Locally stable matchings for the roommates problem

Elise Vandomme[a,*], Yves Crama[a], Marie Baratto[b]

[a] *QuantOM, HEC Liège - Management School of the University of Liège, Rue Louvrex 14, 4000 Liège, Belgium.*

[b] *Technology and Operations management, Rotterdam School of Management, Erasmus University, Burgemeester Oudlaan 50, 3062 PA, Rotterdam, The Netherlands.*

Abstract

Stability of matchings under preferences has been extensively studied in operations research and in economics. In the classical stable roommates problem, the aim is to match pairs of individuals who intend to share living spaces or to trade indivisible resources, as in kidney exchange programs. A matching is called stable when it has no blocking pair, meaning a pair of individuals who would prefer to be matched together rather than to their current mates, or rather than remaining unmatched. Due to its relevance in real-world settings, various extensions of this problem have been investigated in the literature. These extensions have been analyzed from an algorithmic perspective, and polynomial algorithms or hardness results are available for many of them. The present paper examines locally stable matchings, a recently introduced generalization of stable matchings. A matching is locally stable if no blocking pair intersects it. Every stable matching is locally stable, but the converse is not true: a stable matching may not exist for some preference structures, whereas a locally stable matching always exists (though it may be empty). We present several structural results on locally stable matchings, including a strong property of such matchings with respect to stable partitions. From these results, we derive a polynomial-time algorithm to compute a locally stable matching of maximum size.

Keywords: Combinatorial optimization, Stable roommates problem, Stable matching, Complexity

---

1. Introduction

The concept of stability has been extensively studied in the literature on matchings under preferences. This stream of literature has its origins in a classical paper by Gale and Shapley (1962) on the *stable marriage problem*. Numerous extensions have subsequently

*Corresponding author

been investigated in the operations research and economic literature, including in particular non-bipartite matching problems with preferences such as the *stable roommates problem*, as well as many variants that consider complete or incomplete preference lists, with or without ties. Different extensions of these models have also been considered in the context of *kidney exchange programs*, where *stable exchanges* have been described in the pioneering work by Roth et al. (2004) and in many follow-up publications. The associated decision or optimization models have been extensively studied from an algorithmic perspective, and polynomial algorithms, hardness results and computational experiments are available for many of them; see, e.g., among a host of contributions, Gale and Shapley (1962), Irving (1985), Gusfield (1988), Ronn (1990), Tan (1990, 1991a,b), Teo and Sethuraman (1998, 2000), Manlove et al. (2002), Biró (2007), Fleiner et al. (2007), Biró and McDermid (2010), Huang (2010), Fleiner et al. (2011), Liu et al. (2014), Mészáros-Karkus (2017), Delorme et al. (2019), Pass-Lanneau et al. (2020), Agoston et al. (2022), Klimentova et al. (2023), Glitzner and Manlove (2024), as well as the monographs Gusfield and Irving (1989) and Manlove (2013). Some of these results will be discussed in more detail in subsequent sections.

Our objective in this paper is to improve our understanding of a recently introduced concept of *locally stable* (or *L-stable*) *matchings* in the framework of the roommates problem (Baratto et al., 2025). Locally stable matchings provide an interesting generalization of stable matchings, in the sense that their definition is intuitively appealing, that every stable matching is L-stable, and that every roommates instance admits a (possibly empty) L-stable matching. A question left open in Baratto et al. (2025) was to determine the complexity of computing an L-stable matching of maximum size. We will show in this paper that the problem can be solved in polynomial time.

Section 2 states the main definitions that are needed in the paper. Section 3 establishes several useful results regarding the structure of L-stable matchings, in particular of (inclusion-wise) maximal ones. From these results, a polynomial-time algorithm for identifying a maximum L-stable matching is derived in Section 4. Section 5 draws some conclusions and outlines further lines of research.

2. Models and definitions

In this paper, we consider different models of *roommates problems*. In all cases, we assume that each agent in a finite set V has expressed preferences over a subset of the other agents in V. As in Gusfield and Irving (1989) or Manlove (2013), we model this situation by a *preference table* which constitutes an *instance* of the problem: in the table, each line defines the *preference list* of an agent, that is, an ordered list of agents ordered by decreasing preferences. We assume that the preferences are *strict*, meaning that an agent cannot be indifferent between two agents that appear in its preference list. But the preferences may be *incomplete*: the preference list of agent i typically contains a strict subset of $V \setminus \{i\}$, representing the agents that are *admissible* for i.

In roommates problems, we are interested in forming pairs of mutually admissible partners. Hence, we assume without loss of generality that an agent j appears in the list of i if and only if i appears in the list of j.

With a preference table, we can then associate an *undirected graph* $G = (V, E)$ where the *edge* $\{i, j\}$ is in E if and only if agent j appears in the preference list of agent i. Henceforth, we call the agents *vertices*. We often denote an edge $\{i, j\}$ as ij, for brevity. Our graph terminology follows Papadimitriou and Steiglitz (1982), Bang-Jensen and Gutin (2009). We will introduce more terminology as needed.

Strictly speaking, an instance of a roommates problem is defined by a preference table. In this paper, however, we will often focus on the underlying graph $G = (V, E)$ and on pairs of admissible partners forming a matching in G.

Definition 2.1. For a graph $G = (V, E)$, a *matching* of G is a set M of pairwise disjoint edges: $M \subseteq E$ and for all $e_1, e_2 \in M$, $e_1 \cap e_2 = \emptyset$. We denote by $V(M)$ the set of vertices contained in the edges of the matching, and we say that a vertex $x \in V$ is *matched in* M, or is *covered by* M, if $x \in V(M)$. When $xy \in M$, we say that x and y are *mates*, and we write $y = M(x)$, $x = M(y)$. A matching M of G is called *perfect* if every vertex of G is matched in M, that is, if $V(M) = V$.

Definition 2.2. Consider a preference table and let M be a matching of the associated graph $G = (V, E)$. An edge $xy \in E \setminus M$ is said to be *blocking* for M (or to *block* M) if

(i) either x prefers y to its mate $M(x)$ or x is not matched in M, and

(ii) either y prefers x to its mate $M(y)$ or y is not matched in M.

The matching M is *stable* if there is no blocking edge for M in E. As illustrated in the following example, a graph G may not always contain a stable matching.

Example 2.3. Consider the preference table and its associated graph depicted in Figure 1. It is easy to check that there is no stable matching in this case. □

Definition 2.4. Given a preference table, the *stable roommates problem* (SRP) is to find a stable matching in the associated graph $G = (V, E)$ or to prove that there is none.

Irving (1985) has provided the first polynomial-time algorithm for the stable roommates problem. He considered the case where the preferences are complete, in the sense

a	b	c
b	c	a
c	a	b
d	e	
e	d	

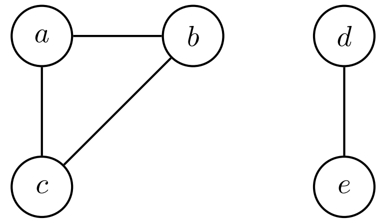

Figure 1: A preference table and its associated graph

that G is a complete graph, and he was mainly interested in the existence of a perfect stable matching. The generalization to incomplete preferences and imperfect matchings is handled in Gusfield and Irving (1989). See also Manlove (2013) for (much) more information on SRP.

Baratto et al. (2025), however, argued that the classical stability requirement expressed by Definition 2.2 may be too strong in some cases. In Example 2.3, for instance, it may be understandable that the edge ab blocks the matching $M_1 = \{ac\}$, since selecting M_1 prevents a from being paired with b, even though b is free and a prefers it to c. But it is more difficult to justify why ab also blocks the matching $M_2 = \{de\}$, since selecting the pair de does not interfere in any way with the choices available to a and b (sheer jealousy on the part of a and b, perhaps?).

Motivated by this observation, Baratto et al. (2025) have proposed to modify Definition 2.2 and Definition 2.4 as follows. Their definitions actually apply to the broader case of cyclic exchanges, which is typical of the kidney exchange problem. We only deal here with the special setting of the roommates problem.

Definition 2.5. Consider a preference table and let M be a matching of the associated graph $G = (V, E)$. An edge $xy \in E \setminus M$ is said to be *locally blocking* for M (or to be *L-blocking* for M, or to *locally block* M, or to *L-block* M) if

(i) xy is blocking for M, and

(ii) x or y is matched in M: $x \in V(M)$ or $y \in V(M)$

The matching M is *locally stable* (or *L-stable*) if there is no L-blocking edge for M in E.

Condition (ii) in Definition 2.5 explicitly requires that an L-blocking edge should intersect the matching (that is, interact with it). This rules out ab as an L-blocking edge for the matching $M_2 = \{de\}$ in Example 2.3.

Since every L-blocking edge is blocking, it follows that every stable matching is L-stable. But the converse implication does not hold in general as illustrated by Example 2.3 where the matching $M_2 = \{de\}$ is L-stable, but not stable. In fact, Definition 2.5 implies that the empty matching $M = \emptyset$ is always L-stable (because there is no edge xy satisfying condition (ii) when M is empty). Numerical experiments performed by Baratto et al. (2025) indicate that when $|V|$ is large enough (e.g., $|V| = 400$), a significant proportion of random instances do not have a stable matching, but have a nonempty L-stable matching.

Since the natural extension of Definition 2.4 to L-stable matchings is meaningless, a more relevant problem is defined as follows.

Definition 2.6. Given a preference table, the *locally stable roommates problem* (L-SRP) is to find an L-stable matching of maximum size in the associated graph $G = (V, E)$.

The locally stable matching problem is the main object of attention in this paper. Before we turn to it in more detail, and for the sake of completeness, let us recall a related definition from Tan (1990, 1991b), Liu et al. (2014).

Definition 2.7. A matching M is *internally stable* (or I-stable) in G if M is stable in the subgraph of G induced by $V(M)$. Equivalently, M is I-stable if there is no blocking edge xy such that $x \in V(M)$ and $y \in V(M)$.

For an L-blocking edge xy, Definition 2.5 only requires that $x \in V(M)$ or $y \in V(M)$. So, L-stability implies I-stability: for a matching M,

$$M \text{ stable} \implies M \text{ L-stable} \implies M \text{ I-stable.} \tag{1}$$

Tan (1990) observed that every graph has a nonempty I-stable matching (because every edge constitutes by itself an I-stable matching). This shows that I-stability is distinct from L-stability.

Tan (1990) says that a matching is "maximum stable" if it is an I-stable matching of maximum size. This terminology is misleading, since it is known that all stable matchings of a graph have the same size; we find it more appropriate to speak of "*maximum I-stable matchings*". Tan (1990) showed that a maximum I-stable matching can be computed in polynomial time; see also Tan (1991b) and Manlove (2013).

3. L-stable matchings: Structural properties

3.1. L-stability vs stability

We have already noted that every stable matching is L-stable. In this section, we further examine some relations between stable and L-stable matchings. We focus on the graph $G = (V, E)$ and we do not explicitly mention the preference table which defines it.

Proposition 3.1. *A perfect matching is L-stable if and only if it is stable.*

Proof. This follows from the trivial observation that when a matching is perfect, every edge intersects it and hence, every blocking edge is L-blocking. □

This observation can be slightly generalized for the case of non-perfect matchings. The following proposition provides a characterization which can be compared with the Definition 2.7 of I-stable matchings.

Proposition 3.2. *A matching M is locally stable in $G = (V, E)$ if and only if M is stable in $G_M = (V, K)$, where K is the set of edges that have at least one endpoint in $V(M)$.*

Proof. If M is not stable in $G_M = (V, K)$, it means that there is an edge $e \in K$ which blocks M. But then, e L-blocks M in G, and M is not L-stable in G.

Conversely, if M is not L-stable in $G = (V, E)$, then there is an edge $e \in E$ which L-blocks M, and this edge must be in K. So, G is not stable. □

We next address some important properties of (inclusion-wise) maximal L-stable matchings. Gusfield and Irving (1989) observed that all stable matchings cover the same subset of vertices (see their Theorem 4.5.2). We are going to show that the same property holds for maximal L-stable matchings, but we first need some additional terminology.

A *path* in a graph $G = (V, E)$ is a sequence $P = (v_0, v_1, \ldots, v_k)$ where $k \geq 0$ is the *length* of the path, $v_0, v_1, \ldots, v_k$ are pairwise distinct vertices, except possibly $v_0 = v_k$, and $e_i = v_i v_{i+1} \in E$ are pairwise distinct edges for $i = 0, \ldots, k-1$. When $v_0 = v_k$ and $k \geq 3$, we say that P is a *cycle*.

Denote by $S \oplus T = (S \setminus T) \cup (T \setminus S)$ the *symmetric difference* of two sets S, T. Recall that we look at a matching as being a set of edges. For two matchings M and M', an *alternating* path (resp., cycle) is a path (resp., cycle) whose edges alternate between edges of M and edges of M'. Notice that if M and M' are two matchings, then each connected component of $H = (V, M \oplus M')$ is either an alternating path (possibly of length 0, i.e., an *isolated vertex*) or an alternating cycle of even length; see, e.g., Section 10.1 in Papadimitriou and Steiglitz (1982) for similar observations. An edge of H is *isolated* if it is a connected component (i.e., a path of length 1) of H.

Lemma 3.3. *If M and M' are two locally stable matchings of $G = (V, E)$, then each connected component of $H = (V, M \oplus M')$ is either an isolated vertex, or an isolated edge, or an alternating cycle of even length.*

Proof. Assume by contradiction that $P = (v_0, v_1, \ldots, v_k)$ is an alternating path of $M \oplus M'$ with length $k \geq 2$, where v_0 and v_k have degree 1 in H. Assume further that the even edges $v_{2i-1}v_{2i}$ are in $M \setminus M'$, for $1 \le i \le \frac{k}{2}$, and the odd edges $v_{2i}v_{2i+1}$ are in $M' \setminus M$, for $0 \le i \le \frac{k-1}{2}$.

Observe that v_1 prefers v_2 to v_0, since otherwise $e_0 = v_0v_1$ (which is not in M) would be L-blocking for M.

Assume now that i ($1 \le i \le k-2$) is any index such that v_i prefers v_{i+1} to v_{i-1}. Then, it must be the case that v_{i+1} prefers v_{i+2} to v_i: otherwise, the edge $e_i = v_iv_{i+1}$ would be L-blocking for the matching (either M or M') which contains e_{i-1} and e_{i+1}.

From the above two facts, we deduce (by induction) that v_{k-1} prefers v_k to v_{k-2}. But then, the last edge $e_{k-1} = v_{k-1}v_k$ is L-blocking for the matching (either M or M') which contains $e_{k-2} = v_{k-2}v_{k-1}$: contradiction. □

Corollary 3.4. *If M is a stable matching and M' is an L-stable matching of $G = (V, E)$, then $V(M') \subseteq V(M)$ and $|M'| \le |M|$.*

Proof. By Lemma 3.3, each connected component of $H = (V, M \oplus M')$ is either an isolated vertex, or an isolated edge, or an alternating cycle of even length. If an isolated vertex is in $V(M')$, then it is in $V(M)$ as well. Every vertex in an alternating cycle is both in $V(M)$ and in $V(M')$. If $e \in M' \setminus M$ is an isolated edge of H, then e blocks M, contradicting the stability of M. So, every isolated edge of H must be in M. This implies that $V(M') \subseteq V(M)$. □

Let us say that an L-stable matching is *maximum* if it has the largest possible size among all L-stable matchings.

Corollary 3.5. *If a graph has a stable matching, then*

1. *all its stable matchings cover the same set of vertices, and*
2. *all its stable matchings are maximum L-stable.*

Proof. Let M be a stable matching. If another matching M' is stable, then from Corollary 3.4 we get $V(M) = V(M')$. If M' is locally stable, then $|M'| \le |M|$, meaning that M is maximum among all L-stable matchings. □

Remark 3.6. The first statement in Corollary 3.5 is (a part of) Theorem 4.5.2 in Gusfield and Irving (1989) where it is established in an algorithmic framework. It follows here directly from Lemma 3.3. The second statement can also be derived from Proposition 4.15 in Manlove (2013), which states that every stable matching is maximum internally stable. Indeed, let M be a stable matching and let M' be a maximum L-stable matching. Since L-stability implies I-stability (see Eq. (1)), M' is I-stable. So, $|M| \ge |M'|$ by Proposition 4.15 in Manlove (2013). □

Let us next say that an L-stable matching is *maximal* if it is not included (edge-wise) in any other L-stable matching.

Theorem 3.7. *All maximal L-stable matchings of a graph cover the same set of vertices and hence, they have the same size.*

Proof. Let M and M' be two maximal L-stable matchings and assume that $V(M')$ contains a vertex that is not in $V(M)$. By Lemma 3.3, this vertex must be the endpoint of an isolated edge of $H = (V, M \oplus M')$, and this edge is contained in $M' \setminus M$. Let F be the nonempty subset containing all isolated edges of $M' \setminus M$. Consider the matching $W = M \cup F$. We are going to prove that W is L-stable, thus contradicting the maximality of M.

Assume by contradiction that W is not L-stable, and let $e = uv$ be an L-blocking edge for W. So, u prefers v to its mate $W(u)$ in W, and v prefers u to its mate $W(v)$ in W (possibly, $W(u)$ or $W(v)$ do not exist, if u or v is not matched in W).

If e intersects $V(M)$, then e is L-blocking for M (which is a contradiction). Indeed, say u is matched in M,

- u prefers v to $M(u) = W(u)$;
- in M, v is either unmatched, or it is matched to $M(v) = W(v)$ but v prefers u to $M(v)$; in both cases, v prefers to be matched with u.

So, we can assume that e does not intersect $V(M)$, i.e., $u \notin V(M)$ and $v \notin V(M)$. Since e is L-blocking for W, it must intersect $V(F)$: say that u is the endpoint of an isolated edge $ux \in F$ (i.e., $M'(u) = W(u) = x$). Let us show that e is L-blocking for M' (a contradiction again).

We have that

- u prefers v to $M'(u) = W(u) = x$;
- in M', v is either unmatched or v is matched to some vertex $z = M'(v)$.

  (1) If v is unmatched in M', then $e = uv$ is L-blocking for M'.

  (2) If $z = M'(v)$, then $vz \in M' \setminus M$ (because $v \notin V(M)$). Hence, $vz \in F$, and $W(v) = z$. But by assumption, v prefers u to its mate in W, so v prefers u to $z = M'(v)$.

  We conclude in all cases that $e = uv$ is L-blocking for M', contradicting the assumption that M' is L-stable.

□

Theorem 3.7 implies that in order to solve the L-stable roommates problem, we only have to find an arbitrary maximal L-stable matching. In order to achieve this objective, it would be handy to be able to identify the subset of vertices that are covered by all maximal L-stable matchings. The proof of the theorem does not provide much information about this set. But we will see in Section 4 that it can be identified in polynomial time.

Let us conclude this section with a result which provides a complement to the second statement of Corollary 3.5.

Theorem 3.8. *If a graph has a stable matching, then all its maximal L-stable matchings are stable.*

Proof. Let M be a stable matching and let M' be a maximal L-stable matching. Corollary 3.5 implies that M is a maximum, and hence maximal L-stable matching. Theorem 3.7 implies in turn that $V(M) = V(M')$.

If M' is not stable, there is an edge $e = uv$ that blocks M'. Neither u nor v is in $V(M') = V(M)$, since otherwise e would be L-blocking for M'. In other words, u and v are both unmatched in M, hence e blocks M, and M is not stable: contradiction. □

The previous result can be rephrased as follows: if M is maximal L-stable, then either M is stable or there is no stable matching. In a sense, this shows that L-stability is an irrelevant concept when there is a stable matching. Similar observations apply to I-stable matchings; see Manlove (2013).

3.2. L-stability and stable partitions

3.2.1. Stable partitions

Tan (1991a,b) revealed the role of *stable partitions* and of *odd parties* in connection with stable matchings. We need to recall his terminology and some of his main results since we heavily rely on them in the sequel.

Definition 3.9. Consider a roommates instance defined by a preference table and the associated graph $G = (V, E)$. The *directed graph* (or *digraph*) $D = (V, A)$ is obtained by replacing each edge vw of G by two directed *arcs* (v, w) and (w, v) (in the terminology of Bang-Jensen and Gutin (2009), D is the complete biorientation of G). A *directed cycle* or *dicycle* in D is a sequence $C = (v_0, v_1, \dots, v_{k-1}, v_0)$ where $k \geq 2$ is the *length* of the dicycle, $v_0, v_1, \dots, v_{k-1}$ are pairwise distinct vertices, and $(v_i, v_{i+1}) \in A$ are pairwise distinct arcs for $i = 0, \dots, k-1$ (the indices are taken modulo k). We call v_{i+1} the *successor* of v_i in C, and we call v_{i-1} the *predecessor* of v_i.

Note that when $C = (v_0, v_1, v_0)$ is a dicycle of length 2, v_1 is both the successor and the predecessor of v_0. It will be convenient to extend the terminology to the case of a sequence $C = (v)$ consisting of a single vertex: then, we say that v is its own successor and predecessor.

Definition 3.10. For each arc $(v, w) \in A$, we denote by $r(v, w) \in \mathbb{N}$ the rank of the element w in the preference list of v: so, v prefers w to z if and only if $r(v, w) < r(v, z)$.

Definition 3.11. When $C = (v_0, v_1, \ldots, v_{k-1}, v_0)$ is a dicycle of length at least 3 in $D = (V, A)$ and when $(v_i, w) \in A$, we say that the arc (v_i, w) is *superior* with respect to C if $r(v_i, w) < r(v_i, v_{i-1})$ (i.e., if v_i prefers w to its predecessor in C). Otherwise, (v_i, w) is *inferior* with respect to C.

Notice that if $(v_i, w) \in A$, then (v_i, w) is either superior or inferior with respect to C. In particular, (v_i, v_{i-1}) is inferior with respect to C.

Definition 3.12. When $C = (v_0, v_1, v_0)$ is a dicycle of length 2 in $D = (V, A)$ and when $(v_i, w) \in A$, we say that the arc (v_i, w) is *superior* with respect to C if $r(v_i, w) < r(v_i, v_{i-1})$ (i.e., if v_i prefers w to its predecessor in C). The arc (v_i, w) is *inferior* with respect to C if $r(v_i, v_{i-1}) < r(v_i, w)$.

According to the latter definition, (v_0, v_1) and (v_1, v_0) are neither superior nor inferior with respect to $C = (v_0, v_1, v_0)$.

Definition 3.13. When $C = (v)$ is a vertex and $(v, w) \in A$, we say that the arc (v, w) is *superior* with respect to C.

Definition 3.14. Given a preference table and the associated digraph $D = (V, A)$, a *stable partition* is a set $\Pi = \{C_1, C_2, \ldots, C_m\}$ where $C_1, C_2, \ldots, C_m$ are called the *parties* of the partition, and

1. each C_i is either a single vertex, or a dicycle of D of length at least 2;
2. each vertex of V is contained in exactly one party C_i (hence the name "partition");
3. if C_i is a dicycle of length at least 3, then each vertex of C_i prefers its successor to its predecessor;
4. if (v, w) is superior with respect to the party containing v, then (w, v) is inferior with respect to the party containing w.

In the sequel, when we mention superior or inferior arcs in relation with a stable partition, we usually omit to specify the relevant party since it is always uniquely determined (for an arc (v, w), it is the party that contains v). We say for short that a party of even length is an *even party*, and a party of odd length is an *odd party.*

Tan (1991a,b) has established the following fundamental results that we will need in the following sections. See also Manlove (2013).

Theorem 3.15. *For any preference table and the associated digraph $D = (V, A)$,*

1. *a stable partition always exists and can be computed in time $O(|A|)$;*
2. *there is a stable partition where all even parties are of length 2;*
3. *all stable partitions contain the same odd parties;*
4. *the instance has a perfect stable matching if and only if there is no odd party in any stable partition.*

In view of the third statement in Theorem 3.15, we do not need to refer to a specific stable partition when we mention an odd party: the odd parties are a feature of the instance itself, not of any stable partition of the instance.

Theorem 3.15 is deep and difficult. Some partial intuition for the last statement can be obtained as follows: assume that $\Pi = \{C_1, \ldots, C_m\}$ is a stable partition which only consists of dicycles of length 2. If dicycle C_i contains the vertices v_i and v_{i+1}, then $M = \{v_i v_{i+1} : i = 1, \ldots, m\}$ is a perfect matching of G. To see that M is stable, let vw be any edge in $E \setminus M$. Since M is perfect, v and w are matched in M. If v prefers w to its mate, then the arc (v, w) is superior in view of Definition 3.12. But then, the last condition in Definition 3.14 implies that (w, v) is inferior, meaning that w prefers its mate in M to v. Hence, vw is not a blocking edge for M.

3.2.2. Odd parties

We are next going to establish Theorem 3.16, which states a necessary condition for a matching to be locally stable. Our proof is inspired by the proof of the last statement in Theorem 3.15 as given by Tan (1991a), but of course, it must be significantly adapted to deal with L-stable matchings.

Theorem 3.16. *In a locally stable matching, all elements of the odd parties are unmatched.*

Proof. Let $\Pi = \{C_1, \ldots, C_m\}$ be a stable partition, let M be an L-stable matching, and let

$$Sup = \{v \in V : v \text{ is matched and } (v, M(v)) \text{ is superior }\},$$

$$Inf = \{v \in V : v \text{ is matched and } (v, M(v)) \text{ is inferior }\}.$$

We have

$$|Sup| \leq |Inf|. \tag{2}$$

Indeed, assume that $v \in Sup$. Then, by definition of a stable partition, $(M(v), v)$ must be inferior, meaning that $M(v)$ is in Inf.

Let $C = (v_0, \ldots, v_{k-1}, v_0)$ be an arbitrary party in Π. Consider an arbitrary element of C, say, v_i. We now want to show that

$$v_i \in Inf \implies v_{i+1} \in Sup. \tag{3}$$

If $|C| = 1$, then either v_i is unmatched or $(v_i, M(v_i))$ is superior, by Definition 3.13. So, $v_i \notin Inf$ and the implication (3) holds.

Assume next that $|C| \geq 2$ and let $v_i \in Inf$. So, $v_i \in V(M)$ and by definition of "inferior",

$$r(v_i, v_{i-1}) \leq r(v_i, M(v_i)), \tag{4}$$

with strict inequality if $|C| = 2$. By definition of stable partitions, $r(v_i, v_{i+1}) \leq r(v_i, v_{i-1})$, with strict inequality if $|C| > 2$. Thus, in all cases,

$$r(v_i, v_{i+1}) < r(v_i, M(v_i)),$$

meaning that v_i prefers v_{i+1} to $M(v_i)$ (strictly, i.e., $v_{i+1} \neq M(v_i)$). But the edge $v_i v_{i+1}$ cannot be L-blocking, since M is L-stable. So, it must be the case that v_{i+1} is matched, and that v_{i+1} prefers $M(v_{i+1})$ to v_i. By Definition 3.11 and Definition 3.12, we conclude that $(v_{i+1}, M(v_{i+1}))$ is superior and that v_{i+1} is in Sup. This establishes (3).

Let us next show that

$$v_i \in Inf \implies v_{i-1} \in Sup. \tag{5}$$

Let $v_i \in Inf$. The case $|C| = 1$ is trivial as above, and the case $|C| = 2$ has already been dealt with, since $v_{i-1} = v_{i+1}$ in this case. Assume now that $|C| \geq 3$. If $v_{i-1} \notin V(M)$, then the edge $v_{i-1}v_i$ is L-blocking for M because of Inequality (4) which holds again. So, $v_{i-1} \in V(M)$. In view of (3), we cannot have $v_{i-1} \in Inf$. But every arc of the form (v_{i-1}, w) is either inferior or superior; hence, v_{i-1} is in Sup. This proves (5).

We next claim that for every odd party C (viewed as a set of vertices, by a slight abuse of notations),

(i) either $|C \cap Sup| > |C \cap Inf|$,

(ii) or $C \cap V(M) = \emptyset$ and in particular, $|C \cap Sup| = |C \cap Inf| = 0$.

As already noted, if $|C| = 1$, say $C = (v)$, then either v is unmatched or $(v, M(v))$ is superior by Definition 3.13. So, the claim follows and we can assume from now on that C is an odd cycle of length at least 3.

When $C \cap V(M)$ is empty, Condition (ii) holds and we are done.

When $C \subseteq V(M)$, every vertex of C is either in Inf or in Sup. The implications (3) and (5) mean that every vertex in $C \cap Inf$ is surrounded by two vertices in $C \cap Sup$. Since C is odd, it follows that $|C \cap Sup| > |C \cap Inf|$, and Condition (i) holds.

This leaves us with the case where $C \cap V(M) \neq \emptyset$ and $C \cap V(M) \neq C$. Then, C is a disjoint union of nonempty subsets of consecutive vertices, where each subset is of the form $P = \{v_i, v_{i+1}, \ldots, v_j\}$ (indices mod $|C|$) with $v_{i-1} \notin V(M)$, $v_\ell \in V(M)$ for $i \leq \ell \leq j$, and $v_{j+1} \notin V(M)$ (possibly, $|P| = 1$, or $v_{i-1} = v_{j+1}$). In view of (5), it must be the case that $v_i \in Sup$ (because $v_i \in Inf$ would imply $v_{i-1} \in Sup$, a contradiction). Similarly, in view of (3), $v_j \in Sup$. Moreover, every element of P is either in Inf or in Sup, and no two successive elements of P can be in Inf (by (3) and (5)). It follows that $|P \cap Sup| > |P \cap Inf|$. Since this holds for any subset of the same form as P, Condition (i) follows.

The previous argument for odd parties is easily adapted to show that for every even party C, $|C \cap Sup| \geq |C \cap Inf|$. Just note that when $C \subseteq V(M)$, it may be the case that $|C \cap Sup| = |C \cap Inf|$.

To conclude the proof of the theorem, assume by contradiction that C_1 is an odd party such that $C_1 \cap V(M) \neq \emptyset$. From the previous parts of the proof, we know that $|C_1 \cap Sup| > |C_1 \cap Inf|$, and $|C_j \cap Sup| \geq |C_j \cap Inf|$ for $2 \leq j \leq m$. So,

$$|Sup| = \sum_{i=1}^{m} |C_i \cap Sup| > \sum_{i=1}^{m} |C_i \cap Inf| = |Inf|$$

which contradicts Inequality (2). □

Remark 3.17. Theorem 3.16 is closely related to the last statement of Theorem 3.15: indeed, if there is an odd party of any size, then its vertices are unmatched in any L-stable matching, which therefore cannot be perfect.

Notice however that Theorem 3.15 states a necessary and sufficient condition for the existence of a stable matching, whereas Theorem 3.16 only provides a necessary condition for a matching to be locally stable. We will see later that this condition is not sufficient to ensure local stability. □

4. A polynomial algorithm for maximum L-stable matchings

In this section, we show that a maximum L-stable matching can be computed in polynomial time. For those readers who are familiar with the literature on the stable roommates problem, it may be tempting to believe that Irving's algorithm can be easily adapted to deal with locally stable matchings. But our attempts in this direction have been unsuccessful. Appendix A provides an example illustrating some of the obstacles that we encountered with this line of attack.

In view of these difficulties, we instead propose an algorithm based on the properties of L-stable matchings described in the previous sections, and in particular on Theorem 3.16. This theorem suggests that when looking for a (maximum) locally stable matching, all vertices of any odd party can simply be removed from the graph, together with their incident edges. But some caution must be exercised in order to effectively exploit this approach. Indeed, the rejected edges may turn out to be L-blocking for a matching of the resulting subgraph, as shown by the following example.

Example 4.1. Consider the preference table T_0 in Figure 2 and the associated graph G_0 represented in Figure 3.

T_0 :

a	d	i	b	
b	a	g	h	c
c	b	d		
d	c	a	e	
e	f	d		
f	e			
g	h	b		
h	b	g		
i	j	k	a	
j	k	i		
k	i	j		

Figure 2: Initial table T_0 for Example 4.1

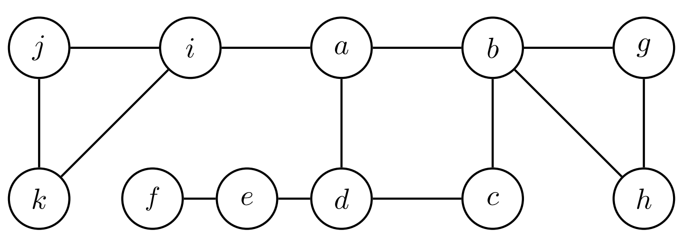

Figure 3: The graph G_0 associated with T_0 in Example 4.1

This instance admits the stable partition

$$\pi_0 = \{(a,b,a),(c,d,c),(e,f,e),(g,h,g),(i,j,k,i)\}.$$

So, $\mathcal{O} = \{i,j,k\}$ is the unique odd party. Deleting the set $\mathcal{O}$ and its incident edges from G_0 yields a graph associated with Table T_1 displayed in Figure 4. The subgraph

T_1 :

a	d	b		
b	a	g	h	c
c	b	d		
d	c	a	e	
e	f	d		
f	e			
g	h	b		
h	b	g		

Figure 4: Table T_1 obtained after deleting the odd party

associated with T_1 has a stable matching $M = \{ab, cd, ef, gh\}$. But M is not L-stable in G_0 since it is L-blocked by the edge ai. □

This example shows that simply deleting the odd parties from the initial instance does not directly lead to a solution of L-SRP. In order to deal with such situations, we will apply a sequence of valid transformations to the initial graph G. These transformations iteratively identify subsets of vertices $V_0^r \subseteq V_1^r \subseteq V_2^r \ldots$ which we can prove are not matched in any L-stable matching, and subsets of edges $E_0^r \subseteq E_1^r \subseteq E_2^r \ldots$ which are never included in any L-stable matching.

To state this strategy more accurately, let us define a constrained version of the L-stable roommates problem, to be denoted as CL-SRP. In this definition, for brevity, we implicitly assume that a preference table is given and we do not explicitly mention it as part of the instance.

Definition 4.2. An instance of CL-SRP is a triplet $I = (G, V^r, E^r)$ consisting of the graph $G = (V, E)$ associated with the preference table, a subset of vertices $V^r \subseteq V$, and a subset of edges $E^r \subseteq E$. A *solution* of CL-SRP for the instance $I = (G, V^r, E^r)$ (or a solution of I for short) is a matching of G, say M, such that

(S1) M is L-stable in G,

(S2) $V(M) \cap V^r = \emptyset$,

(S3) $M \cap E^r = \emptyset$.

In view of conditions (S2)-(S3), we say that V^r is the set of *rejected vertices*, and that E^r is the set of *rejected edges* of the instance.

Remark 4.3. The concept of rejected edge shares some analogy with the concept of "forbidden edges" considered, for example in Fleiner et al. (2007, 2011). Contrary to forbidden edges, however, rejected edges will not be arbitrarily chosen (as part of the input of the instance) in our context, but they will turn out to be intrinsic features of the initial graph G, as will be explained in the next sections. □

Observe that for a given preference table, a matching M is L-stable in G if and only if M is a solution of the instance $I_0 = (G, V_0^r = \emptyset, E_0^r = \emptyset)$ of CL-SRP. In other words, if $\mathcal{S}$ is the set of L-stable matchings of G, then $\mathcal{S}$ also is the set of solutions of I_0.

In the next sections, starting with the instance I_0, we are going to produce a sequence $I_0, I_1, I_2, \ldots$ of instances of CL-SRP which have the exact same solution set $\mathcal{S}$. To achieve this goal, let $I_i = (G, V_i^r, E_i^r)$, $i \geq 0$, be an arbitrary instance in the sequence, and let us show how to transform I_i into a new instance I_{i+1} with the same solutions as I_i.

4.1. Valid transformations: Rejecting vertices and edges

By Theorem 3.16, we already know that no vertex of any odd party of G can be matched in an L-stable matching. We now need an easy, but crucial consequence of this property.

Given the instance $I_i = (G, V_i^r, E_i^r)$ of CL-SRP, define G_i^- to be the graph obtained by deleting all rejected vertices and all rejected edges of I_i, that is, $G_i^- = (V \setminus V_i^r, E \setminus E_i^r)$. The transformations to be introduced below will guarantee that G_i^- is always well defined, in the sense that if either endpoint of an edge is in V_i^r, then the edge is in E_i^r.

We associate a preference relation with G_i^- in the natural way, by simply restricting the preference relation of G to the pairs contained in $E \setminus E_i^r$. In other words, the preference table associated with G_i^- is obtained by deleting the entries corresponding to E_i^r from the table associated with G.

Lemma 4.4. *Every solution of I_i is an L-stable matching of G_i^-.*

Proof. Every solution of I_i is a matching of G_i^-, by Conditions (S1)-(S3). It is L-stable in G_i^- since all edges of G_i^- are edges of G. □

Lemma 4.5. *If $\mathcal{O}_i$ is the union of the odd parties of G_i^-, then no vertex of $\mathcal{O}_i$ can be matched in a solution of I_i.*

Proof. This is a consequence of Lemma 4.4 and of Theorem 3.16. □

Lemma 4.5 will allow us to mark the vertices of $\mathcal{O}_i$ as "rejected". But to avoid the difficulty illustrated by Example 4.1, we also need to simultaneously reject some edges. This operation will be based on the next lemma.

Lemma 4.6. *Let M be a locally stable matching of $G = (V, E)$ and let x, y, z be such that $z \notin V(M)$, $xy \in E$, $xz \in E$, and x prefers z to y. Then, xy is not in M.*

Proof. Indeed, if $xy \in M$, then xz L-blocks M. $\square$

In particular, if we know that z cannot be in any L-stable matching, then it follows from the lemma that xy, where x prefers z to y, cannot be in any L-stable matching either.

We can apply this reasoning to the vertices contained in the set $\mathcal{O}_i$, since we know that these vertices can be rejected. For the instance $I_i = (G, V_i^r, E_i^r)$, define

$$\Delta_i := \{xy \in E \setminus E_i^r : \text{either } x \in \mathcal{O}_i \text{ or } \exists\, z \in \mathcal{O}_i : xz \in E \text{ and } x \text{ prefers } z \text{ to } y\}.$$

The set Δ_i contains all the edges which are not already in E_i^r, but which can be rejected either because one of their endpoints can be rejected, or by virtue of Lemma 4.6. Note that the preference relation between x, y, z is the initial relation associated with $G = (V, E)$, not the relation restricted to G_i^-; that is, it may be the case that the edge xz is in E_i^r and hence, is not in G_i^-; this situation is illustrated in Appendix B.

We are now ready to describe the $(\mathcal{O}, \Delta)$-*transformation* of an instance and to prove that it is valid. This transformation adds to V_i^r all the vertices of $\mathcal{O}_i$, and it adds to E_i^r all the edges of Δ_i. It has no effect if $\mathcal{O}_i$ is empty.

Proposition 4.7. *Let $I_i = (G, V_i^r, E_i^r)$. A matching M is a solution of CL-SRP for I_i if and only if M is a solution of CL-SRP for*

$$I_{i+1} = (G,\ V_{i+1}^r := V_i^r \cup \mathcal{O}_i,\ E_{i+1}^r := E_i^r \cup \Delta_i).$$

Proof. ($\Longleftarrow$) Any solution of I_{i+1} is a solution of I_i since $V_i^r \subseteq V_{i+1}^r$ and $E_i^r \subseteq E_{i+1}^r$.

($\Longrightarrow$) Let M be a solution of I_i. Condition (S1) is identical for I_i and for I_{i+1}. In view of Lemma 4.5, $V(M) \cap \mathcal{O}_i = \emptyset$, so that Condition (S2) holds for I_{i+1} when it holds for I_i.

To establish Condition (S3) for I_{i+1}, let $xy \in E_{i+1}^r = E_i^r \cup \Delta_i$ and let us show that $xy \notin M$. If $xy \in E_i^r$, then $xy \notin M$ in view of Condition (S3) for I_i, and we are done.

Assume next that $xy \in \Delta_i$. If $x \in \mathcal{O}_i$, then $xy \notin M$ by Lemma 4.5, and we are done again. So, $x \in V$ and that there exists $z \in \mathcal{O}_i$ such that x prefers z to y. By Lemma 4.5, $z \notin V(M)$, and Lemma 4.6 implies that $xy \notin M$. $\square$

Example 4.8 (Example 4.1 continued)**.** Consider the preference table T_0 and its associated graph G depicted in Figure 3. Since $G_0^- = G$, the vertex set of the unique odd party of G_0^- is $\mathcal{O}_0 = \{i, j, k\}$. Applying the $(\mathcal{O}, \Delta)$-transformation to the instance $I_0 = (G, \emptyset, \emptyset)$ yields $I_1 = (G, V_1^r, E_1^r)$ with $V_1^r = \mathcal{O}_0 = \{i, j, k\}$ and $E_1^r = \Delta_0 = \{ab, ai, ij, ik, jk\}$. □

Let us assume that we apply a sequence of $(\mathcal{O}, \Delta)$-transformations to produce a sequence of instances $I_0, I_1, I_2, \ldots$. This can be done as long as, in iteration i, the graph G_i^- has at least one odd party. Then, the set $\mathcal{O}_i$ is nonempty and the set of rejected vertices grows after each application of the $(\mathcal{O}, \Delta)$-transformation. So, after at most $|V|$ iterations, the sequence of $(\mathcal{O}, \Delta)$-transformations produces an instance of CL-SRP, say $I_p = (G, V_p^r, E_p^r)$, such that G_p^- has no odd party.

Example 4.9 (Example 4.8 continued)**.** For $G_1^- = (V \setminus V_1^r, E \setminus E_1^r)$, we find the stable partition

$$\pi_1 = \{(a), (b, g, h, b), (c, d, c), (e, f, e)\},$$

with two odd parties (a) and (b, g, h, b). So, $\mathcal{O}_1 = \{a, b, g, h\}$. Performing an $(\mathcal{O}, \Delta)$-transformation again, we obtain $V_2^r = \{a, b, g, h, i, j, k\}$ and $E_2^r = E \setminus \{ef\}$: all the edges are rejected, except ef. For example, $xy = de$ is rejected because $z = a \in V_1^r$ and $x = d$ prefers $z = a$ to $y = e$. Therefore, the transformation yields $I_2 = (G, V_2^r, E \setminus \{ef\})$.

In G_2^-, a stable partition is given by

$$\pi_2 = \{(c), (d), (e, f, e)\}.$$

Hence, a third application of the $(\mathcal{O}, \Delta)$-transformation yields $I_3 = (G, V \setminus \{e, f\}, E \setminus \{ef\})$. Finally, G_3^- only has two vertices e and f and one edge ef. This graph has no odd party and $p = 3$ in this case. We see that $M = \{ef\}$ is a perfect stable matching of G_3^- and this matching also happens to be the unique maximum L-stable matching of G_0. We will prove hereunder, in Theorem 4.14, that it is no coincidence. □

Let us record two straightforward, but crucial, consequences of the previous discussion.

Proposition 4.10. *For a matching M of G, the following statements are equivalent:*

1. *M is L-stable in G;*
2. *M is a solution of $I_0 = (G, \emptyset, \emptyset)$;*
3. *M is a solution of $I_p = (G, V_p^r, E_p^r)$.*

Proof. The first two statements are equivalent in view of Definition 4.2, the second and third statements are equivalent due to Proposition 4.7. □

Proposition 4.11. *The graph $G_p^- = (V \setminus V_p^r, E \setminus E_p^r)$ has a perfect stable matching.*

Proof. By definition of I_p, G_p^- has no odd party and hence, by Theorem 3.15, it has a stable perfect matching (possibly empty if $V \setminus V_p^r$ is empty). □

For the sake of completeness, it is easy to verify that a perfect stable matching in G_p^- is nonempty if and only if G_p^- itself is nonempty, that is, if and only if $V \setminus V_p^r \neq \emptyset$.

In the next section, we will prove that every perfect stable matching of G_p^- is a maximum locally stable matching of G, and conversely. This will immediately yield an efficient algorithm for the L-stable roommates problem.

4.2. Main results

Let $G = (V, E)$ and $G_p^- = (V \setminus V_p^r, E \setminus E_p^r)$ be defined as above.

Proposition 4.12. *Every locally stable matching of G is a locally stable matching of G_p^-.*

Proof. Let M be L-stable in G. By Proposition 4.10, M is a solution of I_p. In particular, $V \cap V_p^r = E \cap E_p^r = \emptyset$ and hence, M is a matching of G_p^-. Moreover, M is L-stable in G_p^- since all edges of G_p^- are in E. □

Proposition 4.13. *Every perfect stable matching of G_p^- is a locally stable matching of G.*

Proof. Let M be a perfect stable matching of G_p^-. So, $V(M) = V \setminus V_p^r$. Assume by contradiction that M is not L-stable in G, that $xy \in E$ is L-blocking for M in G, and that x is matched, say, $xu \in M$ for some $u \in V \setminus V_p^r$. By definition of L-blocking edges, x prefers y to u, and y either is unmatched or prefers x to its mate.

Consider first the case where y is matched in M. Then y is a vertex of G_p^-, and $xy \in E_p^r$ since otherwise xy would block M in G_p^-. For xy to be in E_p^r, it must have been rejected during the sequence of $(\mathcal{O}, \Delta)$-transformations: say, $xy \in \Delta_i$ for some $i < p$. On the other hand, x and y are in $V(M) = V \setminus V_p^r$, meaning that these two vertices have never been rejected. In particular, $x \notin \mathcal{O}_i$ and $y \notin \mathcal{O}_i$. So, by definition of Δ_i, there exists $z \in \mathcal{O}_i$ such that x prefers z to y (or y prefers z to x, but the case is symmetric on x and y). Since x also prefers y to u, transitivity implies that x prefers z to u. But then, $xu \in E_i^r \cup \Delta_i$, and it follows that $xu \in E_{i+1}^r \subseteq E_p^r$. This contradicts the assumption that xu is an edge of the matching M in G_p^-.

Consider now the case where y is unmatched in M. Then y is not a vertex of G_p^-, i.e., $y \in V_p^r$. So, $y \in \mathcal{O}_i$ for some $i < p$. But then, the definition of Δ_i implies that $xu \in E_i^r \cup \Delta_i$ since x prefers y to u. It follows again that $xu \in E_p^r$, in contradiction with $xu \in M$.

This concludes the proof. □

We are now ready for our main result.

Theorem 4.14. *Every perfect stable matching of G_p^- is a maximal locally stable matching of G, and conversely.*

Proof. Let M_p be a perfect stable matching of G_p^-. Such a matching always exists by Proposition 4.11.

By Proposition 4.13, M_p is L-stable in G. To establish that M_p is a maximal L-stable in G, let M be any L-stable matching of G such that $M_p \subseteq M$. Then M is a matching of G_p^- (as in the proof of Proposition 4.12) and $M = M_p$ since M_p is perfect in G_p^-.

Conversely, let M be a maximal locally stable matching of G. By Proposition 4.12, M is a locally stable matching of G_p^-. Moreover, as M and M_p are both maximal L-stable in G, by Theorem 3.7, $V(M) = V(M_p)$. So, M is a perfect matching of G_p^- and it is stable. □

Remark 4.15. Note that the converse of Proposition 4.12 does not hold in general: it may happen that an L-stable matching of G_p^- is not L-stable in G. To see it, consider Figure 5, which displays a simplified version of Example 4.1.

a	d	i	b
b	a	c	
c	b	d	
d	c	a	
i	j	k	a
j	k	i	
k	i	j	

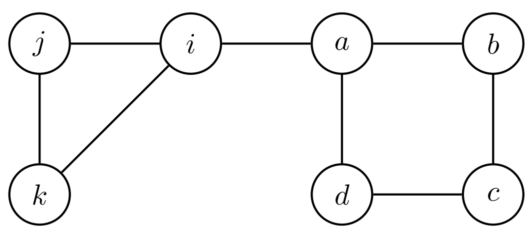

Figure 5: A preference table and its associated graph

There is a single odd party with $\mathcal{O}_0 = \{i, j, k\}$ and $\Delta_0 = \{ab, ai, ij, ik, jk\}$. So, G_1^- has four vertices and three edges $\{ad, bc, cd\}$. This graph has no odd party, so that $G_p^- = G_1^-$. The single-edge matching $M = \{bc\}$ is L-stable in G_1^-, but not in G where it is L-blocked by ab. On the other hand, $M_1 = \{ad, bc\}$ is perfect stable in G_1^- and maximal L-stable in G, as predicted by Theorem 4.14. □

4.3. Complexity of the algorithm

Consider Algorithm 1: starting from $I_0 = (G, \emptyset, \emptyset)$, the algorithm applies $(\mathcal{O}, \Delta)$-transformations as long as they modify the instance. This transformation preserves all the solutions of CL-SRP and does not create new ones. The number of vertices of G^- decreases with each iteration of the **repeat** loop. So, the loop terminates after at most $O(|V|)$ iterations.

Tan (1991a,b) described an $O(|E|)$ algorithm to compute a stable partition where all even parties are of length 2; see also Manlove (2013), Glitzner and Manlove (2024). So, the algorithm simultaneously identifies a perfect stable matching when there is one, which is the case in the last iteration of the **repeat** loop.

The last question to deal with is the complexity of computing Δ in Line 9. It can be analyzed as follows. For each vertex $x \in V$, let $L(x)$ be the (ordered) preference list of x in the table T. Assume that the set $\mathcal{O}$ has been computed in Line 8 of the algorithm. Then, for each $x \in V$, do

- if $x \in \mathcal{O}$: add xy to Δ for all y in $L(x)$ such that $xy \in E \setminus E^r$;
- if $x \notin \mathcal{O}$: for each z in $L(x) \cap \mathcal{O}$, add xy to Δ for all y such that $xy \in E \setminus E^r$ and such that x prefers z to y.

In this procedure, each element of the table is examined exactly once. An appropriate data structure can be used to test in constant time whether a vertex is in $\mathcal{O}$ or an edge is in E^r. So, Δ can be computed in time $O(E)$ in each iteration, and the overall time complexity of Algorithm 1 is $O(|V||E|)$.

Algorithm 1 Finding a maximum L-stable matching

1: **function** MAXLSTABLE(T: a preference table)
2:     Build $G = (V, E)$, the graph associated with T
3:     $V^r \leftarrow \emptyset$ ▷ Initialize V^r
4:     $E^r \leftarrow \emptyset$ ▷ Initialize E^r
5:     **repeat**   ▷ At most $|V|$ times
6:         $G^- \leftarrow (V \setminus V^r, E \setminus E^r)$ ▷ Delete V^r and E^r
7:         $\pi \leftarrow$ stable partition of G^- ▷ $O(|E|)$
8:         $\mathcal{O} \leftarrow$ union of odd parties of π
9:         $\Delta \leftarrow \{xy \in E \setminus E^r : x \in \mathcal{O} \text{ or } \exists\, z \in \mathcal{O} \text{ s.t. } x \text{ prefers } z \text{ to } y\}$
10:         $V^r \leftarrow V^r \cup \mathcal{O}$
11:         $E^r \leftarrow E^r \cup \Delta$
12:     **until** $\mathcal{O} = \emptyset$ ▷ G^- has a perfect stable matching
13:     $M \leftarrow$ perfect stable matching of G^- ▷ Given by π
14:     **return** M
15: **end function**

5. Conclusions and future lines of research

Local stability is a natural extension of the classical stability concept for the roommates problem. In this paper, we have established several important properties of locally stable matchings and of the L-stable roommates problem. As a main result, we have described an algorithm which identifies a maximum L-stable matching in polynomial time.

Many interesting questions could still be investigated in connection with L-stable matchings, just as they have been for stable matchings (see Gusfield and Irving (1989) and Manlove (2013) for lists of such questions). Certain previous results about SRP actually have immediate implications for L-SRP. For example, Gusfield (1988) has shown that all stable matchings can be enumerated efficiently. In view of Theorem 4.14, the same property holds for maximal (and maximum) L-stable matchings. As another example, Baratto et al. (2025) observed that Lemma 1 in Manlove et al. (2002) immediately implies that computing a maximum L-stable matching is NP-hard when the preferences are not strict, i.e., when ties are allowed in the preference lists.

By contrast, many questions about L-stable matchings remain open. In particular, our algorithm for L-SRP computes a maximum L-stable matching in polynomial time, but it does not provide a concise certificate of optimality for the output (other than the trace of the algorithm itself). In the case of the stable roommates problem, any stable partition provides a concise certificate for the existence of a stable matching. It would be nice to obtain a similar result for maximum L-stable matchings.

On a different note, linear programming formulations of the stable roommates have been investigated and have proved numerically useful, for example, in Teo and Sethuraman (1998, 2000). Here again, it would be interesting to adapt this line of work to the case of L-stable matchings. These are but a few examples of open questions about L-stable matchings, many more would be worth investigating.

References

Agoston, K.C., Biró, P., Kováts, E., Jankó, Z., 2022. College admissions with ties and common quotas: Integer programming approach. European Journal of Operational Research 299, 722–734. doi:https://doi.org/10.1016/j.ejor.2021.08.033.

Bang-Jensen, J., Gutin, G.Z., 2009. Digraphs: Theory, Algorithms and Applications. Springer Monographs in Mathematics. second ed., Springer, London.

Baratto, M., Crama, Y., Pedroso, J.P., Viana, A., 2025. Local stability in kidney exchange programs. European Journal of Operational Research 320, 20–34.

Biró, P., 2007. The stable matching problem and its generalizations: An algorithmic and game theoretical approach. Ph.D. thesis. Budapest University of Technology and Economics. Budapest.

Biró, P., McDermid, E., 2010. Three-sided stable matchings with cyclic preferences. Algorithmica 58, 5–18.

Delorme, M., García, S., Gondzio, J., Kalcsics, J., Manlove, D., Pettersson, W., 2019. Mathematical models for stable matching problems with ties and incomplete lists. European Journal of Operational Research 277, 426–441.

Fleiner, T., Irving, R.W., Manlove, D.F., 2007. Efficient algorithms for generalized stable marriage and roommates problems. Theoretical Computer Science 381, 162–176.

Fleiner, T., Irving, R.W., Manlove, D.F., 2011. An algorithm for a super-stable roommates problem. Theoretical Computer Science 412, 7059–7065.

Gale, D., Shapley, L.S., 1962. College admissions and the stability of marriage. The American Mathematical Monthly 69, 9–15.

Glitzner, F., Manlove, D., 2024. Structural and algorithmic results for stable cycles and partitions in the roommates problem, in: Schäfer, G., Ventre, C. (Eds.), Algorithmic Game Theory, Springer Nature Switzerland, Cham. pp. 3–20.

Gusfield, D., 1988. The structure of the stable roommate problem: Efficient representation and enumeration of all stable assignments. SIAM Journal on Computing 17, 742–769.

Gusfield, D., Irving, R.W., 1989. The stable marriage problem: structure and algorithms. MIT press.

Huang, C.C., 2010. Circular stable matching and 3-way kidney transplant. Algorithmica 58, 137–150.

Irving, R.W., 1985. An efficient algorithm for the "stable roommates" problem. Journal of Algorithms 6, 577–595.

Klimentova, X., Biró, P., Viana, A., Costa, V., Pedroso, J.P., 2023. Novel integer programming models for the stable kidney exchange problem. European Journal of Operational Research 307, 1391–1407.

Liu, Y., Tang, P., Fang, W., 2014. Internally stable matchings and exchanges, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1433–1439.

Manlove, D.F., 2013. Algorithmics of Matching under Preferences. World Scientific Publishing, Singapore.

Manlove, D.F., Irving, R.W., Iwama, K., Miyazaki, S., Morita, Y., 2002. Hard variants of stable marriage. Theoretical Computer Science 276, 261–279.

Mészáros-Karkus, Z., 2017. Hardness results for stable exchange problems. Theoretical Computer Science 670, 68–78.

Papadimitriou, C., Steiglitz, K., 1982. Combinatorial Optimization - Algorithms and Complexity. Prentice Hall, Englewood Cliffs.

Pass-Lanneau, A., Igarashi, A., Meunier, F., 2020. Perfect graphs with polynomially computable kernels. Discrete Applied Mathematics 272, 69–74.

Ronn, E., 1990. NP-complete stable matching problems. Journal of Algorithms 11, 285–304.

Roth, A., Sönmez, T., Ünver, M.U., 2004. Kidney exchange. The Quarterly Journal of Economics 119, 457–488.

Tan, J.J.M., 1990. A maximum stable matching for the roommates problem. BIT Numerical Mathematics 30, 631–640.

Tan, J.J.M., 1991a. A necessary and sufficient condition for the existence of a complete stable matching. Journal of Algorithms 12, 154–178.

Tan, J.J.M., 1991b. Stable matchings and stable partitions. International Journal of Computer Mathematics 39, 11–20.

Teo, C.P., Sethuraman, J., 1998. The geometry of fractional stable matchings and its applications. Mathematics of Operations Research 23, 874–891.

Teo, C.P., Sethuraman, J., 2000. On a cutting plane heuristic for the stable roommates problem and its applications. European Journal of Operational Research 123, 195–205.

Appendix A. Irving's algorithm vs. L-stable matchings

We provide in this appendix a small example showing that a straightforward application of Irving's algorithm does not allow us to identify maximal L-stable matchings. Of course, it cannot be ruled out that a smarter adaptation of the algorithm would solve L-SRP, but we have not been able to find such an adaptation.

We assume that the reader is familiar with Irving's algorithm as presented in Irving (1985) or Gusfield and Irving (1989), and we skip all details of its application. Consider the preference table T_0 in Figure A.6. It is easy to check that there is no L-stable matching for T_0.

$$T_0: \quad \begin{array}{c|ccc} a & c & b & \\ b & a & e & \\ c & g & a & \\ d & e & g & f \\ e & b & d & \\ f & d & g & \\ g & f & d & c \end{array}$$

Figure A.6: A preference table T_0

Phase 1 of Irving's algorithm does not reduce the table. There is a unique exposed rotation in T_0, namely $\rho_1 = (b,a),(d,e),(c,g)$. Eliminating ρ_1 yields the table T_1 in Figure A.7

$$T_1: \quad \begin{array}{c|cc} a & c & \\ b & e & \\ c & a & \\ d & g & f \\ e & b & \\ f & d & g \\ g & f & d \end{array}$$

Figure A.7: A preference table T_1

It can already be checked that the matching $M = \{ac, be\}$ is maximal L-stable for T_1 (the situation is essentially the same as in Example 2.3). So, eliminating the rotation ρ_1 as in Irving's algorithm has produced an L-stable matching which did not exist in the initial instance.

The next step of Irving's algorithm would eliminate the rotation

$$\rho_2 = (d,g),(g,f),(f,d)$$

and yield a table T_2 containing some empty preference lists. So it would correctly establish that T_0 has no stable matching. But this step would still preserve the matching M as an L-stable matching of T_2.

Appendix B. Preference relation used to reject edges

In this section, we emphasize that the preference relation involved in the definition of the set Δ_i for an instance $I_i = (G, V_i^r, E_i^r)$ (see Section 4.1) must be the initial relation associated with $G = (V, E)$, and not the relation restricted to G_i^-.

T_0 :

a	c	b	
b	a	d	
c	e	a	
d	b	f	
e	g	f	c
f	d	e	g
g	f	e	

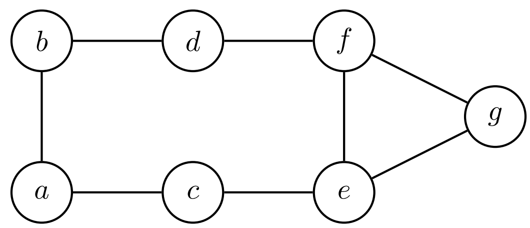

Figure B.8: A preference table with its associated graph

Consider the preference table T_0 and the associated graph G displayed in Figure B.8. This instance admits the stable partition $\pi_0 = \{(a, c, a), (b, d, b), (e, f, g, e)\}$. So, $\mathcal{O}_0 = \{e, f, g\}$. Applying the $(\mathcal{O}, \Delta)$-transformation to the instance $I_0 = (G, \emptyset, \emptyset)$ leads to $I_1 = (G, \{e, f, g\}, E_1^r)$ with

$$E_1^r = \Delta_0 = \{ac, ce, df, ef, eg, fg\}.$$

Here ac is the only edge of Δ_0 derived from Lemma 4.6. The preference table T_1 associated with G_1^- is displayed in Figure B.9. We have the following stable partition $\pi_1 = \{(a, b, a), (c), (d)\}$. It follows that $\mathcal{O}_1 = \{c, d\}$ and $\Delta_1 = \{ab, bd\}$. Observe that $ab \in \Delta_1$ because a prefers c to b with respect to the preference table T_0. Even if ac is not an edge of G_1^-, it has an impact on Δ_1.

T_1 :

a	b	
b	a	d
c		
d	b	

Figure B.9: The preference table T_1

To conclude, $E_2^r = E$ and G_2^- only contains two isolated vertices, namely, a and b. A last application of the $(\mathcal{O}, \Delta)$-transformation yields an empty graph. Hence, the only L-stable matching of G is the empty set.