Recognizing When a Preference System is Close to Admitting a Master List[⋆,⋆⋆]

Ildikó Schlotter[1,2]

[1] Centre for Economic and Regional Studies, Budapest, Hungary;
schlotter.ildiko@krtk.hu
[2] Budapest University of Technology and Economics, Budapest, Hungary

Abstract. A preference system $\mathcal{I}$ is an undirected graph where vertices have preferences over their neighbors, and $\mathcal{I}$ admits a master list if all preferences can be derived from a single ordering over all vertices. We study the problem of deciding whether a given preference system $\mathcal{I}$ is *close to* admitting a master list based on three different distance measures. We determine the computational complexity of the following questions: can $\mathcal{I}$ be modified by (i) k swaps in the preferences, (ii) k edge deletions, or (iii) k vertex deletions so that the resulting instance admits a master list? We investigate these problems in detail from the viewpoint of parameterized complexity and of approximation. We also present two applications related to stable and popular matchings.

1 Introduction

A preference system models a set of agents as an undirected graph where agents are vertices, and each agent has preferences over its neighbors. Preference systems are a fundamental concept in the area of matching under preferences which, originating in the seminal work of Gale and Shapley [16] on stable matchings, is a prominent research field in the intersection of algorithm design and computational social choice that has steadily gained attention over the last two decades.

Preference systems may admit a master list, that is, a global ranking over all agents from which agents derive their preferences. Master lists arise naturally in many practical scenarios such as P2P networks [26], job markets [21], and student housing assignments [30]. Consequently, master lists and its generalizations have been the focus of research in several papers [7, 9, 11, 21, 22, 24, 29].

In this work we aim to investigate the computational complexity of recognizing preference systems that are close to admitting a master list. Such instances may arise as a result of noise in the data set, or in scenarios where a global ranking of agents is used in general, with the exception of a few anomalies.

⋆ Supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and the Hungarian Scientific Research Fund (OTKA grants K128611 and K124171).

⋆⋆ Published in C.-C. Lin et al. (Eds.): WALCOM 2023, LNCS 13973, pp. 317–329, 2023. Final version available at https://doi.org/10.1007/978-3-031-27051-2_27.

Our Contribution. We introduce three measures to describe the distance of a given preference system $\mathcal{I}$ from the class of preference systems admitting a master list. The first measure, $\Delta^{\text{swap}}(\mathcal{I})$ is based on the swap distance between agents' preferences, while the measures $\Delta^{\text{edge}}(\mathcal{I})$ and $\Delta^{\text{vert}}(\mathcal{I})$ are based on classic graph operations, the deletion of edges or vertices; precise definitions follow in Section 2. We study in detail the complexity of computing these values for a given preference system $\mathcal{I}$. After proving that computing any of these three measures is NP-hard, we apply the framework of parameterized complexity and of approximation algorithms to gain a more fine-grained insight.

In addition to the problems of computing $\Delta^{\text{swap}}(\mathcal{I})$, $\Delta^{\text{edge}}(\mathcal{I})$, and $\Delta^{\text{vert}}(\mathcal{I})$, we briefly look at two applications. First, we show that if a strict preference system $\mathcal{I}$ is close to admitting a master list, then we can bound the number of stable matchings as a function of the given distance measure. This yields an efficient way to solve a wide range of stable matching problems in instances that are close to admitting a master list. Second, we consider an optimization problem over popular matchings where the task is to find a maximum-utility popular matching while keeping the number (or cost) of blocking edges low. We prove that this notoriously hard problem can be efficiently solved if preferences are close to admitting a master list. In both of these applications, the running time of the obtained algorithms heavily depends on the distance measure used.

Related Work. Master lists have been extensively studied in the context of stable matchings [7,11,21,22]. Various models have been introduced in the literature to generalize master lists, and capture preferences that are similar to each other in some sense. Closest to our work might be the paper by Bredereck et al. [7] who examine the complexity of multidimensional stable matching problems on instances that are close to admitting a master list. Abraham et al. investigated a setting where agent pairs are ranked globally [1]. Bhatnagar et al. [4] examined three restrictions on preference systems—the k-attribute, the k-range, and the k-list models—that aim to capture similarities among preferences; these models have been studied subsequently by several researchers [9, 24, 29].

Restricted preference profiles have been also examined in the broader context of computational social choice; see the survey by Elkind et al. [13]. In election systems, computing the Kemeny score [23] for a multiset of votes (where each vote is a total linear order over a set of candidates) is analogous to computing the value $\Delta^{\text{swap}}(\mathcal{I})$ for a preference system $\mathcal{I}$, although there are some differences between these two problems. Besides the extensive literature on the complexity of Kemeny voting (see e.g. [3,15]), our work also relates to the problem of computing certain distance measures between elections [5]. Some of the distance measures we use were considered by Gupta et al. in their paper on committee selection [18].

2 Preliminaries

We assume that the reader is familiar with basic concepts in graph theory, classic and parameterized computational complexity, and approximation theory. For directed and undirected graphs, we will use the notation of the book by Bang-Jensen and Gutin [2], unless otherwise stated. For more on complexity and approximations, we refer the reader to corresponding books [12,17,33]. We provide all definitions and notations we use (apart from those defined below), as well as all formal proofs in the full version of our paper [31].

Preference Systems. A *preference system* is a pair $\mathcal{I} = (G, \preceq)$ where G is an undirected graph and $\preceq = \{\preceq_v : v \in V(G)\}$ where $\preceq_v$ is a weak or a strict order over $N_G(v)$ for each vertex $v \in V(G)$, indicating the *preferences* of v. For some $v \in V(G)$ and $a, b \in N_G(v)$, we say that v *prefers* b to a, denoted by $a \prec_v b$, if $a \not\succeq_v b$. We write $a \sim_v b$, if $a \preceq_v b$ and $b \preceq_v a$. A *tie* in v's preferences is a maximal set $T \subseteq N_G(v)$ such that $t \sim_v t'$ for each t and t' in T. If each tie has size 1, then $\mathcal{I}$ is a *strict preference system*, and we may denote it by $(G, \prec)$.

Deletions and Swaps. For a set X of edges or vertices in G, let $\mathcal{I} - X$ denote the preference system whose underlying graph is $G - X$ and where the preferences of each vertex $v \in V(G - X)$ is the restriction of $\preceq_v$ to $N_{G-X}(v)$. We may refer to $\mathcal{I} - X$ as a sub-instance of $\mathcal{I}$.

If vertex v has strict preferences $\prec_v$ in $\mathcal{I}$, then a *swap* is a triple $(a, b; v)$ with $a, b \in N_G(v)$, and it is *admissible* if a and b are consecutive[3] in v's preferences. *Performing* an admissible swap $(a, b; v)$ in $\mathcal{I}$ means switching a and b in v's preferences; the resulting preference system is denoted by $\mathcal{I} \lhd (a, b; v)$. For a set S of swaps, $\mathcal{I} \lhd S$ denotes the preference system obtained by performing the swaps in S in $\mathcal{I}$ in an arbitrary order as long as each swap is admissible (if this is not possible, $\mathcal{I} \lhd S$ is undefined). For non-strict preferences, similar notions will be discussed in Section 3.

Master Lists. A weak or strict order $\preceq^{\mathrm{ml}}$ over $V(G)$ is a *master list* for $(G, \preceq)$, if for each $v \in V(G)$, the preferences of v are *consistent* with $\preceq^{\mathrm{ml}}$, that is, $\preceq_v$ is the restriction of $\preceq^{\mathrm{ml}}$ to $N_G(v)$. We will denote by $\mathcal{F}_{\mathrm{ML}}$ the family of those preference systems that admit a master list. Notice that $\mathcal{F}_{\mathrm{ML}}$ is closed under taking subgraphs: if we delete a vertex or an edge from a preference system in $\mathcal{F}_{\mathrm{ML}}$, the remainder still admits a master list.

3 Problem Definition and Initial Results

Let us first introduce the notion of a *preference digraph*, a directed graph associated with a given preference system, which can be exploited to obtain a useful characterization of preference systems that admit a master list. We then proceed with defining our measures for describing the distance from $\mathcal{F}_{\mathrm{ML}}$.

Characterization of $\mathcal{F}_{\mathrm{ML}}$ through the Preference Digraph. With a strict preference system $\mathcal{I} = (G, \prec)$ where $G = (V, E)$, we associate an arc-labelled directed graph $D_{\mathcal{I}}$ that we call the *preference digraph* of $\mathcal{I}$. We let $D_{\mathcal{I}}$ have the same set of vertices as G, and we define the arcs in $D_{\mathcal{I}}$ by adding an arc (a, b) labelled with v whenever $a \prec_v b$ holds for some vertices a, b and v in V. Note that

[3] Vertices a and b are consecutive in v's preferences, if either $a \prec_v b$ but there is no vertex c with $a \prec_v c \prec_v b$, or $b \prec_v a$ but there is no vertex c with $b \prec_v c \prec_v a$.

several parallel arcs may point from a to b in $D_{\mathcal{I}}$, each having a different label, so we have $|V(D_{\mathcal{I}})| = |V|$ but $|A(D_{\mathcal{I}})| = O(|V||E|)$. Observation 1 immediately follows from the fact that acyclic digraphs admit a topological order.

Observation 1 *A strict preference system* $(G, \prec)$ *admits a master list if and only if the preference digraph of* G *is acyclic.*

For a preference system $\mathcal{I} = (G, \preceq)$ with $G = (V, E)$ that is not necessarily strict we extend the concept of the preference digraph of $\mathcal{I}$ as follows. Again, we let $D_{\mathcal{I}}$ have V as its vertex set, but now we add two types of arcs to $D_{\mathcal{I}}$: for any v in V and $a, b \in N_G(V)$ with $a \neq b$ we add a *strict arc* (a, b) with label v whenever $a \prec_v b$, and we add a pair of *tied arcs* (a, b) and (b, a), both with label v, whenever $a \sim_v b$. Note that this way we indeed generalize our definition above for the preference digraph of strict preference systems. We will call a cycle of $D_{\mathcal{I}}$ that contains a strict arc a *strict cycle*. The following lemma is a straightforward generalization of Observation 1.

Lemma 2 *A preference system* $(G, \preceq)$ *admits a master list if and only if no cycle of the preference digraph of* G *is strict.*

Measuring the Distance from $\mathcal{F}_{\mathrm{ML}}$. Let us now define our three measures to describe the distance of a given strict preference system $\mathcal{I} = (G, \prec)$ from the family $\mathcal{F}_{\mathrm{ML}}$ of preference systems that admit a master list:

- $\Delta^{\mathrm{swap}}(\mathcal{I}) = \min\{|S| : S$ is a set of swaps in $\mathcal{I}$ such that $\mathcal{I} \lhd S \in \mathcal{F}_{\mathrm{ML}}\}$;
- $\Delta^{\mathrm{edge}}(\mathcal{I}) = \min\{|S| : S \subseteq E(G), \mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}\}$;
- $\Delta^{\mathrm{vert}}(\mathcal{I}) = \min\{|S| : S \subseteq V(G), \mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}\}$.

The measures $\Delta^{\mathrm{edge}}(\mathcal{I})$ and $\Delta^{\mathrm{vert}}(\mathcal{I})$ can be easily extended for preference systems that are not necessarily strict, since the above definitions are well-defined for any preference system $(G, \preceq)$.

Extending the measure $\Delta^{\mathrm{swap}}(\mathcal{I})$ for non-strict preference systems is, however, not entirely straightforward. If there are ties in the preferences of some vertex v, how can we define an admissible swap? In this paper we use the following definition for swap distance, which seems to be standard in the literature [6, 8]. Let $\preceq_u$ and $\preceq_v$ be weak orders. If they are not defined over the same sets, then the *swap distance* of $\preceq_u$ and $\preceq_v$, denoted by $\Delta(\preceq_u, \preceq_v)$ is ∞, otherwise

$$\Delta(\preceq_u, \preceq_v) = |\{\{a, b\} : a \prec_u b \text{ but } b \preceq_v a\}| + |\{\{a, b\} : a \sim_u b \text{ but } a \not\sim_v b\}|.$$

For two preferences systems $\mathcal{I} = (G, \preceq)$ and $\mathcal{I}' = (G', \preceq')$ with $G = (V, E)$ and $G' = (V', E')$, we let their swap distance, denoted by $\Delta(\mathcal{I}, \mathcal{I}')$, be ∞ if they are not defined over the same vertex set; otherwise (that is, if $V = V'$) we let $\Delta(\mathcal{I}, \mathcal{I}') = \sum_{v \in V} \Delta(\preceq_v, \preceq'_v)$. Using this, we can define

$$\Delta^{\mathrm{swap}}(\mathcal{I}) = \min \{\Delta(\mathcal{I}, \mathcal{I}') : \mathcal{I}' \in \mathcal{F}_{\mathrm{ML}}\}.$$

The following lemma follows easily from the definitions.

Lemma 3 $\Delta^{\mathrm{swap}}(\mathcal{I}) \geq \Delta^{\mathrm{edge}}(\mathcal{I}) \geq \Delta^{\mathrm{vert}}(\mathcal{I})$ *for any preference system* $\mathcal{I}$.

Let Master List by Swaps (or MLS for short) be the problem whose input is a preference system $\mathcal{I}$ and an integer k, and the task is to decide whether $\Delta^{\text{swap}}(\mathcal{I}) \leq k$. We define the Master List by Edge Deletion (or MLED) and the Master List by Vertex Deletion (or MLVD) problems analogously.

4 Computing the Distance from Admitting a Master List

Let us now present our main results on recognizing when a given preference list is close to admitting a master list. We investigate the classical and parameterized complexity of each of our problems MLS, MLED, and MLVD. In Section 4.1 we consider strict preference systems, and then extend our results for weakly ordered preferences in Section 4.2.

4.1 Strict Preferences

We show that computing the distance from $\mathcal{F}_{\text{ML}}$ is NP-hard for each of our three distance measures. However, when viewed from the perspective of approximation or of parameterized complexity, intrinsic differences between MLS, MLED, and MLVD will surface.

We start with Theorem 4 showing that we cannot expect a polynomial-time algorithm for MLS or for MLED and even a polynomial-time approximation is unlikely to exist already for bipartite graphs, assuming the so-called *Unique Games Conjecture* [25], a standard assumption in complexity theory. The proof of Theorem 4 relies on a connection between MLS, MLED, and the Feedback Arc Set problem which, given a directed graph D and an integer k, asks whether there exists a set of at most k arcs in D whose deletion from D yields an acyclic graph. Interestingly, the connection of this problem to MLS and to MLED can be used both ways: on the one hand, it serves as the basis of our reduction for proving computational hardness, and on the other hand, we will be able to apply already existing algorithms for Feedback Arc Set in our quest for solving MLS and MLED.

Theorem 4. *MLS and MLED are both* NP*-hard, and assuming the Unique Games Conjecture they are* NP*-hard to approximate by any constant factor in polynomial time. All of these hold even if the input graph is bipartite with all vertices on one side having degree 2, and preferences are strict.*

Thanks to Lemma 1 below, for any strict preference system $\mathcal{I}$ we can decide whether $\Delta^{\text{swap}}(\mathcal{I}) \leq k$ for some $k \in \mathbb{N}$ by applying the FPT algorithm of Lokshtanov et al. [27] for Feedback Arc Set on the preference digraph $D_{\mathcal{I}}$ and parameter k. Their algorithm runs in time $O(k!4^k k^6(n+m))$ on an input graph with n vertices and m arcs [27]. If $G = (V, E)$ is the graph underlying $\mathcal{I}$, then $D_{\mathcal{I}}$ has $|V|$ vertices and $O(|V|{\cdot}|E|)$ arcs, implying a running time of $O(k!4^k k^6|V|{\cdot}|E|)$.

Lemma 1. *For a strict preference system $\mathcal{I}$, $\Delta^{\text{swap}}(\mathcal{I}) \leq k$ if and only if the preference digraph of $\mathcal{I}$ admits a feedback arc set of size at most k.*

Algorithm 1 Obtaining a 2-approximation for MLED on input $(\mathcal{I}, k)$ with strict preferences

1: Construct the graph $H_{\mathcal{I}}$.
2: Let F be a solution for Feedback Arc Set on input $(H_{\mathcal{I}}, k)$.
3: Ensure that each arc in F is incident to some vertex in V by replacing all arcs of F entering some a_v^- with (a_v^-, a).
4: Return $S_F = \{\{a, v\} \in E : (a, a_v^+) \in F \text{ or } (a_v^-, a) \in F\}$.

Corollary 5 *If preferences are strict, then MLS is fixed-parameter tractable with parameter k, and can be solved in time $O(k!4^k k^6 |V| \cdot |E|)$.*

We remark that MLS for strict preferences can be formulated as a variant of the Kemeny Score problem with incomplete votes as studied by Betzler et al. [3]. Their results also imply that MLS is FPT with parameter k, though the running time we obtain in Corollary 5 is better than the one stated in [3, Theorem 10].

In contrast to MLS, the MLED problem is W[1]-hard with k as the parameter; the reduction is from Multicolored Clique [14].

Theorem 6. *MLED is* W[1]*-hard with parameter k, even for strict preferences.*

Although Theorem 6 provides strong evidence that there is no FPT algorithm for MLED with parameter k, and by Theorem 4 we cannot hope for a polynomial-time approximation algorithm for MLED either, our next result shows that combining these two approaches yields a way to deal with the computational hardness of the problem. Namely, Theorem 7 provides a 2-approximation for MLED whose running time is FPT with parameter k. This result again relies heavily on the connection between MLED and Feedback Arc Set.

Theorem 7. *There exists an algorithm that achieves a 2-approximation for MLED if preferences are strict, and runs in FPT time with parameter k.*

2-Approximation FPT Algorithm for MLED (Strict Preferences). Let the strict preference system $\mathcal{I} = (G, \prec)$ with underlying graph $G = (V, E)$ and $k \in \mathbb{N}$ be our input for MLED. See Algorithm 1 for a formal description.

First, we construct a directed graph $H_{\mathcal{I}}$ by setting

$$V(H_{\mathcal{I}}) = V \cup \{a_v^-, a_v^+ : \{a, v\} \in E\},$$
$$A(H_{\mathcal{I}}) = \{(a_c^+, b_c^-) : a, b, c \in V, a \prec_c b\} \cup \{(a_v^-, a), (a, a_v^+) : v \in V, a \in N_G(v)\}.$$

Our approximation factor relies on the property if $H_{\mathcal{I}}$ that, roughly speaking, the effect of deleting an edge from G can be achieved by deleting two arcs from $H_{\mathcal{I}}$.

Next, we compute a minimum feedback arc set F in $H_{\mathcal{I}}$ using the algorithm by Lokshtanov et al. [27]. We may assume that F only contains arcs incident to some vertex in V, as we can replace any arc (a_c^+, b_c^-) with the sole are leaving b_c^-, namely (b_c^-, b), since all cycles containing (a_c^+, b_c^-) must also go through (b_c^-, b).

Finally, we return the set $S_F = \{\{a, v\} \in E : (a, a_v^+) \in F \text{ or } (a_v^-, a) \in F\}$.

Note that $H_{\mathcal{I}}$ has $|V| + 2|E|$ vertices and at most $|V|\cdot|E| + 4|E|$ arcs. The total running time of Algorithm 1 is therefore $O(k!4^k k^6 |V|\cdot|E|)$ which is indeed FPT with parameter k.

Contrasting our positive results for MLS and MLED, a reduction from the classic HITTING SET problemshows that MLVD is computationally hard both in the classic and in the parameterized sense, and cannot be approximated by any FPT algorithm, as stated by Theorem 8.

Theorem 8. *MLVD is* NP*-hard and* W[2]*-hard with parameter* k*. Furthermore, no FPT algorithm with* k *as the parameter can achieve an* $f(k)$*-approximation for MLVD for any computable function* f*, unless* FPT = W[1]*. All of these hold even if the input graph is bipartite and preferences are strict.*

4.2 Weakly Ordered Preferences

Let us now consider preference systems that are not necessarily strict. The hardness results of Section 4.1 trivially hold for weakly ordered preferences, so we will focus on extending the algorithmic results of the previous section.

Lemma 2. *For any preference system* $\mathcal{I} = (G, \preceq)$*,* $\Delta^{\text{swap}}(\mathcal{I}) \leq k$ *if and only if there exists a set of at most* k *arcs in the preference digraph* $D_{\mathcal{I}}$ *of* $\mathcal{I}$ *that hits every strict cycle of* $D_{\mathcal{I}}$*.*

Thanks to Lemma 2, we can reduce MLS to a generalization of the FEEDBACK ARC SET problem where, instead of searching for a feedback arc set, the task is to seek an arc set that only hits certain *relevant* cycles. In the SUBSET FEEDBACK ARC SET (or SFAS) problem the input is a directed graph D, a vertex set $W \subseteq V(D)$ and an integer k, and the task is to find a set of at most k *arcs* in D that hits all *relevant* cycles in D, where a cycle is relevant if it goes through some vertex of W.

To solve SFAS, we apply an FPT algorithm by Chitnis et al. [10] for the vertex variant of SFAS, the DIRECTED SUBSET FEEDBACK VERTEX SET (or DSFVS) problem that, given a directed graph D, a set $W \subseteq V(D)$ and a parameter $k \in \mathbb{N}$, asks for a set of at most k *vertices* that hits all relevant cycles in D. Applying a simple, well-known reduction from SFAS to DSFVS, we can use the algorithm by Chitnis et al. [10] to obtain an FPT algorithm for MLS with parameter k.

Theorem 9. *MLS is fixed-parameter tractable with parameter* k*, even if preferences are weak orders.*

Next we extend Theorem 7 for weak orders, by reducing MLED to SFAS.

Theorem 10. *There exists an algorithm that achieves a 2-approximation for MLED, and runs in FPT time with parameter* k*.*

2-Approximation FPT Algorithm for MLED. Let the preference system $\mathcal{I}$ with underlying graph $G = (V, E)$ and $k \in \mathbb{N}$ be our input for MLED. For

Algorithm 2 Obtaining a 2-approximation for MLED on input $(\mathcal{I}, k)$

1: Construct the graph $H_{\mathcal{I}}$.
2: Let F be a solution for Subset Feedback Arc Set on input $(H_{\mathcal{I}}, Z, k)$.
3: Ensure $F \subseteq A_U$ by replacing all arcs pointing to some $a_v^- \in U$ with (a_v^-, a) and all arcs leaving some $a_v^+ \in U$ with (a, a_v^+).
4: Return $S_F = \{\{a, v\} \in E : (a, a_v^+) \in F$ or $(a_v^-, a) \in F\}$.

each $v \in V$, let T_v be the set family containing every tie that appears in the preferences of v. See Algorithm 2 for a formal description.

First, we construct a directed graph $H_{\mathcal{I}}$ with $V(H_{\mathcal{I}}) = V \cup T \cup U \cup Z$ where

$$T = \{t : v \in V, t \in T_v\},$$
$$U = \{a_v^-, a_v^+ : \{a, v\} \in E\},$$
$$Z = \{z_{(a,b,v)} : a \prec_v b \text{ for some } a, b, v \in V\},$$

and with arc set $A(H_{\mathcal{I}}) = A_T \cup A_U \cup A_Z$ where

$$A_T = \{(t, a_v^-), (a_v^+, t) : v \in V, t \in T_v, a \in t\}$$
$$A_U = \{(a_v^-, a), (a, a_v^+) : v \in V, a \in N_G(v)\}$$
$$A_Z = \{(a_v^+, z_{(a,b,v)}), (z_{(a,b,v)}, b_v^-) : z_{(a,b,v)} \in Z\}.$$

Next, we solve the Subset Feedback Arc Set problem $(H_{\mathcal{I}}, Z, k)$ by applying the above reduction from SFAS to DSFVS and then using the algorithm of Chitnis et al. [10]; let F be the solution obtained for $(H_{\mathcal{I}}, Z, k)$. Observe that w.l.o.g. we may assume that F only contains arcs of A_U. Indeed we can replace any arc $f \in F$ in $A_T \cup A_Z$ by an appropriately chosen arc $f' \in A_U$: note that f either points to some $a_v^- \in U$ or it leaves some $a_v^+ \in U$; in the former case we set $f' = (a_v^-, a)$, while in the latter case we set $f' = (a, a_v^+)$. Then any cycle containing f must also contain f', so we can safely replace f with f', as $F \setminus \{f\} \cup \{f'\}$ still hits all relevant cycles. Hence, we will assume $F \subseteq A_U$.

Finally, we return the set $S_F = \{\{a, v\} \in E : (a, a_v^+) \in F$ or $(a, a_v^+) \in F\}$.

It is clear that the above algorithm runs in FPT time with parameter k.

5 Applications

In this section we consider two examples related to stable and popular matchings where we can efficiently solve computationally hard optimization problems on preference systems that are close to admitting a master list; see the book [28] for the definition of stability and popularity.

5.1 Optimization over Stable Matchings

One of the most appealing property of the distances defined in Section 3 is that whenever the distance of a *strict* (but not necessarily bipartite) preference system from admitting a master list is small, we obtain an upper bound on the number of stable matchings contained in the given preference system. Therefore, strict preference systems that are close to admitting a master list are easy to handle, as we can efficiently enumerate their stable matchings, as Lemmas 11 and 13 show.

Lemma 11 *Given a strict preference system $\mathcal{I} = (G, \prec)$ with $G = (V, E)$ and a set $S \subseteq E$ of edges such that $\mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}$, the number of stable matchings in $\mathcal{I}$ is at most $2^{|S|}$, and it is possible to enumerate all of them in time $2^{|S|} \cdot O(|E|)$.*

Corollary 12 *In a strict preference system $\mathcal{I}$, the number of stable matchings is at most $2^{\Delta^{\mathrm{swap}}(\mathcal{I})}$.*

Observe that although the number of stable matchings may grow exponentially as a function of the distance Δ^{edge} or Δ^{swap}, this growth does not depend on the size of the instance. By contrast, this is not the case for the distance Δ^{vert}.

Lemma 13 *Given a strict preference system $\mathcal{I} = (G, \prec)$ with $G = (V, E)$ and a set $S \subseteq V$ of vertices such that $\mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}$, the number of stable matchings in $\mathcal{I}$ is at most $|V|^{|S|}$, and it is possible to enumerate all stable matchings of $\mathcal{I}$ in time $|V|^{|S|} \cdot O(|E|)$.*

There exists an algorithm by Gusfield and Irving [19, 20] that outputs the set $\mathcal{S}(\mathcal{I})$ of stable matchings in a preference system $\mathcal{I}$ over a graph $G = (V, E)$ in $O(|\mathcal{S}(\mathcal{I})| \cdot |E|)$ time after $O(|V| \cdot |E| \log |V|)$ preprocessing time. As a consequence, the bounds of Lemma 11, Corollary 12, and Lemma 13 on $|\mathcal{S}(\mathcal{I})|$ directly yield a way to handle computationally hard problems on any preference system $\mathcal{I}$ where $\Delta^{\mathrm{swap}}(\mathcal{I})$, $\Delta^{\mathrm{edge}}(\mathcal{I})$, or $\Delta^{\mathrm{vert}}(\mathcal{I})$ has small value, even without the need to determine a set S of edges or vertices for which $\mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}$ or a set S of swaps for which $I \lhd S \in \mathcal{F}_{\mathrm{ML}}$. Thus, we immediately have the following result, even without having to use our results in Section 4. For the definitions of the NP-hard problems mentioned as an example in Theorem 14, see the book [28].

Theorem 14. *Let $\mathcal{I}$ be a strict (but not necessarily bipartite) preference system, and Q any optimization problem where the task is to maximize or minimize some function f over $\mathcal{S}(\mathcal{I})$ such that $f(M)$ can be computed in polynomial time for any matching $M \in \mathcal{S}(\mathcal{I})$. Then Q can be solved*
(i) in FPT time with parameter $\Delta^{\mathrm{edge}}(\mathcal{I})$ or $\Delta^{\mathrm{swap}}(\mathcal{I})$;
(ii) in polynomial time if $\Delta^{\mathrm{vert}}(\mathcal{I})$ is constant.
In particular, these results hold for Sex-Equal Stable Matching, Balanced Stable Matching, (Generalized) Median Stable Matching[4], Egalitarian Stable Roommates, *and* Maximum-Weight Stable Roommates.

We remark that the bounds in Lemmas 11 and 13 are tight in the following sense. For any $k, n \in \mathbb{N}$ with $n \geq k$, there exist strict preference systems $\mathcal{I}_k$ and $\mathcal{J}_{k,n}$ such that (i) $\Delta^{\mathrm{edge}}(\mathcal{I}_k) = k$ and $\mathcal{I}_k$ admits 2^k stable matchings, and (ii) $\Delta^{\mathrm{vert}}(\mathcal{J}_{k,n}) = k$, the number of vertices in $\mathcal{J}_{k,n}$ is $2n$, and $\mathcal{J}_k$ admits $\binom{n}{k}$ stable matchings. See the full paper [31] for the details of these constructions.

[4] Although the problem of finding a (generalized) median matching is not an optimization problem over $\mathcal{S}(\mathcal{I})$, it is clear that it can be solved in $|\mathcal{S}(\mathcal{I})| \cdot O(|\mathcal{I}|)$ time.

5.2 Maximum-Utility Popular Matchings with Instability Costs

We now turn our attention to the Max-Utility Popular Matching with Instability Costs problem, studied in [32]: given a strict preference system $\mathcal{I} = (G, \prec)$, a utility function $\omega : E(G) \to \mathbb{N}$, a cost function $c : E(G) \to \mathbb{N}$, an objective value $t \in \mathbb{N}$ and a budget $\beta \in \mathbb{N}$, the task is to find a popular matching in $\mathcal{I}$ whose utility is at least t and whose blocking edges have total cost at most β. Our aim is to investigate whether we can solve this problem efficiently for instances that are close to admitting a master list.

Note that in general this problem is computationally hard even if the given preference system is strict, bipartite, admits a master list, and the cost and utility functions are very simple. Namely, given a strict, bipartite preference system $(G, \prec) \in \mathcal{F}_{\mathrm{ML}}$ for which a stable matching has size $|V(G)|/2 - 1$, it is NP-hard and W[1]-hard with parameter β to find a complete popular matching (i.e., one that is larger than a stable matching) that admits at most β blocking edges [32]. Nevertheless, if the total cost β of the blocking edges that we allow is a constant and each edge has cost at least 1, then Max-Utility Popular Matching with Instability Costs can be solved in polynomial time for bipartite, strict preference systems that admit a master list (in fact, it suffices to assume that the preferences of all vertices on one side of the bipartite input graph are consistent with a master list), representing an island of tractability for this otherwise extremely hard problem [32]. Therefore, it is natural to ask whether we can extend this result for strict preferences systems that are close to admitting a master list. Theorem 15 answers this question affirmatively.

Theorem 15. *Let $\mathcal{I}$ be a strict (but not necessarily bipartite) preference system with $G = (V, E)$. Then an instance $(\mathcal{I}, \omega, c, t, \beta)$ of* Max-Utility Popular Matching with Instability Costs *where $c(e) \geq 1$ for all edges $e \in E$, and β is constant can be solved*

(i) in FPT time with parameter $\Delta^{\mathrm{edge}}(\mathcal{I})$ or $\Delta^{\mathrm{swap}}(\mathcal{I})$;
(ii) in polynomial time if $\Delta^{\mathrm{vert}}(\mathcal{I})$ is constant.

We apply the same approach as in Section 5.1, with a crucial difference: for the algorithms proving Theorem 15 we will need to determine a set of edges or vertices whose deletion yields an instance in $\mathcal{F}_{\mathrm{ML}}$. Using such a set, we then apply Lemma 16 or 17 below; these are generalizations of Lemmas 11 and 13 for the case when we allow a fixed set of edges to block the desired matching.

Lemma 16 *Given a strict preference system $\mathcal{I} = (G, \prec)$ with $G = (V, E)$ and edge sets $B \subseteq E$ and $S \subseteq E$ such that $\mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}$, the number of matchings M for which $B = \mathrm{bp}(M)$ is at most $2^{|S|}$, and it is possible to enumerate them in time $2^{|S|} \cdot O(|E|)$.*

Lemma 17 *Given a strict preference system $\mathcal{I} = (G, \prec)$ with $G = (V, E)$, an edge set $B \subseteq E$, and a vertex set $S \subseteq V$ such that $\mathcal{I} - S \in \mathcal{F}_{\mathrm{ML}}$, the number of matchings M for which $B = \mathrm{bp}(M)$ is at most $|V|^{|S|}$, and it is possible to enumerate them in time $|V|^{|S|} \cdot O(|E|)$.*

Table 1. Summary of our results on MLS, MLED, and MLVD. Results marked by the sign † assume the Unique Games Conjecture.

problem	parameterized complexity	approximation
MLS	FPT wrt k (Cor. 5, Thm. 9)	constant-factor approx. is NP-hard (Thm. 4)†
MLED	W[1]-hard wrt k (Thm. 6)	constant-factor approx. is NP-hard (Thm. 4)† 2-approx. FPT alg wrt k (Thms. 7, 10)
MLVD	W[2]-hard wrt k (Thm. 8)	$f(k)$-approx. is W[1]-hard wrt k (Thm. 8)

The algorithms proving Theorem 15 start with searching for the set of blocking edges using brute force: recall that our budget β is constant, and since each edge has cost at least 1, we know that $|\text{bp}(M)| \leq \beta$ for our desired popular matching M. Thus, there are only polynomially many sets to consider as the set B of blocking edges.

Next, to prove statement (ii) of Theorem 15, we again use brute force to find a set S of $\Delta^{\text{vert}}(\mathcal{I})$ vertices such that $\mathcal{I} - S \in \mathcal{F}_{\text{ML}}$. Thus having the sets S and B at hand, we can apply Lemma 17. For statement (i) however, we need to find a set S of edges such that $\mathcal{I} - S \in \mathcal{F}_{\text{ML}}$ in FPT time with $\Delta^{\text{edge}}(\mathcal{I})$ or $\Delta^{\text{swap}}(\mathcal{I})$ as parameter. Notice that it suffices to use Theorem 7 to obtain an edge set S of size at most $2\Delta^{\text{edge}}(\mathcal{I})$, and then we can apply Lemma 16. For a more detailed description of these algorithms and their correctness, see the full paper [31].

6 Summary and Further Research

We summarize our main results on MLS, MLED, and MLVD in Table 1. Interestingly, all our hardness results hold for strict preference systems, and we were able to extend all our positive results for preference systems with weak orders.

There are a few questions left open in the paper. We gave asymptotically tight bounds on the maximum number of stable matchings in a strict preference system $\mathcal{I}$ as a function of $\Delta^{\text{edge}}(\mathcal{I})$ and $\Delta^{\text{vert}}(\mathcal{I})$, but we were not able to do the same for $\Delta^{\text{swap}}(\mathcal{I})$. Another question is whether the approximation factor of our 2-approximation FPT algorithm for MLED can be improved.

A possible direction of future research would be to identify further problems that can be solved efficiently on preference systems that are close to admitting a master list. Also, it would be interesting to see how these measures vary in different real-world scenarios, and to find those practical applications where preference profiles are usually close to admitting a master list.

References

1. Abraham, D.J., Levavi, A., Manlove, D.F., O'Malley, G.: The stable roommates problem with globally-ranked pairs. Internet Math. **5**, 493–515 (2008)
2. Bang-Jensen, J., Gutin, G.Z.: Digraphs: Theory, Algorithms and Applications. Springer Monographs in Mathematics, Springer, 2nd edn. (2009)

3. Betzler, N., Fellows, M.R., Guo, J., Niedermeier, R., Rosamond, F.A.: Fixed-parameter algorithms for Kemeny rankings. Theor. Comput. Sci. **410**(45), 4554–4570 (2009)
4. Bhatnagar, N., Greenberg, S., Randall, D.: Sampling stable marriages: Why spouse-swapping won't work. In: SODA '08. pp. 1223–1232. SIAM (2008)
5. Boehmer, N., Faliszewski, P., Niedermeier, R., Szufa, S., Wąs, T.: Understanding distance measures among elections. In: IJCAI-22. pp. 102–108 (2022)
6. Bredereck, R., Chen, J., Knop, D., Luo, J., Niedermeier, R.: Adapting stable matchings to evolving preferences. In: AAAI '20. pp. 1830–1837 (2020)
7. Bredereck, R., Heeger, K., Knop, D., Niedermeier, R.: Multidimensional stable roommates with master list. In: WINE '20. pp. 59–73. Springer (2020)
8. Chen, J., Skowron, P., Sorge, M.: Matchings under preferences: Strength of stability and tradeoffs. ACM Trans. Econ. Comput. **9**(4), 20:1–20:55 (2021)
9. Cheng, C.T., Rosenbaum, W.: Stable matchings with restricted preferences: Structure and complexity. In: EC '21. pp. 319–339 (2021)
10. Chitnis, R., Cygan, M., Hajiaghayi, M., Marx, D.: Directed subset feedback vertex set is fixed-parameter tractable. ACM Trans. Algorithms **11**(4), 28:1–28:28 (2015)
11. Chowdhury, R.: A simple matching domain with indifferences and a master list. Rev. Econ. Des. (2022)
12. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms. Springer, Cham (2015)
13. Elkind, E., Lackner, M., Peters, D.: Structured preferences. In: Endriss, U. (ed.) Trends in computational social choice, chap. 10, pp. 187–207. AI Access (2017)
14. Fellows, M.R., Hermelin, D., Rosamond, F.A., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. Theor. Comput. Sci. **410**, 53–61 (2009)
15. Fitzsimmons, Z., Hemaspaandra, E.: Kemeny consensus complexity. In: IJCAI-21. pp. 196–202 (2021)
16. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Amer. Math. Monthly **69**(1), 9–15 (1962)
17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
18. Gupta, S., Jain, P., Saurabh, S.: Well-structured committees. In: IJCAI-20. pp. 189–195 (2020)
19. Gusfield, D.: The structure of the stable roommate problem: Efficient representation and enumeration of all stable assignments. SIAM J. Comput. **17**(4), 742–769 (1988)
20. Gusfield, D., Irving, R.W.: The stable marriage problem: structure and algorithms. Foundations of Computing Series, MIT Press, Cambridge, MA (1989)
21. Irving, R.W., Manlove, D.F., Scott, S.: The stable marriage problem with master preference lists. Discret. Appl. Math. **156**(15), 2959–2977 (2008)
22. Kamiyama, N.: Many-to-many stable matchings with ties, master preference lists, and matroid constraints. In: AAMAS '19. pp. 583–591 (2019)
23. Kemeny, J.: Mathematics without numbers. Daedalus **88**, 577–591 (1959)
24. Khanchandani, P., Wattenhofer, R.: Distributed stable matching with similar preference lists. In: OPODIS 2016. LIPIcs, vol. 70, pp. 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
25. Khot, S.: On the power of unique 2-prover 1-round games. In: STOC 2002. pp. 767–775. Association for Computing Machinery (2002)
26. Lebedev, D., Mathieu, F., Viennot, L., Gai, A.T., Reynier, J., de Montgolfier, F.: On using matching theory to understand P2P network design. In: INOC '07 (2007)

27. Lokshtanov, D., Ramanujan, M.S., Saurabh, S.: When recursion is better than iteration: A linear-time algorithm for acyclicity with few error vertices. In: SODA '18. pp. 1916–1933 (2018)
28. Manlove, D.F.: Algorithmics of matching under preferences, Series on Theoretical Computer Science, vol. 2. World Scientific, Singapore (2013)
29. Meeks, K., Rastegari, B.: Solving hard stable matching problems involving groups of similar agents. Theor. Comput. Sci. **844**, 171–194 (2020)
30. Perach, N., Polak, J., Rothblum, U.G.: A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion. Int. J. Game Theory **36**, 519–535 (2008)
31. Schlotter, I.: Recognizing when a preference system is close to admitting a master list. CoRR **abs/2212.03521** (2022), arXiv:2212.03521 [cs.CC]
32. Schlotter, I., Cseh, Á.: Maximum-utility popular matchings with bounded instability. CoRR **abs/2205.02189** (2022), arXiv:2205.02189 [cs.DM]
33. Vazirani, V.V.: Approximation algorithms. Springer (2001)