perfect line-graphs. *Journal of Combinatorial Theory, Series B*, 55(1):1–8.

Mak-Hau, V. (2017). On the kidney exchange problem: cardinality constrained cycle and chain problems on directed graphs: a survey of integer programming approaches. *Journal of Combinatorial Optimization*, 33(1):35–59.

Mak-Hau, V. (2018). A polyhedral study of the cardinality constrained multi-cycle and multi-chain problem on directed graphs. *Computers and Operations Research*, 99:13,26.

Manlove, D. F. (2013). *Algorithmics of Matching under Preferences*. World Scientific Publishing, Singapore.

Manlove, D. F., Irving, R. W., Iwama, K., Miyazaki, S., and Morita, Y. (2002). Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279.

Mayer, G. and Persijn, G. G. (2006). Eurotransplant kidney allocation system (etkas): rationale and implementation. *Nephrology Dialysis Transplantation*, 21(1):2–3.

Mészáros-Karkus, Z. (2017). Hardness results for stable exchange problems. *Theoretical Computer Science*, 670:68–78.

Nemhauser, G. L. and Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*. John Wiley & Sons.

Ng, C. and Hirschberg, D. S. (1991). Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252.

Pansart, L., Cambazard, H., and Catusse, N. (2022). Dealing with elementary paths in the kidney exchange problem. *arXiv preprint arXiv:2201.08446*.

Pass-Lanneau, A., Igarashi, A., and Meunier, F. (2020). Perfect graphs with polynomially computable kernels. *Discrete Applied Mathematics*, 272:69–74.

Pedroso, J. P. (2014). Maximizing expectation on vertex-disjoint cycle packing. In *Computational Science and Its Applications–ICCSA 2014: 14th International Conference, Guimarães, Portugal, June 30–July 3, 2014, Proceedings, Part II 14*, pages 32–46. Springer.

Rapaport, F. T. (1986). The case for a living emotionally related international kidney donor exchange registry. In *Transplantation Proceedings*, volume 18, pages 5–9.

Ratier, G. (1996). On the stable marriage polytope. *Discrete Mathematics*, 148:141–159.

Ronn, E. (1990). NP-complete stable matching problems. *Journal of Algorithms*, 11(2):285–304.

Roth, A., Ünver, M. U., and Sönmez, T. (2004). Kidney exchange. *The Quarterly Journal of Economics*, 119(2).

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

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

Ruth, D. M., Wood, N. L., and VanDerwerken, D. N. (2022). Fully nonparametric survival analysis in the presence of time-dependent covariates and dependent censoring. *Journal of Applied Statistics*, 0(0):1–15.

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

Santos, N., Tubertini, P., A., V., and Pedroso, J. (2017). Kidney exchange simulation and optimization. *Journal of the Operational Research Society*, 68(12):1521–1532.

Seymour, P. D. (1979). Sums of circuits. In Bondy, J. A. and Murty, U. S. R., editors, *Graph Theory and Related Topics*, pages 341–355, New York. Academic Press.

Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37.

Smeulders, B., Bartier, V., Crama, Y., and Spieksma, F. C. R. (2022). Re-

course in kidney exchange programs. *INFORMS Journal on Computing*, 34:1191–1206.

Tan, J. J. M. (1990). A maximum stable matching for the roommates problem. *BIT Numerical Mathematics*, 30:631–640.

Tarjan, R. E. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160.

von Neumann, J. and Morgenstern, O. (1953). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.

Wolfe, R. A., Ashby, V. B., Milford, E. L., Ojo, A. O., Ettenger, R. E., Agodoa, L. Y., Held, P. J., and Port, F. K. (1999). Comparison of mortality in all patients on dialysis, patients on dialysis awaiting transplantation, and recipients of a first cadaveric transplant. *New England Journal of Medicine*, 341(23):1725–1730.

Wolsey, L. A. (2020). *Integer Programming*. John Wiley & Sons.

List of Figures

List of Tables