Local stability in kidney exchange programs

Marie Baratto
QuantOM, HEC Management School of the University of Liege,
Rue Louvrex 14, 4000 Liège, Belgium
e-mail: marie.baratto@uliege.be

Yves Crama
QuantOM, HEC Management School of the University of Liege,

João Pedro Pedroso
CMUP and Department of Computer Science, Faculty of Sciences, University of Porto,

Ana Viana
INESC TEC and ISEP – School of Engineering, Polytechnic of Porto,

Abstract: When each patient of a kidney exchange program has a preference ranking over its set of compatible donors, questions naturally arise surrounding the stability of the proposed exchanges. We extend recent work on stable exchanges by introducing and underlining the relevance of a new concept of locally stable, or L-stable, exchanges. We show that locally stable exchanges in a compatibility digraph are exactly the so-called local kernels (L-kernels) of an associated blocking digraph (whereas the stable exchanges are the kernels of the blocking digraph), and we prove that finding a nonempty L-kernel in an arbitrary digraph is NP-complete. Based on these insights, we propose several integer programming formulations for computing an L-stable exchange of maximum size. We conduct numerical experiments to assess the quality of our formulations and to compare the size of maximum L-stable exchanges with the size of maximum stable exchanges. It turns out that nonempty L-stable exchanges frequently exist in digraphs which do not have any stable exchange. All the above results and observations carry over when the concept of (locally) stable exchanges is extended to the concept of (locally) strongly stable exchanges.

Keywords: OR in health services, kidney exchange, stable exchange, local kernel, integer programming

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 a cyclic exchange c is proposed by the program, but another cycle c' exists such that all patients of c' are better off in c' than in c, then the exchange c may be considered as *unstable*, in the sense that it may be difficult to convince the patients (and the doctors) to implement c rather than c'. (A more precise definition will be given in Section 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 2. In Section 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 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 5, we propose integer programming formulations for local stable exchanges. Section 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 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.

2 Basic concepts and literature review

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], [Manlove, 2013].

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 $K \in \{2, 3, 4\}$.

There is a large amount of literature documenting formulations and algorithms for kidney exchange optimization problem; see, e.g., [Roth et al., 2007], [Constantino et al., 2013], [Dickerson et al., 2016], [Biró et al., 2021] [Delorme et al., 2023b], [Delorme et al., 2023a]. When $K = 2$ or when $K = n$, maximizing the number of transplants reduces to a bipartite 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 donors 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 altruistic donor: in this way, chains are transformed into cycles in the augmented digraph. For the remainder of this document, chains will not be explicitly mentioned as they can be handled in the same way as cycles.

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. When no ties are allowed in the preference lists, then the existence of stable solutions can be checked in polynomial time [Irving, 1985, Manlove, 2013]. On the other hand, the 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] 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 paper 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.

3 Stability and local stability

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 $i \in V$ has expressed *preferences* over its set of compatible donors. The preferences can be described by a *rank* function $r_i : N_G^-(i) \to \mathbb{R}$, with the interpretation that, for all $j, k \in N_G^-(i)$, patient 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 1 hereunder, the patient of pair 2 prefers the donor of pair 4 to the donor of pair 1.

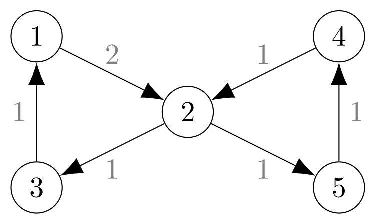

Figure 1: A small digraph with preferences

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). 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 1. 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 2. 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 3. 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 [Roth et al., 2004], [Biró and McDermid, 2010], [Klimentova et al., 2023]).

Definition 4. An exchange $\mathcal{M}$ is *stable* if there is no blocking cycle for $\mathcal{M}$ in $\mathcal{C}_K(G)$.

Example 1. In Figure 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 4 mimicks the definition of stable matchings given in Section 2.

3.2 Local stability: definitions

Example 2. Consider the digraph G in Figure 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 opposition from anyone.

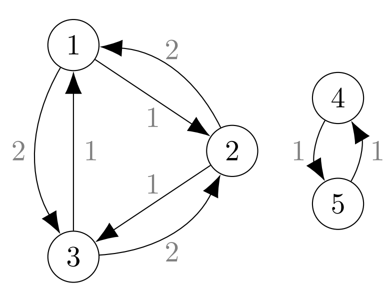

Figure 2: A digraph without stable exchange

The anomaly underlined in Example 2 arises because Definition 3 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 5. 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 6. 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 3 and Definition 5 makes it clear that every L-blocking cycle is also blocking, but not conversely. As a consequence, every stable exchange is locally stable, but not conversely.

Example 3. In Figure 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.

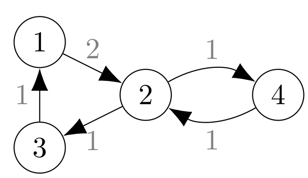

Figure 3: $u := (2, 4, 2)$ is L-blocking for $v := (1, 2, 3, 1)$

Example 4. In Example 2 and Figure 2, the exchange $\mathcal{M} = \{u_4\}$ is locally stable, but not stable.

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 5 to derive integer programming formulations.

Definition 7. For a cycle $v \in \mathcal{C}_K(G)$, we denote by $\mathcal{B}(v)$ the set of all L-blocking cycles of v.

Definition 8. Two intersecting cycles u, v are *friends* 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), or if the two cycles share at least two vertices and one prefers u while the other one prefers v.

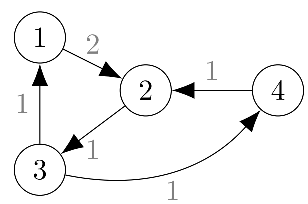

Figure 4: $v = (1, 2, 3, 1)$ and $u = (2, 3, 4, 2)$ are friends when $K = 3$

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. $\square$

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 5.

(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 $w \in \mathcal{B}(v) \cup \mathcal{F}(v)$. $\square$

We are now ready for the characterization theorems.

Theorem 1. *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 4 and from the equivalence of (i)-(ii) in Lemma 2. □

Theorem 2. *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) ⇒ (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) ⇒ (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 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 1. 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 Morgenste and [Boros and Gurvich, 2006] for related literature.

Definition 9. 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 5).

From Theorem 1 and the definition of kernels, we immediately obtain:

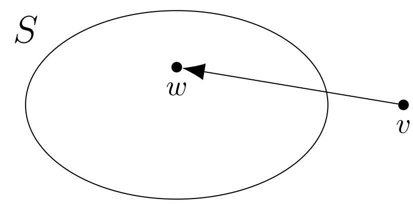

Figure 5: An absorbing set S

Theorem 3. *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 3 does not come as a complete surprise, as similar observations have been formulated in the literature, e.g., for the stable marriage problem; see [Manlove, 2013], [Ratier, 1996]. We are not aware that the connection has been explicitly stated for stable kidney exchanges.

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 10. 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 6 provides an illustration. Clearly, every kernel is a local kernel.

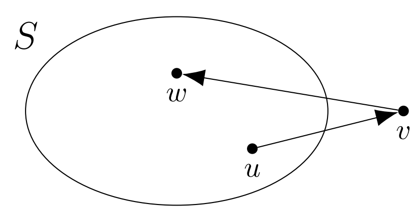

Figure 6: A locally absorbing set S

The relation between L-stable exchanges and L-kernels is akin to the relation between stable exchanges and kernels, namely:

Theorem 4. *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 2 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 2, 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 1, 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 2 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. $\square$

There only seems to be a handful of publications about local kernels. We collect here some simple observations of interest.

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.

Fact 3. Every kernel is a maximal kernel and is a maximal L-kernel. However, in view of Fact 1, a maximal L-kernel is not necessarily a kernel.

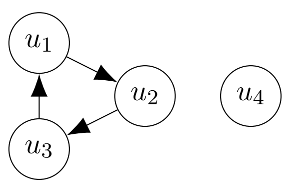

Figure 7: A blocking digraph without kernel but with a nonempty L-kernel

Example 5. Figure 7 illustrates the previous facts. It displays the blocking digraph G^* of the digraph G in Figure 2. In line with the discussion in Section 3.2, G^* has no kernel (essentially, because of the cyclic component (u_1, u_2, u_3, u_1)), but $S = \{u_4\}$ is a maximal L-kernel of G^*.

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 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.

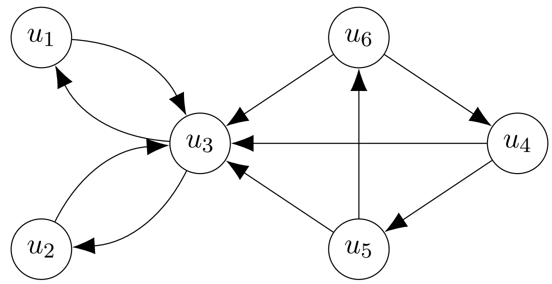

Figure 8: A digraph G^* with an L-kernel larger than the unique kernel

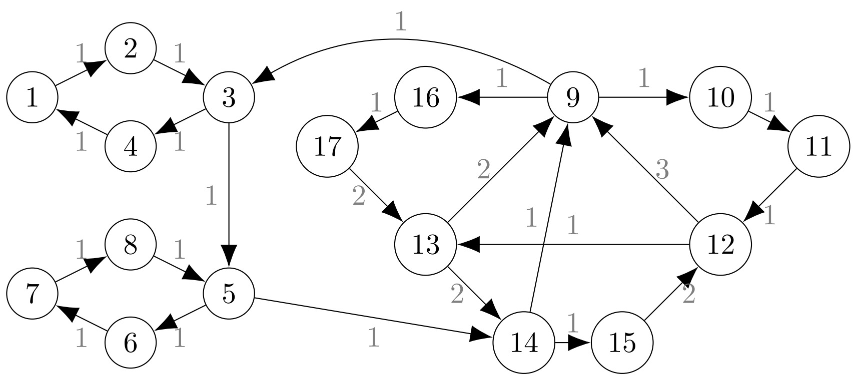

Figure 9: A compatibility digraph G

It is interesting to observe that G^* in Figure 8 is the blocking digraph of the KEP compatibility graph in Figure 9 (for $K = 4$), where $u_1 = (1, 2, 3, 4, 1)$, $u_2 = (5, 6, 7, 8, 5)$, $u_3 = (3, 5, 14, 9, 3)$, $u_3 = (3, 5, 14, 9, 3)$, $u_4 = (12, 13, 14, 15, 12)$, $u_5 = (9, 10, 11, 12, 9)$, and $u_6 = (9, 16, 17, 13, 9)$. As a consequence, the maximum stable exchange in G is $\mathcal{M}^s = \{u_3\}$ where $u_3 = (3, 5, 14, 9, 3)$, whereas the maximum L-stable exchange is $\mathcal{M}^{ls} = \{u_1, u_2\}$ where $u_1 := (1, 2, 3, 4, 1)$ and $u_2 := (5, 6, 7, 8, 5)$.

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 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 2 for stable exchanges strengthen this statement.) The complexity of computing L-kernels has apparently not been investigated in the literature, but we can establish:

Theorem 5. *Given a digraph $G = (V, A)$, deciding whether G has a nonempty local kernel is NP-complete.*

The proof is in A. Note that, as a corollary of Theorem 5, computing a local kernel of maximum size is also NP-hard.

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:

$$y_u + y_v \le 1 \qquad \forall u, v \in \mathcal{C}_K(G) : V(u) \cap V(v) \neq \emptyset \tag{1}$$

$$1 \le y_v + \sum_{w \in \mathcal{B}(v) \cup \mathcal{F}(v)} y_w \qquad \forall v \in \mathcal{C}_K(G) \tag{2}$$

$$y_v \in \{0, 1\} \qquad \forall v \in \mathcal{C}_K(G). \tag{3}$$

Theorem 6. *The solutions of* (1), (2), (3) *describe all stable exchanges of* G.

Proof. Suppose that y satisfies (1), (2), (3), and let $\mathcal{M}$ be the associated set of cycles. Constraints (1) express that $\mathcal{M}$ is an exchange, and constraints (2) express condition (b) in Theorem 1. □

Formulation (1)-(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 (1) can be replaced by the stronger constraints

$$\sum_{v \in \mathcal{C}_K(G): i \in V(v)} y_v \le 1 \qquad \forall i \in V \tag{4}$$

since (4) expresses that at most one cycle containing a given vertex i can be included in an exchange. The collection of cycles $\{v \in \mathcal{C}_K(G) : i \in V(v)\}$ is a clique in G^* and hence, (4) is one of the well-known clique inequalities

$$\sum_{v \in C} y_v \le 1 \qquad \text{if } C \text{ is a maximal clique in } G^*.$$

Observe, however, that (4) does not necessarily include all maximal clique inequalities for G^*. The strengthened formulation (2)-(4) is a restatement of the so-called "cycle formulation" of stable exchanges in [Klimentova et al., 2023].

Let us turn next to a formulation of L-stability. Define the constraints:

$$y_u + y_v \le 1 \qquad \forall u, v \in \mathcal{C}_K(G) : V(u) \cap V(v) \neq \emptyset \tag{5}$$

$$y_u \le \sum_{w \in \mathcal{B}(v) \cup \mathcal{F}(v)} y_w \qquad \forall u \in \mathcal{C}_K(G), \forall v \in \mathcal{B}(u) \cup \mathcal{F}(u) \tag{6}$$

$$y_v \in \{0, 1\} \qquad \forall v \in \mathcal{C}_K(G). \tag{7}$$

Theorem 7. *The solutions of* (5), (6), (7) *describe all L-stable exchanges of* G.

Proof. When y satisfies (5), (6), (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 (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, (5)-(7) exactly describes the local kernels of G^*. □

Here again, the constraints (5) can be replaced by the tighter clique inequalities (4). Moreover, note that the inequalities (6) are redundant when $v \in \mathcal{F}(u)$: indeed, if $v \in \mathcal{F}(u)$ then $u \in \mathcal{F}(v)$, hence the right-hand side of (6) contains y_u. As it stands now, the formulation (5)-(7) can be rewritten as the following natural formulation of local kernels in $G^* = (V^*, A^*)$:

$$y_u + y_v \leq 1 \qquad \forall (u, v) \in A^* \tag{8}$$

$$y_u \leq \sum_{w:(v,w)\in A^*} y_w \qquad \forall (u, v) \in A^* \tag{9}$$

$$y_v \in \{0, 1\} \qquad \forall v \in V^*. \tag{10}$$

The stability constraints (9) can be aggregated by fixing v and summing for all u such that $(u, v) \in A^*$. This leads to

$$\sum_{u:(u,v)\in A^*} y_u \leq \delta^-(v) \sum_{w:(v,w)\in A^*} y_w \qquad \forall v \in V^* \tag{11}$$

where $\delta^-(v) = |\{u : (u, v) \in A^*\}|$. One can easily verify that constraints (4), (10) and (11) correctly describe the L-kernels of G^* (and hence, the L-stable exchanges of G). The linear relaxation of (11) is weaker than that of (9). However, there are only $|V^*|$ aggregated constraints of type (11), while there are $\mathcal{O}(|A^*|)$ constraints of type (9). We will experimentally compare these different formulations in Section 6.

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 proposed in Section 5 and second, to compare optimal stable against optimal L-stable exchanges. All formulations were implemented using Python 3.10 programming language and 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.

6.1 Instances

We have performed numerical tests on a set of randomly generated instances which are meant to reproduce the features of compatibility digraphs arising in real-world KEPs. The instances are described in more 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 $\lceil 0.05 \times n \rceil$ non-directed donors. The chains originating from an NDD are viewed as cycles in an augmented digraph, as explained in Section 2.2. Fifty different digraphs with $|V| = n + \lceil 0.05n \rceil$ vertices are available for each value of n. For each digraph, the preferences on the arcs are strict, that is, a patient is never indifferent between two distinct donors. In total, we have 3300 instances (22 $\times$ 3 $\times$ 50) in this dataset.

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.

By way of illustration, Table 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 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 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

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

$$\max \sum_{u \in \mathcal{C}_K(G)} |V(u)| \, y_u \tag{12}$$

where $|V(u)|$ is the length of cycle u. Four different IP formulations have been tested: beside the integrality constraints (7), they contain the following L-stability constraints.

- Formulation 1: constraints (5) and (6); total: $2|A^*|$ constraints.

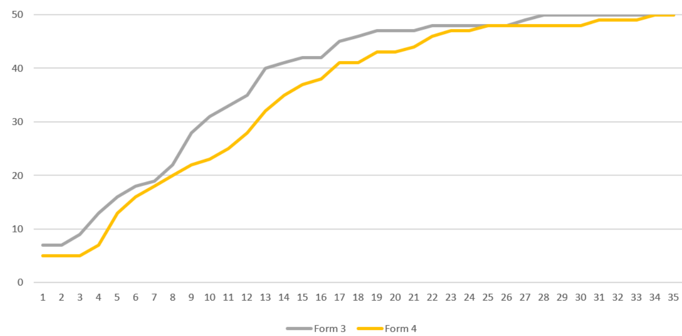

Figure 10: Comparison of Gap_{LP} for Formulations 3 and 4 when $n = 40$, $K = 3$. The horizontal axis displays gaps and the vertical axis displays the number of instances with a gap smaller than a given value.

- Formulation 2: constraints (5) and (11); total: $|A^*| + |V^*|$ constraints.
- Formulation 3: constraints (4) and (6); total: $|V| + |A^*|$ constraints.
- Formulation 4: constraints (4) and (11); total: $|V| + |V^*|$ constraints.

Recall from Section 5 that constraints (4) are stronger than (5), and constraints (6) are stronger than (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 1 and Table 4).

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 10 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) considerably tighten the formulations, whereas the aggregation of constraints (6) into (11) does not deteriorate very much the upper bounds.

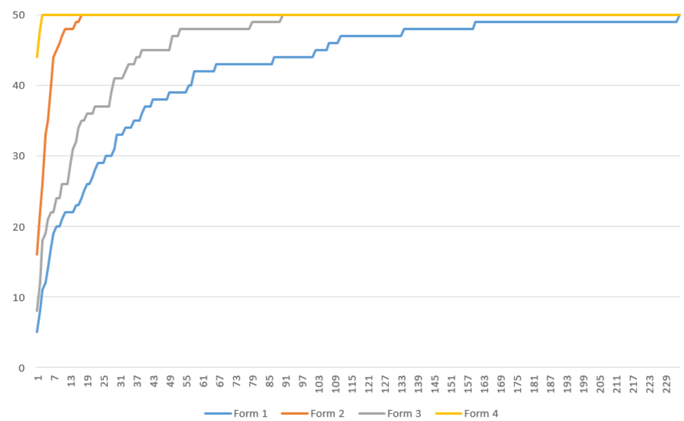

Figure 11: Comparison of running time for Formulations 1–4 when $n = 40$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

Table 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 2 displays the mean running time (in seconds) for each formulation, computed over 50 instances of size $n = 40$ and $K = 3$. Figure 11 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 3 displays the number of instances that were solved within the time limit among 50 instances with $K = 3$ and $n \in \{40, 60, 80, 100\}$.

Table 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. Therefore, we restrict our attention to this formulation in the sequel. When $K = 2$, we will see in the next section that large instances of the L-exchange problem can be solved efficiently. 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.

6.3 Comparison with stable exchanges

As underlined in Section 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]), 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 (2)-(4) and formulation (4), (11), (7) (Formulation 4), respectively, with the same objective function (12).

Let us first briefly comment on the running time of the IP solver for each problem. Figure 12 and Figure 13 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.

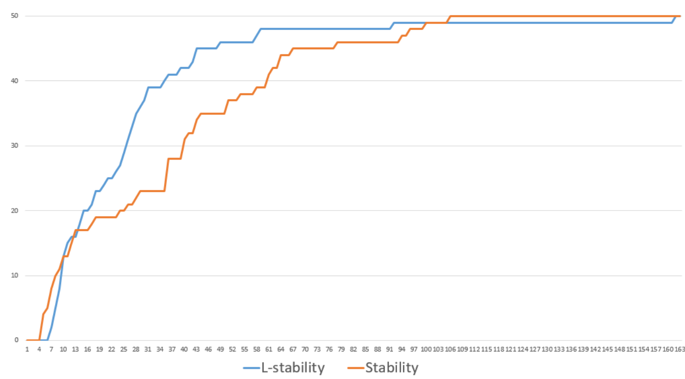

Figure 12: Running time for stable exchanges and L-stable exchanges, $n = 80$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

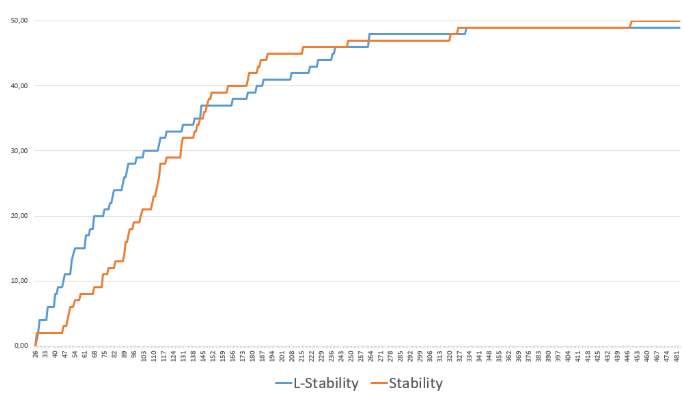

Figure 13: Running time for stable exchanges and L-stable exchanges, $n = 100$, $K = 3$. The horizontal axis displays running times (in seconds) and the vertical axis displays the number of instances with a running time smaller than a given value.

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. For example, among 600 instances with n ranging between 50 and 400, 72 instances do not have a stable exchange. Table 4 provides information about these instances, with the same notations as in Table 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: 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 5 synthesizes some results of our computational experiments. The left part of the table 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 5: 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 5. 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 5 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 43.8 for the remaining 5 instances. Similarly, when $n = 400$, the 32 instances with a stable exchange have an average optimal value equal to 230.4 for both problems, whereas 17 instances with no stable solution have an average maximum L-stable exchange of size 59.41.

7 Local strong stability

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 3.1.

Definition 11. 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 12. 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 13. 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 3, we now propose a seemingly new concept of *locally weakly blocking cycles* and *locally strongly stable exchanges*.

Definition 14. 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 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 15. 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 8. *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 8. 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 16. Two intersecting cycles u and v are *strong friends* 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 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.

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 1 and Theorem 2 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 9. *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 10. *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, 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 11. *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 9 and Theorem 10 (compare with Theorem 3 and Theorem 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:

$$\max \sum_{v \in V^*} |V(v)|\, y_v \tag{13}$$

$$\sum_{v \in V^* : i \in V(v)} y_v \leq 1 \qquad \forall i \in V \tag{14}$$

$$\sum_{u:(u,v) \in A^{**}} y_u \leq \Delta^-(v) \sum_{w:(v,w) \in A^{**}} y_w \qquad \forall v \in V^* \tag{15}$$

$$y_v \in \{0,1\} \qquad \forall v \in V^*. \tag{16}$$

where $\Delta^-(v) := |\{u : (u,v) \in A^{**}\}|$ is the number of in-neighbors of v in the weak blocking digraph G^{**}.

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 6 provides information about the associated graphs. The notations are the same as in Table 4, 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 6: Size parameters of instances of with $K = 3$

n	$\|V\|$	$\|A\|$	$\|V^*\|$	**min**$\|V^*\|$	**max**$\|V^*\|$	$\|A^{**}\|$	**min**$\|A^{**}\|$	**max**$\|A^{**}\|$
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 6 and Table 4, 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 7 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 7: 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 ($\phi_{LS} = 1$). When a strongly stable exchange exists, its average cardinality (for $n = 80$) 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 $a_{LS} = 36.1$, 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.

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

$$\begin{aligned}
&\max \sum_{v \in W} y_v \\
&y_u + y_v \leq 1 && \forall (u, v) \in E \\
&y_u \leq \sum_{w:(v,w) \in E} y_w && \forall (u, v) \in E \\
&y_v \in \{0, 1\} && \forall v \in W.
\end{aligned}$$

Fifty digraphs $D = (W, E)$ have been randomly generated for different values of $|W| = n$ and different densities d, where each arc (i, j), $i \neq j$, is present in E with probability d independently of the other arcs. Each line of Table 8 gives the following information for 50 instances with parameters 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 8: 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 6.3 have a rather small density, whereas the weak blocking digraphs G^{**} in Section 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 8 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 $n = 350$.

9 Conclusions and perspectives

In this paper, 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.

References

[Aharoni and Holzman, 1998] Aharoni, R. and Holzman, R. (1998). Fractional kernels in digraphs. *Journal of Combinatorial Theory Series B*, 73(1):1–6.

[Biró et al., 2021] Biró, P., van de Klundert, J., Manlove, D., Pettersson, W., Andersson, T., Burnapp, L., Chromy, P., Delgado, P., Dworczak, P., Haase, B., Hemke, A., Johnson, R., Klimentova, X., Kuypers, D., Nanni Costa, A., Smeulders, B., Spieksma, F., Valentín, M. O., and Viana, A. (2021). Modelling and optimisation in European kidney exchange programmes. *European Journal of Operational Research*, 291(2):447–456.

[Biró and McDermid, 2010] Biró, P. and McDermid, E. (2010). Three-sided stable matchings with cyclic preferences. *Algorithmica*, 58:5–18.

[Boros and Gurvich, 2006] Boros, E. and Gurvich, V. (2006). Perfect graphs, kernels, and cores of cooperative games. *Discrete Mathematics*, 306(19):2336–2354.

[Chen et al., 2016] Chen, Q., Chen, X., and Zang, W. (2016). A polyhedral description of kernels. *Mathematics of Operations Research*, 41(3):969–990.

[Chvátal, 1973] Chvátal, V. (1973). On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de recherches mathématiques, Université de Montréal.

[Constantino et al., 2013] 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.

[Delorme et al., 2023a] 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 et al., 2023b] 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 et al., 2016] Dickerson, J., Manlove, D., Plaut, B., Sandholm, T., and Trimble, J. (2016). Position-indexed formulations for kidney exchange. *arXiv.org*.

[Duchet and Meyniel, 1993] Duchet, P. and Meyniel, H. (1993). Kernels in directed graphs: a poison game. *Discrete Mathematics*, 115(1):273–276.

[Gale and Shapley, 1962] Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.

[Galeana-Sánchez and Neumann-Lara, 1984] Galeana-Sánchez, H. and Neumann-Lara, V. (1984). On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76.

[Irving, 1985] Irving, R. W. (1985). An efficient algorithm for the “stable roommates” problem. *Journal of Algorithms*, 6(4):577–595.

[Klimentova et al., 2023] 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.

[Maffray, 1992] Maffray, F. (1992). Kernels in perfect line-graphs. *Journal of Combinatorial Theory, Series B*, 55(1):1–8.

[Manlove, 2013] Manlove, D. F. (2013). *Algorithmics of matching under preferences*. World Scientific Publishing, Singapore.

[Manlove et al., 2002] 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.

[Mészáros-Karkus, 2017] Mészáros-Karkus, Z. (2017). Hardness results for stable exchange problems. *Theoretical Computer Science*, 670:68–78.

[Ng and Hirschberg, 1991] Ng, C. and Hirschberg, D. S. (1991). Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4(2):245–252.

[Ratier, 1996] Ratier, G. (1996). On the stable marriage polytope. *Discrete Mathematics*, 148:141–159.

[Ronn, 1990] Ronn, E. (1990). NP-complete stable matching problems. *Journal of Algorithms*, 11(2):285–304.

[Roth et al., 2004] Roth, A., Sönmez, T., and Ünver, M. U. (2004). Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488.

[Roth et al., 2007] 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.

[Shapley and Scarf, 1974] Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37.

[Smeulders et al., 2022] Smeulders, B., Bartier, V., Crama, Y., and Spieksma, F. C. R. (2022). Recourse in kidney exchange programs. *INFORMS Journal on Computing*, 34:1191–1206.

[von Neumann and Morgenstern, 1953] von Neumann, J. and Morgenstern, O. (1953). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.

A Proof of Theorem 5

Theorem 5. *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 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 14 for a clause C_k containing a literal x_i.

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^*.

Conversely, assume that S^* is a nonempty local kernel of D. Let us see what vertices can be in S^*.

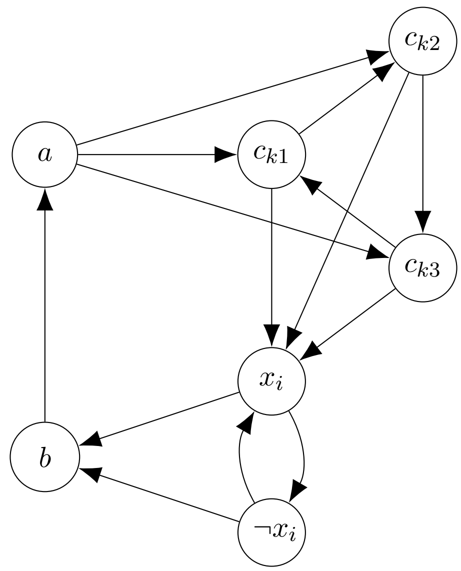

Figure 14: Construction for a clause C_k containing a literal x_i

- 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 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.) $\square$