Maximum-utility popular matchings with bounded instability

Ildikó Schlotter[1] and Ágnes Cseh[2]

[1] HUN-REN Centre for Economic and Regional Studies, Hungary,
schlotter.ildiko@krtk.hun-ren.hu
[2] University of Bayreuth, Germany, agnes.cseh@uni-bayreuth.de

Abstract

In a graph where vertices have preferences over their neighbors, a matching is called popular if it does not lose a head-to-head election against any other matching when the vertices vote between the matchings. Popular matchings can be seen as an intermediate category between stable matchings and maximum-size matchings. In this paper, we aim to maximize the utility of a matching that is popular but admits only a few blocking edges.

We observe that for general graphs finding a popular matching with at most one blocking edge is already NP-complete. For bipartite instances, we study the problem of finding a maximum-utility popular matching with a bound on the number (or more generally, the cost) of blocking edges applying a multivariate approach. We show classical and parameterized hardness results for severely restricted instances. By contrast, we design an algorithm for instances where preferences on one side admit a master list, and show that this algorithm is roughly optimal.

1 Introduction

In the classic stable matching problem, we are given a bipartite graph, where the two sets of vertices represent two agent sets. Each agent has a strictly ordered preference list over their possible partners from the other agent set. A matching is *stable* if it is not *blocked* by any edge, that is, no pair of agents exists who are mutually inclined to abandon their partners for each other. The existence of stable matchings was shown in the seminal paper of Gale and Shapley [31]. The optimality notion was later extended to various other input settings in order to suit the growing number of applications such as employer allocation markets [56], university admission decisions [4,9], campus housing assignments [15,55] and bandwidth allocation [30].

In the area of matchings under preferences, the trade-off between stability and size (or utility) has been an actively investigated topic [2, 8, 13, 29, 34]. The most extensively studied compromise between these optimality criteria might be the notion of popular matchings, first defined by Gärdenfors [32]. Matching M is more popular than another matching M' if the number of vertices preferring M to M' is larger than the number of vertices preferring M' to M. A matching M is called *popular* if there is no matching M' that is more popular than M.

In bipartite graphs, all stable matchings have the same size, at least $\frac{1}{2}|M_{\max}|$ where $M_{\max}$ is a maximum-size matching. Stable matchings are minimum-size popular matchings [32, 38], and maximum-size popular matchings have size at least $\frac{2}{3}|M_{\max}|$ [44]. In non-bipartite graphs it is NP-complete to decide whether a popular matching even exists [24, 35]. If edges have utilities in $\{1, 2\}$, computing a popular matching of maximum utility is NP-hard even in bipartite instances [24]. The difference between a maximum-utility stable and a maximum-utility popular matching can be arbitrarily large in terms of the objective.

In this paper, we concentrate on the stability versus maximum utility question within the set of popular matchings. We investigate *popular matchings with bounded instability*, that is, matchings that are popular but admit only a few blocking edges. Can we find popular matchings with, say, at most one blocking edge efficiently? If the given instance admits a stable matching (which is always the case for bipartite graphs), the answer is obviously yes, since all stable matchings are popular. But can we efficiently find a maximum-utility matching among all popular matchings with at most one blocking edge? How far can we venture into the realm of instability, if we want to keep our ability to find maximum-utility matchings efficiently?

In order to examine such questions in detail, we use a general model where with each edge we associate a *utility* and a *cost* as well, and we ask for a popular matching whose total utility is above a given threshold, while its blocking edges have total cost not exceeding a given budget. This setting can be interpreted as follows. Vertices are agents whose partnership brings as much profit to a central authority as the utility of the edge connecting them. The cost of an edge is the regret of the agents if the edge blocks the matching. The central authority disposes of limited resources to compensate agents who could be better off by switching to blocking edges. Up to this limit, blocking edges can be paid for and thus tolerated. The goal is to find a matching whose utility reaches our target, while ensuring that agents can be compensated from the cost budget.

Since the question is computationally intractable in such a general form as we will see in Section 2, we apply the framework of parameterized complexity [20], and take a multivariate approach [53] in order to understand how exactly the several parameters appearing in such an instance contribute to its intractability, and to identify cases that can be solved efficiently. Apart from the several natural parameters we can associate with the problem (such as our objective value or our budget), we also investigate various restrictions on the preference profiles and how they influence the computational complexity of the question.

1.1   Related work

We briefly review some classical and parameterized complexity results on stable and popular matchings. Then we elaborate on the versions of these problems with edge utilities, and finally, we discuss the relaxation of stability.

Complexity results for stable and popular matchings.   From the seminal paper of Gale and Shapley [31] we know that in bipartite instances a stable matching always exists and can be found in linear time. If the graph is not bipartite, then the existence of a stable solution is not guaranteed. However, Irving's linear-time algorithm finds a stable matching or reports that none exists [40].

For bipartite instances, it was already noticed by Gärdenfors [32] that all stable matchings are popular, which implies that in bipartite instances popular matchings always exist. In fact stable matchings are smallest-size popular matchings, as shown by Biró et al. [6]; maximum-size popular matchings can be found in polynomial time as well [38, 44]. Interestingly, finding a popular matching that has exactly some prescribed size, or in fact, one that has neither minimum nor maximum size, is NP-hard [24]. Only recently Faenza et al. [24] and Gupta et al. [35] resolved the long-standing open question on the complexity of deciding whether a popular matching exists in a non-bipartite instance and showed that the problem is NP-complete.

Stable matchings have been studied extensively from a parameterized viewpoint. For an overview, please consult the survey by Chen [12]. Only a few results consider popularity within the parameterized framework, and most of them do so in the context of the house allocation problem [17, 45].

There is a large set of results on stable and popular matchings in instances where preferences

admit a master list (see Section 2 for a definition) [11,43,46,51]. Master lists naturally occur in a number of applications, such as P2P networks [30,49], job markets [42], and student housing assignments [55].

Stable matchings with edge utilities. For bipartite instances, Irving et al. [41] proposed an algorithm for finding a maximum-utility stable matching in $O(n^4 \log n)$ time for an n-vertex graph; see also [36]. For non-negative integer edge utilities that satisfy a certain monotonicity requirement, the fastest known algorithm is due to Feder [25,26], running in $O(n^2 \log(\frac{K}{n^2} + 2)) \cdot \min\{n, \sqrt{K}\}$ time where K is the utility of an optimal solution. A maximum-utility stable matching can also be computed using a simple and elegant formulation of the stable matching polytope [57]. For the non-bipartite case, finding a maximum-utility stable matching is NP-hard, but 2-approximable under certain monotonicity constraints using LP methods [59,60].

Popular matchings with edge utilities. For bipartite instances, Faenza et al. [23] showed that it is NP-complete to decide if there exists a popular matching that contains two given edges, which is a very restricted case of popular matchings with edge utilities. The same authors provided a 2-approximation algorithm for non-negative edge utilities. NP-hardness was established for non-bipartite instances with edge utilities a couple of years earlier already [39].

Almost stable matchings. Non-bipartite stable matching instances need not admit a stable solution. The number of blocking edges is a characteristic property of every matching. A natural goal is to find a matching that minimizes the number of blocking edges—such a matching is called *almost stable*. This approach has a broad literature: almost stable matchings have been investigated in bipartite [8,34,37,47] and non-bipartite instances [1,7,13,18]. Closest to our work is the paper by Gupta et al. [34] studying the trade-off between size and stability from a parameterized complexity viewpoint.

1.2 Our results and structure of the paper

Using a multivariate approach, we gain insight into the computational complexity of finding a maximum-utility popular matching respecting a bound on the cost of its blocking edges. We draw a detailed map of the problem's complexity in terms of parameters such as the cost budget k, the desired utility value t, the form of the cost and utility functions (e.g., being binary or uniform), and the structural properties of the preference profile; see Table 1 for a detailed summary.

Sections 2 and 3. We first define our model, optimality notions, and problems. In Observation 1 we prove that for general graphs, already finding *any* popular matching with at most one blocking edge is NP-complete. We thus restrict ourselves to the bipartite case and show in Theorems 6 and 7 that finding a maximum-utility popular matching with at most one blocking edge is NP-complete even for highly restricted inputs.

Section 4. To contrast these strong intractability results, we next focus on the "tractability island" of bipartite instances that admit a master list on one side. We propose a simple algorithm that finds a maximum-utility popular matching whose blocking edges have total cost at most k, given a positive integer cost function on the edges. Our algorithm runs in time $O(|E|^k)$ where E is the edge set of the input graph (Theorem 10). This running time is tight in the sense that the problem is W[1]-hard with parameter k (Theorem 11). We show our algorithm's optimality also in the sense that the few assumptions we have on the input (besides admitting a master list), namely that preferences are strict and edges have positive integer costs, are necessary: allowing for ties or for zero-cost edges undermines the tractability of the problem (Theorems 12 and 13).

3

Section 5. We close our investigations with relaxing the requirement of popularity, and focusing on the less restrictive requirement of Pareto-optimality instead. Hence, we ask the following question: can we efficiently find a maximum-utility Pareto-optimal matching with only a few blocking edges or, more generally, one whose blocking edges have total cost not exceeding a given budget? In Theorem 14 we propose an algorithm for this problem that runs in $2^{O(k \log k)}|E|^{k+2}$ time for a positive integer cost function on the edge set E and a budget k. We also prove that this algorithm is essentially optimal (Corollary 16).

2 Preliminaries

We first introduce our model and the most important concepts in Section 2.1, and then state our problem definitions and start our investigations in Section 2.2.

2.1 Model and notation

Graphs. For a graph $G = (V, E)$, we let $V(G)$ and $E(G)$ denote its vertex and edge set, respectively. If $F \subseteq E$, then $V(F)$ is the set of all endpoints in F. All our graphs are simple (without loops or parallel edges). For a vertex $v \in V$, $N_G(v)$ denotes the set of its *neighbors* and $\delta_G(v) = |N_G(v)|$ its *degree* in G. The maximum degree of G is $\Delta_G = \max_{v \in V(G)} \delta_G(v)$. Two edges are *adjacent*, if they share an endpoint. A *matching* in G is a set of edges such that no two of them are adjacent. For a matching M and an edge $(a, b) \in M$, we let $M(a) = b$, and conversely, $M(b) = a$. For a set X of edges or vertices in G, let $G - X$ be the subgraph of G obtained by deleting X from G; for a singleton $X = \{x\}$ we may simply write $G - x$. We also let $G[X] = G - (V(G) \setminus X)$ for some $X \subseteq V(G)$. For a set F of edges, we let $V(F)$ denote the set of all vertices that are incident to some edge in F. We use $\triangle$ for the symmetric difference operator.

Preference systems. A *preference system* is a pair $(G, \preceq)$ where $G = (V, E)$ is a graph and $\preceq$ is a collection of preference orders $\preceq_v$ for each $v \in V$, where $\preceq_v$ can be a strict or a weak linear order over $N_G(v)$. We let $\prec_v$ be the strict part of $\preceq_v$, i.e., $a \prec_v b$ means $b \npreceq_v a$. We say that v *prefers* a to b if $b \prec_v a$, and v *weakly prefers* a to b if $b \preceq_v a$. Mostly we will deal with *strict preference systems* where $\preceq_v$ is a strict linear order for each $v \in V$; in this case we write $(G, \prec)$. We say that $(G, \preceq)$ is *complete*, if G is a complete graph or, in case we assume a bipartite setting, if G is a complete bipartite graph. We will say that $(G, \preceq)$ is *compatible* with a complete preference system $(G', \preceq')$ if G is a subgraph of G', and for any $v \in V(G)$, the restriction of $\preceq'_v$ to $N_G(v)$ is exactly $\preceq_v$. For a set X of edges or vertices, we define the deletion of X from $(G, \preceq)$ as $(G - X, \preceq^{G-X})$ where $\preceq^{G-X}$ is the restriction of $\preceq$ to $G - X$, containing for each $v \in V(G - X)$ a preference order $\preceq_v^{G-X}$ over $N_{G-X}(v)$.

Stability, popularity, and Pareto-optimality. Given a preference system $(G, \preceq)$ and a matching M in G, some $(a, b) \in E$ is a *blocking edge for* M if a is unmatched or prefers b to $M(a)$, and b is unmatched or prefers a to $M(b)$; we denote by $\mathrm{bp}_G(M)$ the set of blocking edges for M in G. If G is clear from the context, we may omit the subscript (also from notations $\delta_G(\cdot)$ or $N_G(\cdot)$). We say that M is *stable* in G if $\mathrm{bp}_G(M) = \emptyset$.[1]

For two matchings M and M' in G, some vertex v *prefers* M over M', if either v is matched in M but unmatched in M', or $M'(v) \prec_v M(v)$. We say that M is *more popular* than M', if more vertices prefer M to M' than vice versa. A matching M is *popular*, if no matching is more popular than M.

---

[1]When $(G, \preceq)$ is not strict, stability as we define it is often called *weak stability*. See the book [50] for other stability notions for weakly ordered preferences.

A *Pareto-improvement* of a matching M is a matching M' such that no vertex prefers M to M', and at least one prefers M' to M. A matching M is *Pareto-optimal*, if there is no Pareto-improvement for it. Although stability, popularity, and Pareto-optimality are defined in the context of a preference system $(G, \preceq)$, when $\preceq$ is clear from the context, we may simply say that a matching is stable, popular, or Pareto-optimal in G. Notice that for strict preference systems, stable matchings are popular, and popular matchings are Pareto-optimal.

Structured preferences. In a bipartite preference system $(G = (A, B; E), \preceq)$, a *master list* over vertices of A is defined as an ordering $\mathcal{L}_A$ of all vertices in A such that for any $b \in B$, restricting $\mathcal{L}_A$ to $N_G(b)$ yields exactly the ordering $\preceq_b$. We say that $(G, \preceq)$ *admits a master list on one side*, if there exists a master list over either A or B; if both holds, then $(G, \preceq)$ *admits a master list on both sides*.

Single-peaked and single-crossing preferences originate from problems in the context of elections, where preferences are complete linear orders. The two notions have been formally defined for certain stable matching problems [5, 10] and their applicability in real-world matching instances has also been argued for [3, 22, 58]. We follow standard definitions from the social choice literature that adapt these notions to incomplete preferences [10, 21, 28]. For simplicity, let us assume that $(G, \prec)$ is a strict preference system. Then $(G, \prec)$ has *single-peaked* preferences, if there exists a strict linear ordering $\rhd$ of all vertices in $V(G)$ called an *axis* such that for any vertex $v \in V(G)$ and for every $a, b, c \in N_G(v)$ with $a \rhd b \rhd c$, the relation $b \prec_v a$ implies $c \prec_v b$. In this case we also say that $(G, \prec)$ is *single-peaked with respect to the axis* $\rhd$. If $G = (A, B; E)$ is bipartite, then it suffices to provide a suitable axis for A and for B separately.

To define single-crossing preferences, let us first assume that $(G, \prec)$ is a strict and complete preference system. For any $a, b \in V(G)$ let $V^{a \prec b} = \{v \in V(G) : a \prec_v b\}$ denote the set of vertices preferring b to a. We say that $(G, \prec)$ is *single-crossing with respect to a strict linear ordering* $\rhd$ of $V(G)$, if for any $a, b \in V(G)$ either all vertices in $V^{a \prec b}$ precede all vertices in $V^{b \prec a}$ according to $\rhd$, or just the opposite, all vertices in $V^{b \prec a}$ precede all vertices in $V^{a \prec b}$ according to $\rhd$. We say that $(G, \prec)$ is *single-crossing*, if it is single-crossing with respect to some strict linear ordering of $V(G)$. An incomplete strict preference system is *single-crossing*, if it is compatible with a complete single-crossing preference system. Note that if G is bipartite, then it suffices to provide a complete bipartite preference system compatible with $(G, \prec)$ and separate linear orders for A and for B.

Example 1. Consider the complete bipartite preference system $(G, \prec)$ over $G = (A, B; E)$ where $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, b_2, b_3\}$, with the preferences of the vertices given as follows.

$$
\begin{array}{llll}
a_1: & b_1, b_2, b_3; & b_1: & a_1, a_2, a_3, a_4; \\
a_2: & b_1, b_3, b_2; & b_2: & a_2, a_1, a_4, a_3; \\
a_3: & b_2, b_3, b_1; & b_3: & a_2, a_4, a_1, a_3; \\
a_4: & b_2, b_1, b_3. &
\end{array}
$$

It is straightforward to see that $(G, \prec)$ is single-crossing: as Figure 1 shows, the preferences are single-crossing with respect to any ordering $\rhd$ over $A \cup B$ that satisfies $a_2 \rhd a_1 \rhd a_4 \rhd a_3$ and $b_1 \rhd b_2 \rhd b_3$.

By contrast, consider the modified preference system $(G, \prec')$ obtained by changing the preferences of a_4 so that $b_3 \prec'_{a_4} b_2 \prec'_{a_4} b_1$; the preferences of all other vertices remain unchanged. We claim that $(G, \prec')$ is not single-crossing. To see this, first note that a_1 and a_3 both prefer b_2 to b_3, while a_2 and a_4 prefer b_3 to b_2 according to $\prec'$. Assuming that $(G, \prec')$ is single-crossing with respect to some strict linear ordering $\rhd$ of $A \cup B$, this means that either both a_1 and a_3 precede each of a_2 and a_4 in $\rhd$, or vice versa. However, notice that a_1 and a_3 disagree about which

$$a_2: b_1, b_3, b_2; \qquad b_1: a_1, a_2, a_3, a_4;$$
$$\triangledown \qquad\qquad \triangledown$$
$$a_1: b_1, b_2, b_3; \qquad b_2: a_2, a_1, a_4, a_3;$$
$$\triangledown \qquad\qquad \triangledown$$
$$a_4: b_2, b_1, b_3; \qquad b_3: a_2, a_4, a_1, a_3;$$
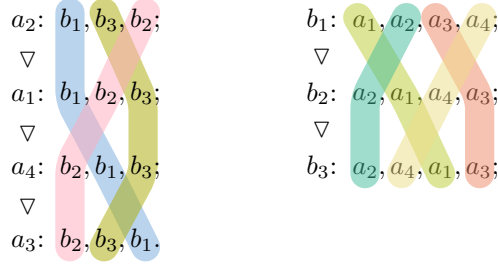$$\triangledown$$
$$a_3: b_2, b_3, b_1.$$

Figure 1: Illustration showing the single-crossing property of the complete bipartite preference system defined in Example 1.

of b_1 and b_2 is better, and additionally, a_2 and a_4 also disagree about the ranking of b_1 and b_2. Hence, it is not possible to determine a "threshold-vertex" t_{b_1, b_2} for the ordering $\triangleright$ so that all vertices preceding t_{b_1, b_2} in $\triangleright$ rank b_1 and b_2 in the same way, and all vertices following t_{b_1, b_2} in $\triangleright$ rank b_1 and b_2 in the opposite way, contradicting our assumption that $(G, \prec')$ is single-crossing with respect to $\triangleright$.

Classical and parameterized complexity. We assume that the reader is familiar with basic notions and techniques of classical and parameterized complexity theory; for an introduction and definitions we refer to the books [19, 33, 52].

2.2 Problem definitions and initial results

Let us now formally define the problem whose computational complexity is the main focus of our paper.

MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS:

Input: A strict preference system $(G, \prec)$, 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 $k \in \mathbb{N}$.

Question: Is there a popular matching in G whose utility is at least t and whose blocking edges have total cost at most k?

For a set $F \subseteq E$ of edges in G, let $\omega(F) = \sum_{e \in F} \omega(e)$ and $c(F) = \sum_{e \in F} c(e)$ be its utility and cost, respectively. A matching M in G is *feasible*, if both $\omega(M) \geq t$ and $c(\mathrm{bp}_G(M)) \leq k$ hold.

A very natural special case of the above problem is when we simply limit the number of blocking edges: this amounts to setting all edge costs to 1. Hence, we are looking for a popular matching (in the hope of finding matchings with greater utility or size when compared to stable matchings) while also setting an upper bound on the degree of instability of the matching.

POPULARITY WITH BOUNDED INSTABILITY:

Input: A strict preference system $(G, \prec)$ and an integer k.

Question: Is there a popular matching M in G with $|\mathrm{bp}_G(M)| \leq k$?

Our first result shows that this problem is NP-complete even for $k = 1$.

Observation 1. *The* POPULARITY WITH BOUNDED INSTABILITY *problem is* NP*-complete for* $k = 1$.

Proof. See the reduction in [16, Section 5.3] which provides a simple proof that deciding the existence of a popular matching in a (non-bipartite) strict preference system is NP-hard. There is no stable matching in the constructed instance, and if there exists a popular matching M then there is a unique blocking edge (r, r'') to M. □

Corollary 2. *The* MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS *problem is* NP-*complete for any fixed utility function* $\omega : E \to \mathbb{N}$, *even if the objective value is* $t = 0$, *the cost function is* $c \equiv 1$, *and our budget is* $k = 1$.

Motivated by this strong intractability, in the remainder we focus on the case where the graph is bipartite. In this case, interestingly, the MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS problem has a strong connection to the problem of finding a popular matching with a fixed set of blocking edges.

POPULARITY WITH FIXED INSTABILITY:
Input: A strict preference system $(G, \prec)$ and a subset $S \subseteq E(G)$ of edges.
Question: Is there a popular matching M in G such that $\mathrm{bp}_G(M) = S$?

On the one hand, a natural idea for finding maximum-utility matchings with bounded instability is to "guess" the set S of blocking edges, and search for a maximum-utility matching only among matchings M for which $\mathrm{bp}(M) = S$, using some structural insight. On the other hand, given an instance of POPULARITY WITH FIXED INSTABILITY where our aim is to ensure that each edge in S is a blocking edge, a possible approach is to define a utility function that enforces certain edges in the neighborhood of the vertices covered by S to be included in any feasible matching M, thus yielding $\mathrm{bp}(M) = S$. This intuitive two-way connection between the two problems can be observed in the details of our reductions in Section 3, and will also form the basis of our positive results in Section 4.

3   Hardness results

The main result of this section, Theorem 6, shows that both MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS and POPULARITY WITH FIXED INSTABILITY are computationally intractable even if the input graph G is bipartite, $\Delta_G = 3$, and preferences are single-peaked and single-crossing.

The path to these results leads through Observation 4, which characterizes popular matchings with only a single blocking edge, and Theorem 5, which is a weaker version of Theorem 6. Finally, Theorem 7 shows that MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS remains hard even if the utility function is $\omega \equiv 1$.

To depict an overview of the main reduction, we provide an additional proof outline to Theorem 5. The proofs of Theorems 6 and 7 are modified versions of the proof of Theorem 5.

3.1   Popular matchings blocked by a single edge

To characterize popular matchings with only a single blocking edge, we need some further notation. Given a matching M in a graph G, the subgraph G_M of G is obtained by deleting those edges (a, b) outside M from G where both a and b prefer their partner in M to each other.

The following well-known characterization of popular matchings is due to Huang and Kavitha.

Theorem 3 ([38]). *A matching M is popular in G if and only if G_M does not contain any of the following:*

(i) *an M-alternating cycle containing a blocking edge;*
(ii) *an M-alternating path containing two distinct blocking edges;*
(iii) *an M-alternating path containing a blocking edge and an unmatched vertex as an endpoint.*

The correctness of our reductions in this section heavily relies on Observation 4, which is a re-formulation of the above characterization by Huang and Kavitha for the case of popular matchings with a single blocking edge.

Observation 4. *Given a strict preference system* $(G, \prec)$ *where* $G = (A, B; E)$ *is bipartite, a matching* M *in* G *is a popular matching with exactly one blocking edge* $e = (u, v) \in E$ *if and only if the following conditions hold:*

(c1) *e blocks M in G;*

(c2) *M is a stable matching in $G - e$;*

(c3) *there exists no M-alternating path in $(G - e)_M$*

    (c3/i) *from a vertex unmatched by M to u or to v, having even length, or*

    (c3/ii) *from u to v, starting and ending with an edge of M.*

3.2 A weaker form of Theorem 6

We start with stating a weaker version of Theorem 6, whose proof provides a reduction that is the basis of all our reductions in this section.

Theorem 5. *The* Max-Utility Popular Matching with Instability Costs *problem is* NP-*complete even if*

- *the input graph G is bipartite,*
- *the cost function is $c \equiv 1$,*
- *the budget is $k = 1$,*
- *the utility function is* $\omega(e) = \begin{cases} 1 & \text{if } e = f^\star, \\ 0 & \text{otherwise,} \end{cases}$ *for some $f^\star \in E(G)$, and*
- *the objective value is $t = 1$.*

The Popularity with Fixed Instability *problem is* NP-*complete even if*

- *the input graph is bipartite, and*
- *$|S| = 1$ for the set S of blocking edges.*

Before the full proof, we sketch a proof outline to this theorem.

Proof outline for Theorem 5. Given an instance φ of Exact-3-SAT, we construct a bipartite graph G with only a single edge $f^\star$ of non-zero utility, not contained in any stable matching, so that achieving the target utility requires the inclusion of $f^\star$ in the matching. However, including $f^\star$ in a matching will turn another edge $e^\star$, adjacent to $f^\star$, into a blocking edge, thus introducing instability. In view of Observation 4, the computational hardness of finding a matching that contains $f^\star$ and is blocked only by $e^\star$ lies in ensuring conditions (c3/i) and (c3/ii) while avoiding the emergence of additional blocking edges.

In our construction, each clause c and each variable x in φ is represented by a vertex a^c and b^x, respectively, left unmatched by any stable matching of G. Moreover, each literal ℓ in some clause c is represented by a cycle $C^{c,\ell}$ in G, and similarly, the possible values true and false of some variable x are represented by cycles $C^{x,\mathtt{t}}$ and $C^{x,\mathtt{f}}$, respectively. Next, for each clause c we create a path that leads from a^c to the blocking edge $e^\star$ and goes through certain edges of the three cycles corresponding to the three literals of c. Similarly, for each variable x we create a path that leads from b^x to $e^\star$ and goes through certain edges of the cycles $C^{x,\mathtt{t}}$ and $C^{x,\mathtt{f}}$.

The preferences of the vertices along these paths and cycles are defined in a way that ensuring condition (c3/i) requires the desired matching M to have the property that $M \triangle M_0$ contains certain cycles, where M_0 is a fixed stable matching (defined as part of the construction). Namely, condition (c3/i) for the unmatched vertex a^c corresponding to some clause c guarantees that $M \triangle M_0$ contains at least one cycle $C^{c,\ell}$, corresponding to a literal of c. Similarly, condition (c3/i) for b^x for some variable x guarantees that either $C^{x,\mathtt{t}}$ or $C^{x,\mathtt{f}}$ is contained in $M \triangle M_0$. The crux of the construction is the addition of so-called consistency edges running between each cycle $C^{c,\ell}$ corresponding to a literal ℓ of some clause c and the cycle $C^{x,\mathtt{t}}$ in case ℓ is the positive literal of variable x, or the cycle $C^{x,\mathtt{f}}$ in case ℓ is the literal $\overline{x}$. The condition that no such consistency edge can block the desired matching M ensures that the truth assignment encoded by M indeed satisfies each clause, and vice versa, any truth assignment that satisfies φ implies a matching M that is popular, contains $f^\star$, and is blocked only by the edge $e^\star$.

Proof of Theorem 5. Since we can test both feasibility and popularity in polynomial time [38], both problems are in NP. We now give a reduction from EXACT-3-SAT, the variant of 3-SAT where each clause contains exactly three literals. Consider an input formula $\varphi = c_1 \wedge c_2 \wedge \cdots \wedge c_m$ over variables $x_1, \ldots, x_n$.

Construction. We construct an instance of MAX-UTILITY POPULAR MATCHING WITH IN-STABILITY COSTS with a graph $G = (A, B; E)$ as follows; see Figure 2 for an illustration. For an integer $i \in \mathbb{N}$, we let $[i] = \{1, 2, \ldots, i\}$.

For each variable x_i, we create a 4-cycle $C^{x_i,\mathtt{t}}$ on vertices $a_0^{x_i,\mathtt{t}}$, $b_0^{x_i,\mathtt{t}}$, $a_1^{x_i,\mathtt{t}}$, and $b_1^{x_i,\mathtt{t}}$, and another 4-cycle $C^{x_i,\mathtt{f}}$ on vertices $a_0^{x_i,\mathtt{f}}$, $b_0^{x_i,\mathtt{f}}$, $a_1^{x_i,\mathtt{f}}$, and $b_1^{x_i,\mathtt{f}}$, representing the `true` and `false` value assignments for variable x_i, correspondingly. Similarly, for each clause c_j, we construct a 4-cycle on $C^{c_j,\ell}$ on vertices $a_0^{c_j,\ell}$, $b_0^{c_j,\ell}$, $a_1^{c_j,\ell}$ and $b_1^{c_j,\ell}$ where $\ell \in [3]$, representing the ℓ-th literal in clause c_j. Furthermore, we add vertices of $\hat{A} = \{a^{c_1}, \ldots, a^{c_m}\}$ corresponding to clauses, as well as vertices of $\hat{B} = \{b^{x_1}, \ldots, b^{x_n}\}$ corresponding to variables. We will also have four special vertices: u, v, u' and v'. To define our vertex set, we let

$$
\begin{aligned}
A^x &= \{a_0^{x_i,\lambda}, a_1^{x_i,\lambda} : i \in [n] \text{ and } \lambda \in \{\mathtt{t}, \mathtt{f}\}\}, \\
A^c &= \{a_0^{c_j,\ell}, a_1^{c_j,\ell} : j \in [m] \text{ and } \ell \in [3]\}, \\
B^x &= \{b_0^{x_i,\lambda}, b_1^{x_i,\lambda} : i \in [n] \text{ and } \lambda \in \{\mathtt{t}, \mathtt{f}\}\}, \\
B^c &= \{b_0^{c_j,\ell}, b_1^{c_j,\ell} : j \in [m] \text{ and } \ell \in [3]\}, \\
A &= A^x \cup A^c \cup \hat{A} \cup \{u, u'\}, \quad \text{and} \\
B &= B^x \cup B^c \cup \hat{B} \cup \{v, v'\}.
\end{aligned}
$$

To define the edge set of our bipartite graph $G = (A, B; E)$, in addition to the edges of the above defined 4-cycles, for each clause c_j we add edges so that G contains the path

$$ P_j = (u, v', a_0^{c_j,1}, b_0^{c_j,1}, a_0^{c_j,2}, b_0^{c_j,2}, a_0^{c_j,3}, b_0^{c_j,3}, a^{c_j}). $$

Similarly, for each variable x_i we add edges so that G contains the path

$$ Q_i = (v, u', b_0^{x_i,\mathtt{t}}, a_0^{x_i,\mathtt{t}}, b_0^{x_i,\mathtt{f}}, a_0^{x_i,\mathtt{f}}, b^{x_i}). $$

Additionally, we define a set F of *consistency edges*: for each clause c_j and each $\ell \in [3]$ we add an edge $(b_1^{c_j,\ell}, a_1^{x_i,\mathtt{t}})$ if the ℓ-th literal in c_j is $\overline{x_i}$, and we add an edge $(b_1^{c_j,\ell}, a_1^{x_i,\mathtt{f}})$ if the ℓ-th literal in c_j is x_i. Note that we connect vertices corresponding to *negative* literals with vertices corresponding to a `true` value assignment for the given variable, and similarly, we

connect vertices corresponding to *positive* literals with vertices corresponding to a `false` value assignment for the given variable. Finally, we create the edge $e^\star = (u, v)$.

We define the preference relation $\prec$ through the following preference lists, ordered in decreasing order of preference. Here and henceforth, any set in a preference list should be interpreted as a list of its elements ordered arbitrarily.

u: v, v';
v: u, u';
u': $v, b_0^{x_1,\mathtt{t}}, \ldots, b_0^{x_n,\mathtt{t}}$;
v': $u, a_0^{c_1,1}, \ldots, a_0^{c_m,1}$;

$a_0^{c_j,1}$: $b_1^{c_j,1}$, v', $b_0^{c_j,1}$	where $j \in [m]$;
$a_0^{c_j,\ell}$: $b_1^{c_j,\ell}$, $b_0^{c_j,\ell-1}$, $b_0^{c_j,\ell}$	where $j \in [m]$ and $\ell \in \{2,3\}$;
$b_0^{c_j,\ell}$: $a_0^{c_j,\ell}$, $a_1^{c_j,\ell}$, $a_0^{c_j,\ell+1}$	where $j \in [m]$ and $\ell \in \{1,2\}$;
$b_0^{c_j,3}$: $a_0^{c_j,3}$, $a_1^{c_j,3}$, a^{c_j}	where $j \in [m]$;
$a_1^{c_j,\ell}$: $b_0^{c_j,\ell}$, $b_1^{c_j,\ell}$	where $j \in [m]$ and $\ell \in [3]$;
$b_1^{c_j,\ell}$: $a_1^{c_j,\ell}$, $a_1^{x_i,\lambda}$, $a_0^{c_j,\ell}$	where $j \in [m]$, $\ell \in [3]$ and $(b_1^{c_j,\ell}, a_1^{x_i,\lambda}) \in F$;
$a_0^{x_i,\mathtt{t}}$: $b_0^{x_i,\mathtt{t}}$, $b_1^{x_i,\mathtt{t}}$, $b_0^{x_i,\mathtt{f}}$	where $i \in [n]$;
$a_0^{x_i,\mathtt{f}}$: $b_0^{x_i,\mathtt{f}}$, $b_1^{x_i,\mathtt{f}}$, b^{x_i}	where $i \in [n]$;
$b_0^{x_i,\mathtt{t}}$: $a_1^{x_i,\mathtt{t}}$, u', $a_0^{x_i,\mathtt{t}}$	where $i \in [n]$;
$b_0^{x_i,\mathtt{f}}$: $a_1^{x_i,\mathtt{f}}$, $a_0^{x_i,\mathtt{t}}$, $a_0^{x_i,\mathtt{f}}$	where $i \in [n]$;
$a_1^{x_i,\lambda}$: $b_1^{x_i,\lambda}$, $N_{G[F]}(a^{x_i,\lambda})$, $b_0^{x_i,\lambda}$	where $i \in [n]$;
$b_1^{x_i,\lambda}$: $a_0^{x_i,\lambda}$, $a_1^{x_i,\lambda}$	where $i \in [n]$;
a^{c_j}: $b_0^{c_j,3}$	where $j \in [m]$;
b^{x_i}: $a_0^{x_i,\mathtt{f}}$	where $i \in [n]$.

We finish the construction by setting the cost function as $c \equiv 1$, our budget as $k = 1$, the desired utility value as $t = 1$, and the utility function as

$$\omega(e) = \begin{cases} 1 & \text{if } e = (u, v'), \\ 0 & \text{otherwise.} \end{cases}$$

Correctness. First, let us observe that the utility of a matching M achieves our objective value $t = 1$ if and only if it contains the edge (u, v'). However, if M contains (u, v'), then the edge $e^\star = (u, v)$ must be a blocking edge. Since $e^\star$ has cost 1 and our budget is $k = 1$, the edge $e^\star$ must be the unique blocking edge. Therefore, if M is feasible, then $\mathrm{bp}(M) = \{e^\star\}$. In fact, the converse is also true: if a popular matching is blocked solely by $e^\star$, then by Observation 4 it must match u, and hence must contain the edge (u, v'). Thus, $(G, \prec, \omega, c, t, k)$ is a 'yes'-instance of MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS if and only if $(G, \prec, \{e^\star\})$ is a 'yes'-instance of POPULARITY WITH FIXED INSTABILITY.

Let us define a matching M_0 that contains edges (u, v') and (u', v) and otherwise matches each vertex $a_h^\sigma \in A^x \cup A^c$ for any possible values of σ and h to the vertex $b_h^\sigma \in B^x \cup B^c$ (see again Figure 2). To see that M_0 is stable in $G - e^\star$, note that vertices in $A^x \cup B^c \cup \{u', v'\}$ get their top choice in M_0, and they cover all edges (including consistency edges) in $G - e^\star$. Observe that M_0 leaves exactly the vertices in $\hat{A} \cup \hat{B}$ unmatched.

We now show that φ is satisfiable if and only if $(G, \prec, \omega, c, t, k)$ admits a feasible popular matching.

Direction "⇒". Assume that a truth assignment $\alpha : \{x_1, \ldots, x_n\} \to \{\mathtt{t}, \mathtt{f}\}$ satisfies φ, where $\mathtt{t}$ and $\mathtt{f}$ stand for `true` and `false`, respectively. Let $\tau(c_j)$ denote some literal in c_j that is set
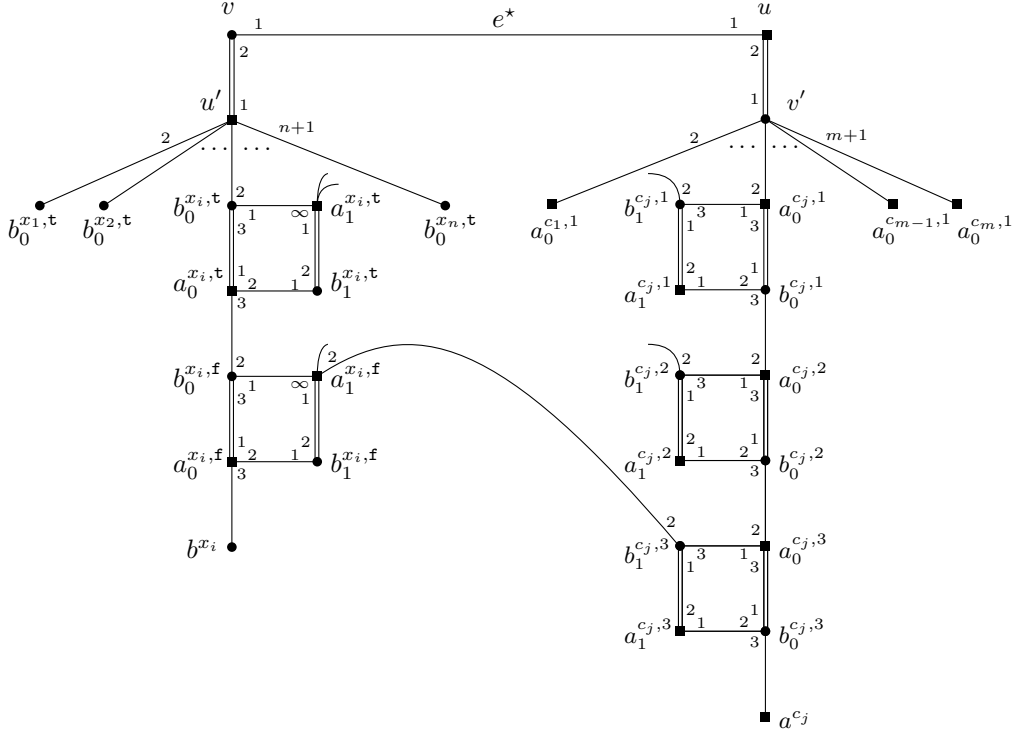
Figure 2: Illustration of the reduction in the proof of Theorem 5. Edges of M_0 are depicted as double lines. The figure only shows one consistency edge, corresponding to a situation where the third literal in c_j is x_i as a positive literal. Here and in every figure depicting a bipartite graph, we use squares and circles to distinguish between vertices on the two sides.

to `true` by α. We define a matching M through determining its symmetric difference with M_0 as

$$M \triangle M_0 = \left(\bigcup_{i=1}^{n} C^{x_i, \alpha(x_i)} \right) \cup \left(\bigcup_{j=1}^{m} C^{c_j, \tau(c_j)} \right).$$

We claim that M is a popular matching in G with $\mathrm{bp}(M) = \{e^\star\}$; we prove this by checking all conditions in Observation 4.

Condition (c1) clearly holds, as $e^\star = (u, v)$ blocks M. To see condition (c2), we show that no other edge blocks M. Note that any blocking edge must be incident to some vertex in $V(M \triangle M_0)$. However, no edge contained in a cycle $C^{x_i, \alpha(x_i)}$ can block M, since M assigns vertices of B in that cycle their top choice. Similarly, no edge in a cycle $C^{c_j, \tau(c_j)}$ may block M, since M assigns vertices of A in that cycle their top choice. Neither can edges of P_j or Q_i block M, due to similar reasons. Hence, any blocking edge must be a consistency edge. However, both endpoints of a consistency edge $f \in F$ are assigned their top choice in M_0, so f can only block M if both of its endpoints belong to $V(M \triangle M_0)$. Let $f = (a^{x_i, \lambda}, b^{c_j, \ell})$ be a consistency edge; then x_i is the ℓ-th variable in c_j. Suppose that both endpoints of f are contained in $V(M \triangle M_0)$: then $\tau(c_j) = \ell$ and $\alpha(x_i) = \lambda$. However, by construction of F, if c_j contains x_i as a positive literal, then $\lambda = \mathtt{f}$, but then setting x_i to `false` does not yield a true literal in c_j, contradicting $\alpha(x_i) = \lambda = \mathtt{f}$. Similarly, if c_j contains x_i as a negative literal, then $\lambda = \mathtt{t}$, but then setting x_i to `true` does not yield a true literal in c_j, contradicting $\alpha(x_i) = \lambda = \mathtt{t}$. Hence, $f \notin \mathrm{bp}(M)$. Thus, M is stable in $G - e^\star$.

Let us now show that condition (c3) in Observation 4 holds. Let us call an edge (a, b) where

11

both a and b prefer their partner in M to each other a $(-,-)$ edge; recall that $(G - e^\star)_M$ is obtained by deleting all $(-,-)$ edges from $G - e^\star$. Clearly, the even-length M-alternating paths leading from vertices unmatched by M (and by M_0) to u are exactly the paths $P_j^\triangle := P_j \triangle C^{c_j, \tau(c_j)}$, $j \in [m]$, and similarly, the even-length M-alternating paths leading from vertices unmatched by M to v are exactly the paths $Q_i^\triangle := Q_i \triangle C^{x_i, \alpha(x_i)}$, $i \in [n]$. Note that any path $P_j^\triangle$ contains a $(-,-)$ edge w.r.t. M, namely the edge connecting $a_0^{c_j, \tau(c_j)}$ to its second choice, that is, v' if $\tau(c_j) = 1$ and $b_0^{c_j, \tau(c_j)-1}$ otherwise. Similarly, $Q_i^\triangle$ contains a $(-,-)$ edge w.r.t. M, namely the edge connecting $b_0^{x_i, \alpha(x_i)}$ to its second choice, that is, u' if $\alpha(x_i) = \mathtt{t}$ and $a_0^{x_i, \mathtt{t}}$ otherwise. So neither $P_j^\triangle$ nor $Q_i^\triangle$ is a path in $(G - e^\star)_M$ for any j or i. Hence, condition (c3/i) holds.

Let us now prove that there is no M-alternating path in $(G - e^\star)_M$ from u to v. Observe that any such path must have the following properties.

- It starts with a subpath of P_j for some $j \in [m]$,
- reaches the cycle $C^{c_j, \ell}$ for some $\ell \in [3]$ through the second choice of $a_0^{c_j, \ell}$ (let us call this edge e^{c_j}),
- then after traversing edges of this cycle goes through some consistency edge $(b_1^{c_j, \ell}, a^{x_i, \lambda})$ for some $\lambda \in \{\mathtt{f}, \mathtt{t}\}$ and $i \in [n]$,
- traverses edges of the cycle $C^{x_i, \lambda}$ and leaves it via the second choice of $b_0^{x_i, \lambda}$ (let us call this edge e^{x_i}).

If $\ell = \tau(c_j)$, then e^{c_j} is a $(-,-)$ edge with respect to M. Similarly, if $\lambda = \alpha(x_i)$, then e^{x_i} is a $(-,-)$ edge. If neither of these conditions hold, then $b_1^{c_j, \ell}$ and $a_1^{x_i, \lambda}$ both get their top choice in M, implying that $(b_1^{c_j, \ell}, a_1^{x_i, \lambda})$ is a $(-,-)$ edge. Hence, any M-alternating path from u to v contains a $(-,-)$ edge, and therefore condition (c3/ii) holds as well, proving this direction of our reduction.

Direction "$\Leftarrow$". Let us now suppose that M is a popular matching in G blocked only by edge $e^\star = (u, v)$. By Observation 4, M is stable in $G - e^\star$, and so (u, v') and (v, u') are both in M. Also, since all stable matchings in $G - e^\star$ leave the same vertices unmatched, we know from the stability of M_0 in $G - e^\star$ that the set of unmatched vertices in M is $\hat{A} \cup \hat{B}$. Using this, a simple reckoning of the structure of G implies that M contains exactly two edges of each 4-cycle, and does not contain any consistency edge. Note that for each $j \in [m]$ the cycle $C^{c_j, \ell}$ must be contained in $M \triangle M_0$ for some $\ell \in [3]$, as otherwise P_j would violate condition (c3/i); we set $\tau(c_j) = \ell$ for such an ℓ. Analogously, for each $i \in [n]$ the cycle $C^{x_i, \lambda}$ must be contained in $M \triangle M_0$ for some $\lambda \in \{\mathtt{t}, \mathtt{f}\}$, as otherwise Q_i would violate condition (c3/i); we set $\alpha(x_i) = \lambda$.

We claim that the truth assignment α satisfies φ. To prove this, let us consider some $j \in [m]$. By the definition of τ, we know that $M(b_1^{c_j, \tau(c_j)}) = a_0^{c_j, \tau(c_j)}$, which is the worst choice of $b_1^{c_j, \tau(c_j)}$. Since the consistency edge f incident to $b_1^{c_j, \tau(c_j)}$ cannot block M in $G - e^\star$ by condition (c2), we obtain that f's other endpoint, let us denote it by $a_1^{x_i, \lambda}$, must be matched to its top choice by M. Therefore, we get $\lambda \neq \alpha(x_i)$. By our definition of the consistency edges, this implies that the $\tau(c_j)$-th literal in c_j is set to $\mathtt{true}$ by α, finishing our proof. $\square$

3.3 Main hardness result

Theorem 6 is the strengthening of Theorem 5 for the case when the input graph has maximum degree 3, and preferences are single-peaked and single-crossing. To obtain this generalization, we reduce from a variant of 3-SAT where each variable occurs at most three times; additionally, we use a well-known technique where vertices a with $\delta(a) > 3$ are replaced by a path. Despite the conceptual simplicity of this modification, maintaining the crucial properties of the reduction

while also ensuring that the constructed preference system is single-peaked and single-crossing is a delicate task.

Theorem 6. *The* Max-Utility Popular Matching with Instability Costs *problem is* NP-*complete even if*

- *the input graph G is bipartite with $\Delta_G = 3$,*
- *preferences are single-peaked and single-crossing,*
- *the cost function is $c \equiv 1$,*
- *the budget is $k = 1$,*
- *the utility function is $\omega(e) = \begin{cases} 1 & \text{if } e = f^\star, \\ 0 & \text{otherwise,} \end{cases}$ for some $f^\star \in E(G)$, and*
- *the objective value is $t = 1$.*

The Popularity with Fixed Instability *problem is* NP-*complete even if*

- *the input graph G is bipartite,*
- *$\Delta_G = 3$, preferences are single-peaked and single-crossing, and*
- *$|S| = 1$ for the set S of blocking edges.*

Proof. Since we can test feasibility and popularity in polynomial time [38], both problems are in NP. We give a reduction from the variant of 3-SAT where each clause has three literals and each variable occurs at most three times. We are going to modify the reduction presented in the proof of Theorem 5, re-using all definitions there. See Figure 3 for an illustration.
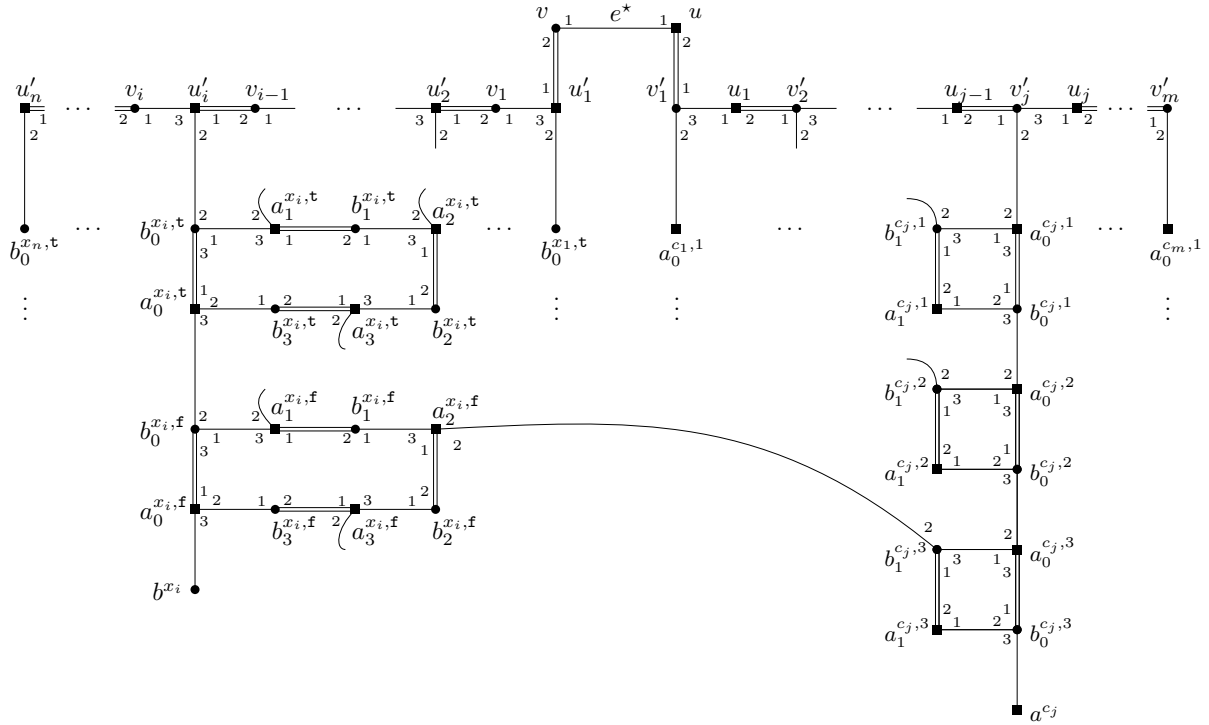

Figure 3: Illustration of the reduction in the proof of Theorem 6. Edges of $\widetilde{M_0}$ are depicted as double lines. The figure only shows one consistency edge, corresponding to a situation where the third literal in c_j is x_i as a positive literal, and x_i in c_j is the second occurrence of the variable x_i in the input formula.

13

Construction. First, for each $i \in [n]$ and $\lambda \in \{\mathtt{t}, \mathtt{f}\}$ we replace $C^{x_i, \lambda}$ with an 8-cycle $\widetilde{C}^{x_i, \lambda}$ containing vertices $a_h^{x_i, \lambda}$ and $b_h^{x_i, \lambda}$ for $h \in \{0, 1, 2, 3\}$ (in the usual order). We replace the set F with a new set $\widetilde{F}$ of consistency edges as follows: for each $j \in [m]$ and $\ell \in [3]$, if the ℓ-th literal of c_j is the h-th occurrence of variable x_i for some $i \in [n]$ and $h \in [3]$, then we add the edge $(a_h^{x_i, \lambda}, b_1^{c_j, \ell})$ to $\widetilde{F}$, where $\lambda = \mathtt{f}$ if the h-th occurrence of x_i is as a positive literal (the ℓ-th literal in c_j) and $\lambda = \mathtt{t}$ otherwise.

Next, we replace v' by a path $(v_1', u_1, \ldots, v_{m-1}', u_{m-1}, v_m')$, replacing each edge $(v', a_0^{c_j, 1})$ with the edge $(v_j', a_0^{c_j, 1})$. Analogously, we replace u' by a path $(u_1', v_1, \ldots, u_{n-1}', v_{n-1}, u_n')$, replacing each edge $(u', b_0^{x_i, \mathtt{t}})$ with the edge $(u_i', b_0^{x_i, \mathtt{t}})$.

The preferences of the vertices are as follows; we omit those vertices whose preferences are the same as defined in the proof of Theorem 5.

$\quad u$: v, v_1';
$\quad v$: u, u_1';
$\quad u_i'$: $v_{i-1}, b_0^{x_i, \mathtt{t}}, v_i$ $\qquad\qquad$ where $i \in [n-1]$ and $v_0 = v$;
$\quad u_n'$: $v_{n-1}, b_0^{x_n, \mathtt{t}}$
$\quad v_j'$: $u_{j-1}, a_0^{c_j, 1}, u_j$ $\qquad\qquad$ where $j \in [m-1]$ and $u_0 = u$;
$\quad v_m'$: $u_{m-1}, a_0^{c_m, 1}$
$\quad v_i$: u_i', u_{i+1}' $\qquad\qquad\qquad$ where $i \in [n-1]$;
$\quad u_j$: v_j', v_{j+1}' $\qquad\qquad\qquad$ where $j \in [m-1]$;
$\quad a_0^{c_j, 1}$: $b_1^{c_j, 1}, v_j', b_0^{c_j, 1}$ $\qquad\quad$ where $j \in [m]$;
$\quad b_1^{c_j, \ell}$: $a_1^{c_j, \ell}, a_h^{x_i, \lambda}, a_0^{c_j, \ell}$ $\quad\;$ where $j \in [m]$, $\ell \in [3]$ and $(b_1^{c_j, \ell}, a_h^{x_i, \lambda}) \in \widetilde{F}$;
$\quad a_0^{x_i, \mathtt{t}}$: $b_0^{x_i, \mathtt{t}}, b_3^{x_i, \mathtt{t}}, b_0^{x_i, \mathtt{f}}$ $\qquad$ where $i \in [n]$;
$\quad a_0^{x_i, \mathtt{f}}$: $b_0^{x_i, \mathtt{f}}, b_3^{x_i, \mathtt{f}}, b^{x_i}$ $\qquad$ where $i \in [n]$;
$\quad a_h^{x_i, \lambda}$: $b_h^{x_i, \lambda}, b_1^{c_j, \ell}, b_{h-1}^{x_i, \lambda}$ $\qquad$ where $i \in [n]$, $h \in [3]$ and $(b_1^{c_j, \ell}, a_h^{x_i, \lambda}) \in \widetilde{F}$;
$\quad a_h^{x_i, \lambda}$: $b_h^{x_i, \lambda}, b_{h-1}^{x_i, \lambda}$ $\qquad\qquad$ where $i \in [n]$, $h \in [3]$ and $a_h^{x_i, \lambda} \notin V(\widetilde{F})$;
$\quad b_0^{x_i, \mathtt{t}}$: $a_1^{x_i, \mathtt{t}}, u_i', a_0^{x_i, \mathtt{t}}$ $\qquad\quad$ where $i \in [n]$;
$\quad b_h^{x_i, \lambda}$: $a_{h+1 \bmod 4}^{x_i, \lambda}, a_h^{x_i, \lambda}$ $\qquad$ where $i \in [n]$ and $h \in [3]$.

Let $(\widetilde{G}, \prec)$ be the preference system defined this way. Note that $\Delta_{\widetilde{G}} = 3$. To create an instance of MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS, we set all utilities to 0 except for the edge (u, v_1') whose utility is 1, we set $c \equiv 1$, and we let $t = k = 1$.

Correctness. To show that $\widetilde{G}$ admits a popular matching with $\mathrm{bp}(M) = \{e^\star\}$ if and only if φ is satisfiable, one can apply essentially the same arguments used in the proof of Theorem 5; for this, however, we need to define a new matching $\widetilde{M_0}$ instead of M_0. To do so, for each $i \in [n]$ and $\lambda \in \{\mathtt{t}, \mathtt{f}\}$ we first add the edges $(a_2^{x_i, \lambda}, b_2^{x_i, \lambda})$ and $(a_3^{x_i, \lambda}, b_3^{x_i, \lambda})$ to M_0. Second, we replace the edge (v, u') with edges (v, u_1') and (v_i, u_{i+1}') for each $i \in [n-1]$. Similarly, we replace the edge (u, v') with edges (u, v_1') and (u_j, v_{j+1}') for each $j \in [m-1]$. Let $\widetilde{M_0}$ denote the matching obtained this way. Note that $\widetilde{M_0}$ is stable in $\widetilde{G} - e^\star$.

Now, the reasoning used in the proof of Theorem 5 can be applied, with $\widetilde{M_0}$ taking the place of M_0, and with some trivial modifications where necessary. Assuming that the preference system has the desired properties, this shows that POPULARITY WITH FIXED INSTABILITY is NP-hard under the conditions stated in the theorem.

To prove the result for MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS, observe that M is a popular matching in $\widetilde{G}$ with $\mathrm{bp}(M) = \{e^\star\}$ if and only if M is a popular matching in $\widetilde{G}$ whose utility is at least $t = 1$ and whose blocking edges have cost at most $k = 1$ (here we use Observation 4 again).

It remains to prove that the constructed preference system $(\widetilde{G}, \prec)$ has the required properties. Let $(\widetilde{A}, \widetilde{B})$ denote the (unique) bipartition of $\widetilde{G}$ where $\widetilde{A}$ contains u and $\widetilde{B}$ contains v.

Single-peaked property. To show single-peakedness, we define two lists for each $j \in [m]$ and also for each $i \in [n]$:

$$
\begin{aligned}
A_j^c &= (a^{c_j}, a_0^{c_j,3}, a_1^{c_j,3}, a_0^{c_j,2}, a_1^{c_j,2}, a_0^{c_j,1}, a_1^{c_j,1}), \\
B_j^c &= (b_0^{c_j,3}, b_1^{c_j,3}, b_0^{c_j,2}, b_1^{c_j,2}, b_0^{c_j,1}, b_1^{c_j,1}), \\
A_i^x &= (a_3^{x_i,\mathsf{t}}, a_2^{x_i,\mathsf{t}}, a_1^{x_i,\mathsf{t}}, a_0^{x_i,\mathsf{t}}, a_3^{x_i,\mathsf{f}}, a_2^{x_i,\mathsf{f}}, a_1^{x_i,\mathsf{f}}, a_0^{x_i,\mathsf{f}}), \\
B_i^x &= (b_3^{x_i,\mathsf{t}}, b_2^{x_i,\mathsf{t}}, b_1^{x_i,\mathsf{t}}, b_0^{x_i,\mathsf{t}}, b_3^{x_i,\mathsf{f}}, b_2^{x_i,\mathsf{f}}, b_1^{x_i,\mathsf{f}}, b_0^{x_i,\mathsf{f}}, b^{x_i}).
\end{aligned}
$$

Then we can define an axis for each of $\widetilde{A}$ and $\widetilde{B}$ as follows:

$$
u_{m-1}, \ldots, u_1, A_1^c, \ldots, A_m^c, u, u_1', \ldots, u_n', A_1^x, \ldots, A_n^x; \qquad \text{(Axis } \widetilde{A}\text{)}
$$

$$
B_1^c, \ldots, B_m^c, v_m', \ldots, v_1', v, B_1^x, \ldots, B_n^x, v_1, \ldots, v_{m-1}. \qquad \text{(Axis } \widetilde{B}\text{)}
$$

It is straightforward to check that each vertex has single-peaked preferences with respect to the axis containing its neighbors. Note that the preferences of a degree-2 vertex are always single-peaked, while the preferences of a vertex a with $\delta(a) = 3$ are single-peaked w.r.t. a given axis exactly if the least-preferred neighbor of a does not lie between its two other neighbors on the axis.

Single-crossing property. To show that the preference system $(\widetilde{G}, \prec)$ is single-crossing, we define a complete bipartite strict preference system $(K, \prec^K)$ compatible with $(\widetilde{G}, \prec)$ that is single-crossing; here K is the complete bipartite graph whose two partitions are $\widetilde{A}$ and $\widetilde{B}$.

Let us define the following vertex sets and lists of vertices:

$$
\begin{aligned}
A_h^\ell &= \{a_h^{c_j,\ell} : j \in [m]\} && \text{for each } h \in \{0,1\}, \ell \in [3]; \\
A_h^\lambda &= \{a_h^{x_i,\lambda} : i \in [n]\} && \text{for each } h \in \{0,1,2,3\}, \lambda \in \{\mathsf{t},\mathsf{f}\}; \\
B_h^\ell &= \{b_h^{c_j,\ell} : j \in [m]\} && \text{for each } h \in \{0,1\}, \ell \in [3]; \\
B_h^\lambda &= \{b_h^{x_i,\lambda} : i \in [n]\} && \text{for each } h \in \{0,1,2,3\}, \lambda \in \{\mathsf{t},\mathsf{f}\}; \\
U' &= (u_1', \ldots, u_n'); \\
V' &= (v_1', \ldots, v_m').
\end{aligned}
$$

Next, we define the preferences in $(K, \prec^K)$. We deal with the two sides separately, so first we define $\prec_b^K$ for each vertex $b \in \widetilde{B}$. Each such $\prec_b^K$ will be one of $m+2$ complete total orders over $\widetilde{A}$: these are π_0^j for each $j \in [m]$, π_1 and π_2, as they are defined below. We set $\pi_0 = \pi_0^m$, and for any set appearing in the definition of these orders, we fix one arbitrary ordering (used in all of the lists below).

$$
\begin{aligned}
\pi_0^j\colon \quad & u, u_1, \ldots, u_{j-1}, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, u_j, \ldots, u_{m-1}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}}; \\
\pi_0\colon \quad & u, u_1, \ldots, u_{m-1}, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}}; \\
\pi_1\colon \quad & u, u_1, \ldots, u_{m-1}, A_1^1, A_1^2, A_1^3, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}}, A_0^1, A_0^2, A_0^3, \hat{A}; \\
\pi_2\colon \quad & u, u_1, \ldots, u_{m-1}, A_1^1, A_1^2, A_1^3, A_0^{\mathsf{t}}, A_0^{\mathsf{f}}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^1, A_0^2, A_0^3, \hat{A}.
\end{aligned}
$$

The following claims follow directly from the definitions:
- the preference list of v_j' in $\widetilde{G}$ is a restriction of π_0^j for any $j \in [m]$;
- the preference list of $b_0^{c_j,\ell}$ in $\widetilde{G}$ is a restriction of π_0 for any $j \in [m], \ell \in [3]$;
- the preference list of $b_3^{x_i,\lambda}$ in $\widetilde{G}$ is a restriction of π_2 for any $i \in [n], \lambda \in \{\mathsf{t},\mathsf{f}\}$;

15

- the preference list of any other vertex of $\widetilde{B}$ in $\widetilde{G}$ is a restriction of π_1.

This means that each preference relation $\prec_b$ for some $b \in \widetilde{B}$ is compatible with one of the preference relations in the set $\Pi = \{\pi_0^j \mid j \in [m]\} \cup \{\pi_1, \pi_2\}$. We aim to show that $(\widetilde{G}, \prec)$ is compatible with the strict and complete preference system $(K, \prec^K)$: so far we have proved that compatibility holds for all vertices in $\widetilde{B}$, and we will deal with vertices of $\widetilde{A}$ later in the proof. Assuming now that $(\widetilde{G}, \prec)$ is compatible with $(K, \prec^K)$, in order to prove that $(\widetilde{G}, \prec)$ is single-crossing, it suffices to prove that $(K, \prec^K)$ is single-crossing.

Recall that to prove that $(K, \prec^K)$ is single-crossing, we need to shows that it is single-crossing with respect to some linear ordering of the vertices of K, that is, of $\widetilde{A} \cup \widetilde{B}$. Since $(K, \prec^K)$ is bipartite, it suffices to provide a separate ordering over $\widetilde{A}$ and over $\widetilde{B}$; we will first deal with vertices of $\widetilde{B}$. Notice, however, that defining an ordering over $\widetilde{B}$ is equivalent with defining an ordering over all preference lists in $\{\prec_b^K : b \in \widetilde{B}\}$. Therefore, it suffices to define an ordering over $\{\prec_b^K : b \in \widetilde{B}\}$. Recall that $\{\prec_b^K : b \in \widetilde{B}\}$ is, in fact, a multiset of $m + 2$ complete orders over $\widetilde{A}$, and these $m + 2$ complete orders form exactly the family Π of preference lists. Thus, we provide an ordering $\rhd$ over Π as $\pi_0^1 \rhd \cdots \rhd \pi_0^m = \pi_0 \rhd \pi_1 \rhd \pi_2$.

For any $a', a'' \in \widetilde{A}$, let $\Pi^{a' \prec a''} = \{\pi \in \Pi \mid a'' \text{ precedes } a' \text{ in } \pi\}$. It is now straightforward to verify that for any two distinct vertices $a', a'' \in \widetilde{A}$, either all preference relations in $\Pi^{a' \prec a''}$ are followed by all preference relations in $\Pi^{a'' \prec a'}$ according to $\rhd$, or vice versa: this can be checked by observing Figure 4, where we have depicted Π according to the ordering $\rhd$ and highlighted how certain sets are moved between one preference list and the next one; the figure shows that there are no two vertices in $\widetilde{A}$ that switch places more than once when traversing Π according to $\rhd$. This proves that the single-crossing property holds for vertices of $\widetilde{B}$ in $(K, \prec^K)$.

$$\pi_0^1:\ u, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, \boldsymbol{u_1}, u_2, \ldots, u_{m-1}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}};$$

$$\pi_0^2:\ u, u_1, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, u_2, \ldots, u_{m-1}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}};$$

$$\vdots$$

$$\pi_0^{j-1}:\ u, u_1, \ldots, u_{j-1}, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, \boldsymbol{u_j}, \ldots, u_{m-1}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}};$$

$$\pi_0^j:\ u, u_1, \ldots, u_j, A_0^1, A_1^1, A_0^2, A_1^2, A_0^3, A_1^3, \hat{A}, u_{j+1}, \ldots, u_{m-1}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}};$$

$$\vdots$$

$$\pi_0^m = \pi_0:\ u, u_1, \ldots, u_{m-1}, \boldsymbol{A_0^1}, A_1^1, \boldsymbol{A_0^2}, A_1^2, \boldsymbol{A_0^3}, A_1^3, \hat{\boldsymbol{A}}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^{\mathsf{t}}, A_0^{\mathsf{f}}$$

$$\pi_1:\ u, u_1, \ldots, u_{m-1}, A_1^1, A_1^2, A_1^3, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', \boldsymbol{A_0^{\mathsf{t}}}, \boldsymbol{A_0^{\mathsf{f}}}, A_0^1, A_0^2, A_0^3, \hat{A}$$

$$\pi_2:\ u, u_1, \ldots, u_{m-1}, A_1^1, A_1^2, A_1^3, A_0^{\mathsf{t}}, A_0^{\mathsf{f}}, A_3^{\mathsf{f}}, A_2^{\mathsf{f}}, A_1^{\mathsf{f}}, A_3^{\mathsf{t}}, A_2^{\mathsf{t}}, A_1^{\mathsf{t}}, U', A_0^1, A_0^2, A_0^3, \hat{A}$$
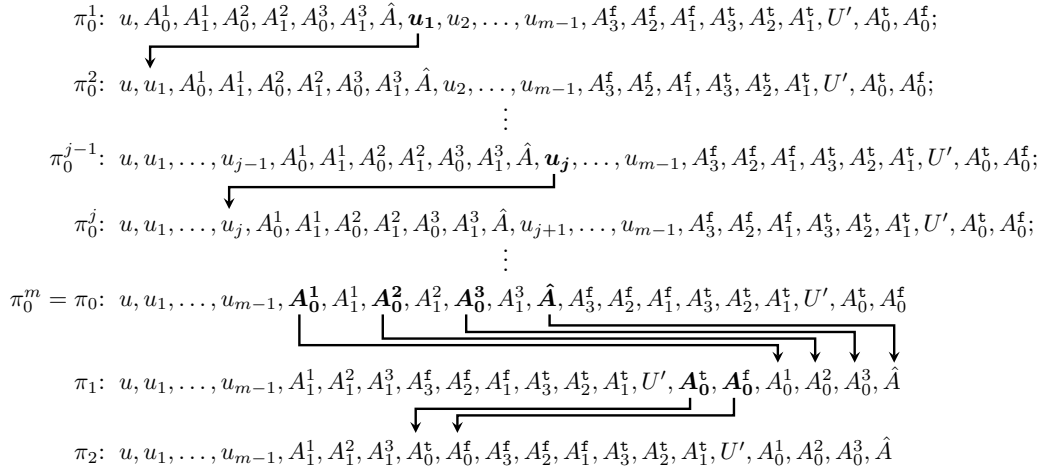
Figure 4: Illustration for proving the single-crossing property of the constructed instance, for vertices in $\widetilde{B}$. The figure shows the preference lists in Π, ordered according to $\rhd$ (from top to bottom). At each row we highlighted in bold those vertex sets that have to be "moved" in order to obtain the next row, and we used arrows to display these changes. Observe that no two vertices $a', a'' \in \widetilde{A}$ switch places more than once when traversing Π in the given order.

To deal with vertices of $\widetilde{A}$ in an analogous way, we define $\prec_a^K$ for each $a \in \widetilde{A}$ to be one of $n + 4$ complete total orders over $\widetilde{B}$: these are ϕ_0^i for each $i \in [n]$, ϕ_1, ϕ_2, ϕ_3 and ϕ_4, as they are defined below. We set $\phi_0 = \phi_0^n$, and for any set appearing below in these orders, we fix one arbitrary ordering (used in all of the lists below).

16

$$\phi_0^i: \quad v, v_1, \ldots, v_{i-1}, B_3^{\mathtt{t}}, B_3^{\mathtt{f}}, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathtt{t}}, B_0^{\mathtt{f}}, \hat{B}, v_{i-1}, \ldots, v_{n-1};$$
$$\phi_0: \quad v, v_1, \ldots, v_{n-1}, B_3^{\mathtt{t}}, B_3^{\mathtt{f}}, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathtt{t}}, B_0^{\mathtt{f}}, \hat{B};$$
$$\phi_1: \quad v, v_1, \ldots, v_{n-1}, B_3^{\mathtt{t}}, B_3^{\mathtt{f}}, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, B_0^{\mathtt{t}}, B_0^{\mathtt{f}}, \hat{B};$$
$$\phi_2: \quad v, v_1, \ldots, v_{n-1}, B_3^{\mathtt{t}}, B_3^{\mathtt{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, B_0^{\mathtt{t}}, B_0^{\mathtt{f}}, \hat{B};$$
$$\phi_3: \quad v, v_1, \ldots, v_{n-1}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_3^{\mathtt{t}}, B_3^{\mathtt{f}}, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, B_0^{\mathtt{t}}, B_0^{\mathtt{f}}, \hat{B};$$
$$\phi_4: \quad v, v_1, \ldots, v_{n-1}, B_0^1, B_0^2, B_0^3, B_1^3, B_1^2, B_1^1, V', B_0^{\mathtt{t}}, B_3^{\mathtt{t}}, B_0^{\mathtt{f}}, B_3^{\mathtt{f}}, B_2^{\mathtt{t}}, B_2^{\mathtt{f}}, B_1^{\mathtt{t}}, B_1^{\mathtt{f}}, \hat{B}.$$

The following claims follow directly from the definitions:

- the preference list of u_i' in $\widetilde{G}$ is a restriction of ϕ_0^i for any $i \in [n]$;
- the preference list of $a_1^{x_i,\lambda}$ in $\widetilde{G}$ is a restriction of ϕ_0 for any $i \in [n], \lambda \in \{\mathtt{t},\mathtt{f}\}$;
- the preference list of $a_2^{x_i,\lambda}$ in $\widetilde{G}$ is a restriction of ϕ_1 for any $i \in [n], \lambda \in \{\mathtt{t},\mathtt{f}\}$;
- the preference list of $a_3^{x_i,\lambda}$ in $\widetilde{G}$ is a restriction of ϕ_2 for any $i \in [n], \lambda \in \{\mathtt{t},\mathtt{f}\}$;
- the preference list of $a_0^{x_i,\lambda}$ in $\widetilde{G}$ is a restriction of ϕ_4 for any $i \in [n], \lambda \in \{\mathtt{t},\mathtt{f}\}$;
- the preference list of $a_1^{c_j,\ell}$ in $\widetilde{G}$ is a restriction of ϕ_4 for any $j \in [m], \ell \in [3]$;
- the preference list of any other vertex in $\widetilde{G}$ is a restriction of ϕ_3.

This means that each preference relation $\prec_a$ for some $a \in \widetilde{A}$ is compatible with one of the preference relations in the set $\Phi = \{\phi_0^i \mid i \in [n]\} \cup \{\phi_1, \phi_2, \phi_3, \phi_4\}$. Hence, to prove that $(\widetilde{G}, \prec)$ is single-crossing with regard to vertices of $\widetilde{A}$, it suffices to prove that $(K, \prec^K)$ is single-crossing with regard to vertices of $\widetilde{A}$.

We apply the same approach that we used for proving the single-crossing property of $(K, \prec^K)$ when restricted to the vertices in $\widetilde{B}$. Using the same line of thought, we get that in order to prove the single-crossing property of $(K, \prec^K)$ for vertices of $\widetilde{A}$, it suffices to order the elements of Φ instead of ordering $\widetilde{A}$ itself. Therefore, we provide an ordering $\triangleright$ over Φ as $\phi_0^1 \triangleright \cdots \triangleright \phi_0^n = \phi_0 \triangleright \phi_1 \triangleright \phi_2 \triangleright \phi_3 \triangleright \phi_4$. For any $b', b'' \in \widetilde{B}$, let $\Phi^{b' \prec b''} = \{\phi \in \Phi \mid b'' \text{ precedes } b' \text{ in } \phi\}$. It is now straightforward to verify that for any two distinct vertices $b', b'' \in \widetilde{B}$, either all preference relations in $\Phi^{b' \prec b''}$ are followed by all preference relations in $\Phi^{b'' \prec b'}$ according to $\triangleright$, or vice versa: this can be checked in Figure 5 by observing that there are no two vertices $b', b'' \in \widetilde{B}$ that switch places more than once when traversing Φ. This proves that the single-crossing property holds for vertices of $\widetilde{A}$ in $(K, \prec^K)$ as well, finishing our proof. $\qquad\square$

3.4 Uniform utilities

We also investigate whether MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS becomes easier if we do not allow edges with zero utility: Theorem 7 below shows that if the utility function is $\omega \equiv 1$ and we aim for a popular matching that has only one more edge than a stable matching, the problem is NP-complete even if all other restrictions of Theorem 6 remain in place.

Theorem 7. *The* MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS *problem is* NP-*complete even if*

- *the input graph $G = (V, E)$ is bipartite and $\Delta_G = 3$,*
- *preferences are single-peaked and single-crossing,*
- *the cost function is $c \equiv 1$,*
- *the budget is $k = 1$,*
- *the utility function is $\omega \equiv 1$, and*
- *the objective value is $t = |M_s| + 1 = \frac{|V|}{2}$ where M_s is a stable matching in G.*
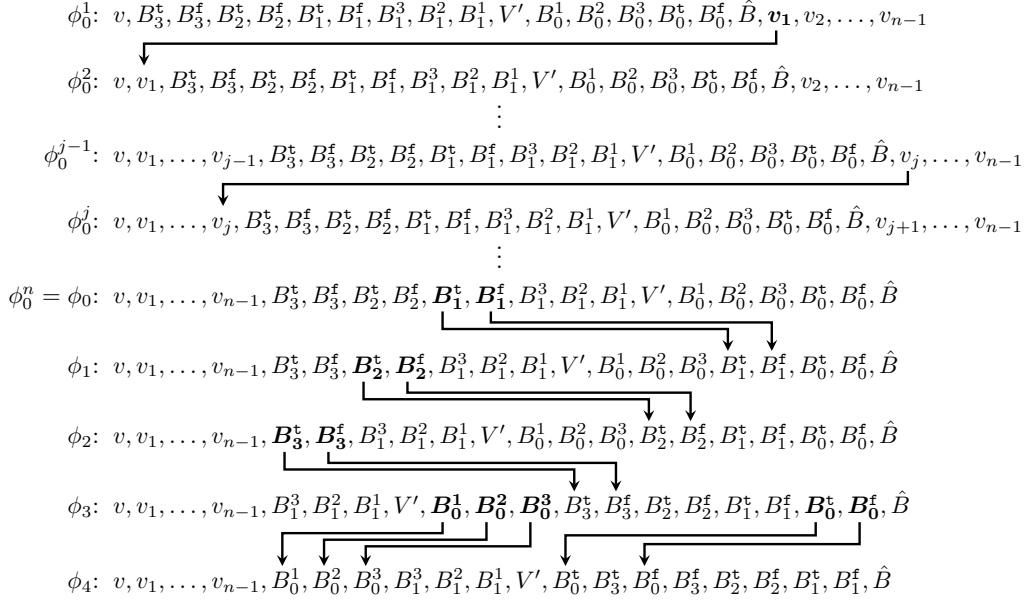
$$\phi_0^1:\ v, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}, \boldsymbol{v_1}, v_2, \ldots, v_{n-1}$$

$$\phi_0^2:\ v, v_1, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}, v_2, \ldots, v_{n-1}$$

$$\vdots$$

$$\phi_0^{j-1}:\ v, v_1, \ldots, v_{j-1}, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}, v_j, \ldots, v_{n-1}$$

$$\phi_0^{j}:\ v, v_1, \ldots, v_j, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}, v_{j+1}, \ldots, v_{n-1}$$

$$\vdots$$

$$\phi_0^n = \phi_0:\ v, v_1, \ldots, v_{n-1}, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, \boldsymbol{B_1^{\mathsf{t}}}, \boldsymbol{B_1^{\mathsf{f}}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}$$

$$\phi_1:\ v, v_1, \ldots, v_{n-1}, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, \boldsymbol{B_2^{\mathsf{t}}}, \boldsymbol{B_2^{\mathsf{f}}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}$$

$$\phi_2:\ v, v_1, \ldots, v_{n-1}, \boldsymbol{B_3^{\mathsf{t}}}, \boldsymbol{B_3^{\mathsf{f}}}, B_1^3, B_1^2, B_1^1, V', B_0^1, B_0^2, B_0^3, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, B_0^{\mathsf{t}}, B_0^{\mathsf{f}}, \hat{B}$$

$$\phi_3:\ v, v_1, \ldots, v_{n-1}, B_1^3, B_1^2, B_1^1, V', \boldsymbol{B_0^1}, \boldsymbol{B_0^2}, \boldsymbol{B_0^3}, B_3^{\mathsf{t}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, \boldsymbol{B_0^{\mathsf{t}}}, \boldsymbol{B_0^{\mathsf{f}}}, \hat{B}$$

$$\phi_4:\ v, v_1, \ldots, v_{n-1}, B_0^1, B_0^2, B_0^3, B_1^3, B_1^2, B_1^1, V', B_0^{\mathsf{t}}, B_3^{\mathsf{t}}, B_0^{\mathsf{f}}, B_3^{\mathsf{f}}, B_2^{\mathsf{t}}, B_2^{\mathsf{f}}, B_1^{\mathsf{t}}, B_1^{\mathsf{f}}, \hat{B}$$

Figure 5: Illustration for proving the single-crossing property of the constructed instance, for vertices in $\widetilde{A}$. The figure shows the preference lists in Φ, ordered according to $\rhd$ using the notation of Figure 4. Observe that no two vertices $b', b'' \in \widetilde{B}$ switch places more than once when traversing Φ in the given order.

Proof. Again, we give a reduction from the variant of 3-SAT where each clause has three literals and each variable occurs at most three times. We are going to modify the reduction presented in the proof of Theorem 6, re-using definitions there, and placing emphasis on condition (c3/ii) instead of condition (c3/i) of Observation 4. See Figure 6 for an illustration.
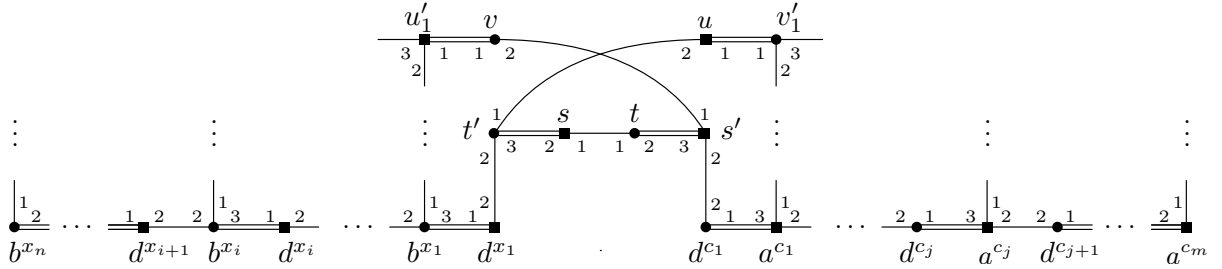
Figure 6: Illustration of the reduction in the proof of Theorem 7, depicting the part of graph $\hat{G}$ where it differs from graph $\widetilde{G}$ (shown in Figure 3). Double lines denote the matching $\hat{M}_0$.

Construction. We start with adding a set $D^X \cup D^C \cup \{s, s', t, t'\}$ of new vertices to $\widetilde{G}$ where $D^X = \{d^{x_i} : i \in [n]\}$ and $D^C = \{d^{c_j} : j \in [m]\}$. Among the vertices of $\widetilde{G}$, only those in $\hat{A} \cup \hat{B} \cup \{u, v\}$ will be adjacent to newly introduced vertices; the preferences of all other vertices remain unchanged. The preferences of the new vertices and of those that are adjacent to them are as follows.

$$s:\ t, t';$$
$$t:\ s, s';$$

18

$$s':\ v, d^{c_1}, t;$$
$$t':\ u, d^{x_1}, s;$$
$$u:\ v'_1, t';$$
$$v:\ u'_1, s';$$
$$d^{x_1}:\ b^{x_1}, t';$$
$$d^{x_i}:\ b^{x_i}, b^{x_{i-1}} \qquad\qquad \text{where } i \in \{2, \ldots, n\};$$
$$b^{x_i}:\ a_0^{x_i, \mathbf{f}}, d^{x_{i+1}}, d^{x_i} \qquad\qquad \text{where } i \in [n-1];$$
$$b^{x_n}:\ a_0^{x_n, \mathbf{f}}, d^{x_n};$$
$$d^{c_1}:\ a^{c_1}, s';$$
$$d^{c_j}:\ a^{c_j}, a^{c_{j-1}} \qquad\qquad \text{where } j \in \{2, \ldots, m\};$$
$$a^{c_j}:\ b_0^{c_j, 3}, d^{c_{j+1}}, d^{c_j} \qquad\qquad \text{where } j \in [m-1];$$
$$a^{c_m}:\ b_0^{c_m, 3}, d^{c_m}.$$

As promised in the theorem, we set both the cost function and the utility function to be uniformly 1, and we set $t = |M_s| + 1$ and $k = 1$.

Correctness. Recall the matching $\widetilde{M_0}$ defined for the graph $\widetilde{G}$ (see Figure 3); observe that $\widetilde{M_0}$ is a matching in $\hat{G}$ as well. Let us now define a matching M_s by adding the all edges in the set $\{(d^{x_i}, b^{x_i}) : i \in [n]\} \cup \{(d^{c_j}, a^{c_j}) : j \in [m]\} \cup \{(s, t)\}$ to $\widetilde{M_0}$. It is straightforward to verify that M_s is stable in $\hat{G}$. Note that the only vertices left unmatched by M_s are s' and t'. Therefore a matching is feasible if and only if it is complete and has at most one blocking edge.

We are going to show that there is a feasible and popular matching in $\hat{G}$ if and only if the input formula φ is satisfiable.

Direction "⇒". Suppose that M is a feasible and popular matching in $\hat{G}$. We claim that $M(s') = t$ and $M(t') = s$.

First, if $M(t') = u$, then (u, v'_1) blocks M. Second, if $M(t') = d^{x_1}$, then at least one edge incident to some vertex of D^X must block M: this follows from an observation that M cannot connect a vertex b^{x_i} to its top choice (i.e., $a_0^{x_i, \mathbf{f}}$). To see this, suppose that $(a_0^{x_i, \mathbf{f}}, b^{x_i}) \in M$. This means that $(a_0^{x_i, \mathbf{f}}, b_3^{x_i, \mathbf{f}}) \in \mathrm{bp}(M)$, so no other edge in $\widetilde{C}^{x_i, \mathbf{f}}$ can block M; however, this leads to $(b_h^{x_i, \mathbf{f}}, a_h^{x_i, \mathbf{f}}) \in M$ for $h = 3, 2, 1$, implying also $(b_0^{x_i, \mathbf{f}}, a_0^{x_i, \mathbf{t}}) \in M$. This in turn yields that the edge $(a_0^{x_i, \mathbf{t}}, b_3^{x_i, \mathbf{t}})$ blocks M, a contradiction. Hence, if $M(t') \neq s$, then at least one edge in $\mathrm{bp}(M)$ is incident to a vertex of $D^X \cup \{u\}$.

Analogously, the same arguments yield that if $M(s') \neq t$, then at least one edge in $\mathrm{bp}(M)$ is incident to a vertex of $D^C \cup \{v\}$. Hence, by $|\mathrm{bp}(M)| \leq 1$, and since no vertex of $D^X \cup \{u\}$ is adjacent to a vertex of $D^C \cup \{v\}$, at least one of (s', t) and (s, t') is in M. This implies $\mathrm{bp}(M) = \{(s, t)\}$, and consequently, $\{(s, t'), (s', t)\} \subseteq M$ because M is complete.

Since M is complete, d^{x_1} must be matched in M, yielding $(d^{x_1}, b^{x_1}) \in M$. Applying the same argument repeatedly, we get $(d^{x_i}, b^{x_i}) \in M$ for $i = 2, \ldots, n$ as well. Similarly, we get $(d^{c_j}, a^{c_j}) \in M$ for each $j \in [m]$, and also $(u'_i, v_{i-1}) \in M$ for each $i \in [n]$ and $(v'_j, u_{j-1}) \in M$ for each $j \in [m]$ where $v_0 = v$ and $u_0 = u$.

Let us define a matching $\hat{M}_0 = M_s \setminus \{(s, t)\} \cup \{(s, t'), (s', t)\}$; see again Figure 6. By the observations of the previous paragraph and using that M is complete, it follows that $M \triangle \hat{M}_0$ is the union of cycles of the form $\widetilde{C}^{x_i, \lambda}$ and $C^{c_j, h}$. Moreover, we claim that for each $i \in [n]$ there exists some $\lambda \in \{\mathbf{t}, \mathbf{f}\}$ such that $\widetilde{C}^{x_i, \lambda} \subseteq M \triangle \hat{M}_0$, and that for each $j \in [m]$ there exists some $h \in [3]$ such that $C^{c_j, h} \subseteq M \triangle \hat{M}_0$. To see this for some $i \in [n]$, consider the path

$$\hat{P}_i = (s, t', d^{x_1}, b^{x_1}, \ldots, d^{x_i}, b^{x_i}, a_0^{x_i, \mathbf{f}}, b_0^{x_i, \mathbf{f}}, a_0^{x_i, \mathbf{t}}, b_0^{x_i, \mathbf{t}}, u'_i, v_{i-1}, u'_{i-1}, \ldots, v_1, u'_1, v, s', t).$$

Note that there are no $(-, -)$ edges on $\hat{P}_i$ with respect to the matching $\hat{M}_0$. By Observation 4, this path cannot be present in $(\hat{G} - (s, t))_M$, implying that M cannot contain all edges of $\hat{M}_0 \cap \hat{P}_i$.

Thus, indeed there exists some $\lambda \in \{\mathtt{t}, \mathtt{f}\}$ for which $\widetilde{C}^{x_i, \lambda} \subseteq M \triangle \hat{M}_0$; we define $\alpha(x_i)$ to be such a value of λ.

Arguing analogously about the path

$$\hat{Q}_j = (t, s', d^{c_1}, a^{c_1}, \ldots, d^{c_j}, a^{c_j}, b_0^{c_j, 3}, a_0^{c_j, 3}, b_0^{c_j, 2}, a_0^{c_j, 2}, b_0^{c_j, 1}, a_0^{c_j, 1}, v_j', u_{j-1}, v_{j-1}', \ldots, u_1, v_1', u, t', s),$$

we get that for each $j \in [m]$ there exists some $h \in [3]$ for which $C^{c_j, h} \subseteq M \triangle \hat{M}_0$; we define $\tau(c_j)$ to be such a value h.

Based on the fact that no consistency edge can block M, one can argue in the same way as in the proof of Theorems 5 and 6 to prove that α is a truth assignment satisfying the input formula φ.

Direction "⇐". Assume that $\alpha : \{x_1, \ldots, x_n\} \to \{\mathtt{t}, \mathtt{f}\}$ is a truth assignment that satisfies φ. Let $\tau(c_j) = h$ if the h-th literal in clause c_j is set to $\mathtt{true}$ by α (any such value h works). Then we define M through determining its symmetric difference with $\hat{M}_0$ as

$$M \triangle \hat{M}_0 = \left(\bigcup_{i=1}^{n} C^{x_i, \alpha(x_i)} \right) \cup \left(\bigcup_{j=1}^{m} C^{c_j, \tau(c_j)} \right).$$

It is easy to see that M is feasible, in particular, $\mathrm{bp}(M) = \{(s, t)\}$. Using Observation 4, we can show that M is popular as well. It is straightforward to check that conditions (c1), (c2), and (c3/i) of Observation 4 hold. To see that (c3/ii) holds as well, let P be an M-alternating path P starting with (s, t') and ending with (s', t). First note that if P contains no consistency edge, then it must contain a $(-, -)$ edge w.r.t. M: either an edge connecting some $b_0^{x_i, \alpha(x_i)}$ with its second choice (let us denote this edge by e^{x_i}), or an edge connecting some $a_0^{c_j, \tau(c_j)}$ with its second choice (let us denote this edge by e^{c_j}). If P does contain some consistency edge f, then either f itself is a $(-, -)$ edge or it must be adjacent to an edge in some cycle C in $M \triangle \hat{M}_0$. Thus, either $C = \widetilde{C}^{x_i, \alpha(x_i)}$ for some $i \in [n]$, or $C = C^{c_j, \tau(c_j)}$ for some $j \in [m]$. In both cases we can identify a $(-, -)$ edge on P, namely e^{x_i} in the former case, and e^{c_j} in the latter case. This proves that M is indeed popular, and hence the reduction is correct.

It remains to prove that the constructed preference system $(\hat{G}, \hat{\prec})$ is single-peaked and single-crossing.

Single-peaked property. Observe that all newly added vertices except for s' and t' have degree 2 in $\hat{G}$, as do u and v as well. Hence, these vertices have preferences that are trivially single-peaked with respect to any axis. To deal with s', t', and the vertices in $\hat{A} \cup \hat{B}$, it suffices to append the vertices $d^{x_n}, d^{x_{n-1}}, \ldots, d^{x_1}, s, s'$ in this order after Axis $\widetilde{A}$, and similarly, to append $d^{c_m}, d^{c_{m-1}}, \ldots, d^{c_1}, t, t'$ in this order after Axis $\widetilde{B}$. It can easily be verified that $(\hat{G}, \hat{\prec})$ is single-peaked with respect to the two axis obtained this way.

Single-crossing property. To show that $(\hat{G}, \hat{\prec})$ is single-crossing, recall the orders in Π and in Φ as defined in the proof of Theorem 6. We apply two modifications to each $\pi \in \Pi$: first, we append the vertices $d^{x_n}, d^{x_{n-1}}, \ldots, d^{x_1}, s, s'$ to π in this order; second, we fix the order of the vertices in $\hat{A}$ as $a^{c_m}, a^{c_{m-1}}, \ldots, a^{c_1}$. Similarly, we apply two modifications to each $\phi \in \Phi$: first, we append the vertices $d^{c_m}, d^{c_{m-1}}, \ldots, d^{c_1}, t, t'$ to ϕ in this order; second, we fix the order of the vertices in $\hat{B}$ as $b^{x_n}, b^{x_{n-1}}, \ldots, b^{x_1}$. It is straightforward to check that the preference list of any vertex in $D^C \cup \{t, t'\}$ can be obtained as the restriction of any $\pi \in \Pi$, while the preference list of any vertex in $D^X \cup \{s, s'\}$ can be obtained as the restriction of any $\phi \in \Phi$. Moreover, any vertex whose preference list in $\widetilde{G}$ is the restriction of some $\pi \in \Pi$ or $\phi \in \Phi$ has the same property in $\hat{G}$ with respect to the modified orders (as defined above). Therefore, $(\hat{G}, \hat{\prec})$ is single-crossing, proving the theorem. □

Corollary 8. *Given a preference system* $(G, \prec)$, *finding a popular matching in* G *with at most one blocking edge that is larger than a stable matching is* NP-*complete, even if preferences in* $(G, \prec)$ *are single-peaked and single-crossing.*

Remark 1. The NP-hardness of finding a complete popular matching with at most one blocking edge (a weaker form of Theorem 7 and Corollary 8) can also be proved using the reduction in [16, Section 5.1]. It can be verified that adding a path (s, x, y, t) to the graph constructed in the reduction presented in [16, Section 5.1], with newly introduced vertices x and y being each other's top choice, and with s ranking x, as well as t ranking y as their worst choice, the obtained graph admits a complete popular matching if and only if it admits a complete popular matching with (x, y) as the unique blocking edge, which in turn happens if and only if the input instance of 3-SAT is satisfiable. This argument is due to Telikepalli Kavitha (personal communication); the reduction we present in the proof of Theorem 7 is a combination of her ideas and our reduction for Theorem 6.

4 Algorithms for preferences admitting a master list

In this section we focus on the case when preferences admit a master list on one side. Our two simple algorithms for POPULARITY WITH FIXED INSTABILITY and MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS are presented in Section 4.1. Then in Section 4.2 we show that their running time is optimal. Finally, in Section 4.3. we investigate whether these approaches can handle somewhat less restricted instances.

4.1 Our algorithms

It is known that strict preference systems with a master list on one side admit a unique stable matching [42]; this fact is the backbone of our simple approach.

Theorem 9. *An instance* $(G, \prec, S)$ *of* POPULARITY WITH FIXED INSTABILITY *where the input graph* $G = (A, B; E)$ *is bipartite and* $(G, \prec)$ *admits a master list on one side can be solved in* $O(|E|)$ *time.*

Proof. Let M be a popular matching in G with $\mathrm{bp}_G(M) = S$. Then M is stable in the graph $G' = G - S$. Since $(G, \prec)$ admits a master list on one side, G' admits a unique stable matching M' which can be found in $O(|E|)$ time [42]. If M' is not popular in G or $\mathrm{bp}_G(M') \neq S$, we output 'No', otherwise we output M'. Finding M' in G', computing $\mathrm{bp}_G(M')$ and testing whether M' is popular in G can all be done in $O(|E|)$ time (for testing popularity, see [38]). $\square$

Theorem 10. *An instance* $(G, \prec, \omega, c, t, k)$ *of* MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS *where the input graph* $G = (A, B; E)$ *is bipartite,* $(G, \prec)$ *admits a master list on one side, and* $c(e) \geq 1$ *for all edges* $e \in E$ *can be solved in* $O(|E|^{k+1})$ *time.*

Proof. Since all edges have cost at least 1, the desired matching may admit at most k blocking edges. As observed in the proof of Theorem 9, for a given subset $S \subseteq E$ of edges, there exists at most one matching M with $\mathrm{bp}_G(M) = S$. By trying all edge sets $S \subseteq E$ of size at most k, we can check all possible solutions in time $O(|E|^{k+1})$, since we can check feasibility in linear time. $\square$

4.2   The optimality of our algorithms

As our algorithm in Theorem 9 runs in $O(|E|)$ time, which is the time needed to read the input, it clearly is as fast as possible asymptotically. We contrast Theorem 10 by showing that the running time $O(|E|^{k+1})$ cannot be improved to get an algorithm that runs in $f(k) \cdot |E|^{O(1)}$ time for some function f, i.e., an FPT algorithm with parameter k, as the problem is W[1]-hard with parameter k, even for $t = 1$.

Theorem 11. *The* Max-Utility Popular Matching with Instability Costs *problem is* NP-*complete and* W[1]-*hard with parameter* k, *even if*

- *the input graph* G *is bipartite,*
- *preferences admit a master list on both sides,*
- *the cost function is* $c \equiv 1$, *and*
- *either* (a) *the utility function is* $\omega(e) = \begin{cases} 1 & \text{if } e = f^\star, \\ 0 & \text{otherwise,} \end{cases}$ *for some* $f^\star \in E(G)$
    with $\omega(M_s) = 0$, *and the objective value is* $t = 1$, *or*
    (b) $\omega \equiv 1$ *and* $t = |M_s| + 1 = |V(G)|/2$,
    where M_s *is the unique stable matching in* G.

To prove Theorem 11, we need a different approach than the one used to prove the results of Section 3. There is a simple reason why preferences in those constructions do not admit a master list (on either side): vertices within cycles corresponding to literals or to truth assignments of a variable are cyclic (in the sense that each vertex on such a cycle prefers the "next" vertex along the cycle to the "previous" vertex on the cycle, when traversing the cycle in one direction), and hence do not admit a master list on either side.

The reduction proving Theorem 11 is from the W[1]-hard Multicolored Clique problem, and although it applies standard techniques from the literature (e.g., a similar approach is used in [34]), proving its correctness requires detailed arguments. To give the reader some intuition about the workings of the reduction, we present a short sketch, as we did with Theorem 5 as well.

Proof outline for Theorem 11. The input of the Multicolored Clique problem is a graph $G = (V, E)$ with V partitioned into sets $V_1, \dots, V_q$, and an integer parameter q, and the task is to decide whether G contains a clique of size q containing exactly one vertex from each of the sets V_i.

We introduce a vertex gadget G_i for each $i \in [q]$, and an edge gadget $G_{i,j}$ for each $\{i, j\} \subseteq [q]$ with $i < j$. Both vertex and edge gadgets will have the same underlying structure, which we present now for G_i: It contains an edge (a_v, b_v) for each $v \in V_i$, a "source" vertex s_i and a "sink" vertex t_i.[2] We connect s_i to each vertex in $\{a_v : v \in V_i\}$, and analogously, we connect t_i to each vertex in $\{b_v : v \in V_i\}$. Edge gadgets are defined analogously, with $G_{i,j}$ containing edges of the form (a_e, b_e) for each edge $e \in E$ running between V_i and V_j.

The next idea is to connect all gadgets, one after the other, threading them along a path by connecting the sink vertex of each gadget with the source vertex of the next gadget (see Figure 7 for an illustration). Preferences are defined so that the unique stable matching M_s contains all edges of the form (a_v, b_v) and (a_e, b_e), and leaves unmatched only the source vertex of the first gadget, say s_1, and the sink vertex of the last gadget, say $t_{q-1,q}$. Thus, a desired matching M will be such that $M \triangle M_s$ is a path P_M from s_1 to $t_{q-1,q}$ that traverses all gadgets. The edges of P_M used within the gadgets will therefore correspond to selecting q vertices and $\binom{q}{2}$ edges in the input graph G. Additional inter-gadget edges, running between vertex and edge gadgets,

---

[2]We only use the terms "source" and "sink" for illustration; the constructed graph is undirected.

will ensure that M is a popular matching of size $|M_s| + 1$ with $|\text{bp}(M)| \leq q + \binom{q}{2}$ exactly if the endpoints of any edge "selected" by $P_M = M_S \triangle M$ are also "selected", which in turn can happen exactly if there exists a clique of size q in G as required.

Proof of Theorem 11. We give a reduction from the W[1]-hard MULTICOLORED CLIQUE problem, parameterized by the size of the solution [27]. Let $G = (V, E)$ be the input of MULTICOLORED CLIQUE, with V partitioned into q set $V_1, \ldots, V_q$. We let $E_{i,j}$ denote the edges of G that run between V_i and V_j for some $1 \leq i < j \leq q$.

Construction. For each $i \in [q]$, we construct a vertex gadget G'_i on vertex set $A_i \cup B_i \cup \{s_i, t_i\}$ where $A_i = \{a_v : v \in V_i\}$ and $B_i = \{b_v : v \in V_i\}$. For each $v \in V_i$, vertices a_v and b_v are each other's top choices, the worst choice of a_v is s_i, and the worst choice of b_v is t_i; the gadget contains only these $3|V_i|$ edges. Similarly, for each $i, j \in [q]$ with $i < j$, we construct an edge gadget $G'_{i,j}$ on vertex set $A_{i,j} \cup B_{i,j} \cup \{s_{i,j}, t_{i,j}\}$ where $A_{i,j} = \{a_e : e \in E_{i,j}\}$ and $B_{i,j} = \{b_e : e \in E_{i,j}\}$. For each $e \in E_{i,j}$, vertices a_e and b_e are each other's top choices, the worst choice of a_e is $s_{i,j}$, and the worst choice of b_e is $t_{i,j}$; the gadget contains only these $3|E_{i,j}|$ edges. We also add vertices s_0 and t_0, connected with each other.

We next create edges that connect our gadgets along a path. For this, we need an ordering over our gadgets, so let μ be any fixed bijection from $[q + \binom{q}{2}]$ to $[q] \cup \{(i, j) \in [q] \times [q] : i < j\}$ satisfying the property that all vertex gadgets precede all edge gadgets (formally, this amounts to $\mu^{-1}(i) < \mu^{-1}(i', j')$ for any $i, i', j' \in [q]$ with $i' < j'$). For simplicity, we also assume $\mu(1) = 1$. We can now add the edge set

$$E_{TS} = \{(t_0, s_1)\} \cup \{(t_{\mu(h)}, s_{\mu(h)+1}) : h \in [q + \binom{q}{2} - 1]\}.$$

We further create a set F of *consistency edges* as follows. For each edge $e \in E$ connecting vertices $x \in V_i$ and $y \in V_j$ for some indices i and j, we connect vertex b_e with all vertices in the set $\{a_v : v \in V_i \setminus \{x\}\} \cup \{a_v : v \in V_j \setminus \{y\}\}$.
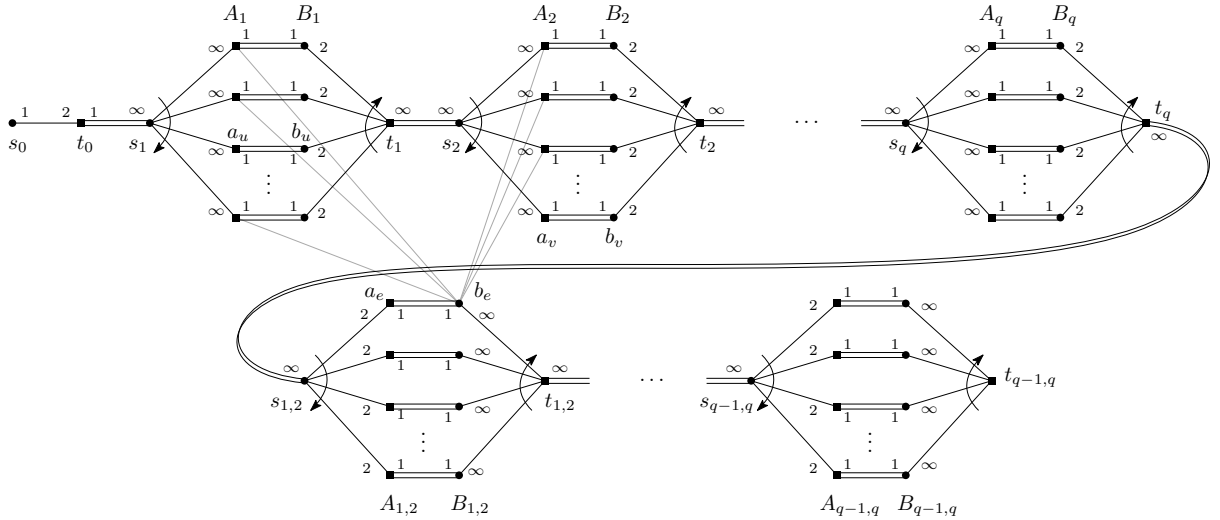

Figure 7: Illustration of the reduction in the proof of Theorem 11. Vertex gadgets are shown in the upper row, with edge gadgets below them. We assume $\mu(i) = i$ for each $i \in [q]$, as well as $\mu(q + 1) = (1, 2)$ and $\mu(q + \binom{q}{2}) = (q - 1, q)$. Double lines denote edges of M_s. The bent arrows around a given vertex indicate the preferences of the vertex, with the arrow progressing from the least-preferred neighbor towards the most-preferred neighbor. Among consistency edges only those incident to b_e are shown (in grey). The figure assumes $e = (u, v) \in E_{1,2}$.

Let G' denote the resulting graph; see Figure 7 for an illustration. We set

$$A_V = \bigcup_{i=1}^{q} A_i, \quad A_E = \bigcup_{1 \le i < j \le q} A_{i,j}, \quad S = \{s_i : i \in [q]\} \cup \{s_{i,j} : 1 \le i < j \le q\}$$

and we define the sets B_V, B_E, and T analogously. Then $A = A_V \cup A_E \cup S \cup \{s_0\}$ and $B = B_V \cup B_E \cup T \cup \{t_0\}$ are the two partitions of G'. We proceed with providing a master list over vertices in A and in B, denoted by $\mathcal{L}_A$ and $\mathcal{L}_B$, respectively. To this end, we fix an arbitrary ordering π over $V \cup E$. Then we write $\overrightarrow{A_V}$ to denote the ordering of A_V in which a_v precedes $a_{v'}$ if and only if v precedes v' according to π. By contrast, let $\overleftarrow{B_V}$ to denote the ordering of B_V in which b_v precedes $b_{v'}$ for some $v' \ne v$ if and only if v does *not* precede v' according to π. We define $\overrightarrow{A_E}$ and $\overleftarrow{B_E}$ analogously. We are now ready to define the preferences:

$$\mathcal{L}_A: \ \overrightarrow{A_E}, \ \overrightarrow{A_V}, \ S, \ s_0;$$
$$\mathcal{L}_B: \ \overleftarrow{B_V}, \ \overleftarrow{B_E}, \ T, \ t_0.$$

Observe that $M_s = \{(a_v, b_v) : v \in V\} \cup \{(a_e, b_e) : e \in E\} \cup E_{TS}$ is the unique stable matching in G'.

All edges have cost 1, and our budget is $k = q + \binom{q}{2}$. Regarding the utility function ω and the objective value t, we define two equivalent variants, as required. To show hardness for variant (a) we set ω such that the edge (s_0, t_0) has utility 1, while all other edges have utility 0, and we set $t = 1$. Note that $\omega(M_s) = 0$ indeed holds in this case. For variant (b), we set $\omega \equiv 1$ and $t = |V(G')|/2$; note that in this case $t = |M_s| + 1$. Observe that the presented reduction is a parameterized reduction with parameter k (in both variants).

We prove that a feasible popular matching exists in the constructed instance $(G', \prec, \omega, c, t, k)$ if and only if G contains a clique of size q containing a vertex from each set V_i.

Direction "$\Rightarrow$". Suppose that M is a matching that is popular and feasible for $(G', \prec, \omega, c, t, k)$.

First, let us observe that a popular matching M cannot leave a vertex v in $A_V \cup A_E \cup B_V \cup B_E$ unmatched: supposing otherwise we immediately get that $(v, M_s(v))$ blocks M, and so v cannot be unmatched due to condition (c3/i) of Observation 4, a contradiction.

Claim. ($\blacklozenge$) If $(s_{\mu(h)}, a) \in M$ for some vertex $a \in A_{\mu(h)}$, then $(M_s(a), t_{\mu(h)}) \in M$, and also $M(s_{\mu(h+1)}) \in A_{\mu(h+1)}$ unless $h = q(q+1)/2$.

Proof. We prove this first for vertex gadgets, i.e., for $h \in [q]$. Suppose $(s_{\mu(h)}, a) \in M$ for some vertex $a \in A_{\mu(h)}$ with $h \in [q]$. Let $b = M_s(a)$. Since a and b are each other's top choice, $(a, b) \in \mathrm{bp}(M)$ and so by condition (c3/i) of Observation 4 we know that b must be matched in M. Since $G'_{\mu(h)}$ is a vertex gadget, $\delta_{G'}(b) = 2$. Hence, we get $(b, t_{\mu(h)}) \in M$. Using again condition (c3/i), we also know that $s_{\mu(h+1)}$ must be matched in M, and since it clearly cannot be matched to $t_{\mu(h)}$, claim ($\blacklozenge$) follows in this case.

Before proving claim ($\blacklozenge$) for edge gadgets, let us show first that M contains no consistency edges. By the feasibility of M we know $(s_0, t_0) \in M$, and hence s_1 must be matched by M, as otherwise (t_0, s_1) would be a blocking edge of M with an unmatched endpoint, contradicting condition (c3/i) of Observation 4. Thus, $M(s_1) \in A_1$. Moreover, by our choice of μ, all vertex gadgets precede all edge gadgets, and thus repeatedly applying claim ($\blacklozenge$) for vertex gadgets (proved in the previous paragraph) implies that for each $i \in [q]$ there exists a unique vertex $v \in V_i$ such that (s_i, a_v) and (b_v, t_i) are both in M, with $(a_v, b_v) \in \mathrm{bp}(M)$; we denote this vertex v by v_i. Assuming that some consistency edge $(a_v, b_e) \in F$ is contained in M, it is clear that $v \ne v_i$ for any $i \in [q]$. However, then b_v cannot be matched to any vertex by M, a contradiction.

24

Suppose now that $(s_{\mu(h)}, a) \in M$ for some vertex $a \in A_{\mu(h)}$ in an edge gadget $G'_{\mu(h)}$. Again, let $\breve{b} = M_s(a)$. Then b must be matched in M by condition (c3/i) of Observation 4. Note that $\delta_{G'-F}(b) = 2$, and hence from $M \cap F = \emptyset$ we get $(b, t_{\mu(h)}) \in M$. Thus, if $s_{\mu(h+1)}$ exists, then it must also be matched in M by condition (c3/i), and can only be matched to some vertex in $A_{\mu(h+1)}$, proving claim ($\blacklozenge$). $\qquad\square$

Therefore, for any $i, j \in [q]$ with $i < j$ there exists a unique edge $e \in E_{i,j}$ such that $(s_{i,j}, a_e)$ and $(b_e, t_{i,j})$ are both in M, with $(a_e, b_e) \in \mathrm{bp}(M)$; let $e_{i,j}$ denote this edge e.

As a consequence, each gadget contains an edge of M_s that blocks M. By $k = q + \binom{q}{2}$ we get that no consistency edge can block M without exceeding the budget. We claim that each edge $e_{i,j}$ connects v_i and v_j. Indeed, if $e_{i,j}$ is not incident to v_i, then a_{v_i} and $b_{e_{i,j}}$ are connected in G' by a consistency edge, and they form a blocking edge for M (as both of them are matched to their least favorite neighbor in M), a contradiction. By symmetry, $e_{i,j}$ is incident to v_j as well, proving our claim. Therefore, vertices $v_1, \ldots, v_q$ form a clique in G, with $v_i \in V_i$ for each $i \in [q]$ as desired.

Direction "$\Leftarrow$". Suppose that vertices $v_i \in V_i$, $i \in [q]$, form a clique in G; let us define $K = G[\{v_1, \ldots, v_q\}]$. We define a matching M as follows.

$$
\begin{aligned}
M \;=\; & \{(s_0, t_0)\} \cup \{(a_v, b_v) : v \in V \setminus V(K)\} \cup \{(s_i, a_{v_i}), (b_{v_i}, t_i) : i \in [q]\} \\
& \cup \{(a_e, b_e) : e \in E \setminus E(K)\} \cup \{(s_{i,j}, a_{(v_i, v_j)}), (b_{(v_i, v_j)}, t_{i,j}) : i, j \in [q], i < j\}.
\end{aligned}
$$

Observe that M is a complete matching, and the set of edges blocking M is

$$
\mathrm{bp}(M) = \{(a_{v_i}, b_{v_i}) : i \in [q]\} \cup \{(a_{(v_i, v_j)}, b_{(v_i, v_j)}) : i, j \in [q], i < j\},
$$

since all vertices in $A_V \cup B_V \cup A_E \cup B_E$ except those in $V(\mathrm{bp}(M))$ are matched to their top choice in M; note that since K is a clique, the definition of G' implies that no consistency edge has both of its endpoints in $V(\mathrm{bp}(M))$, and therefore no consistency edge blocks M. This shows that the edges blocking M have total cost exactly $k = |\mathrm{bp}(M)|$, and thus M is feasible. It remains to show that M is popular in G'.

For the sake of contradiction, suppose that M' is more popular than M. Note that s_0 obtains her top choice in M, and t_0 can only be better off in M' if s_0 is worse off in M'. Hence, as all remaining vertices in G' belong to a vertex or an edge gadget, there must exist a vertex or an edge gadget in which M' beats M, i.e., where more vertices prefer M' to M than vice versa. Suppose that G'_i is such a gadget; the same argument works for edge gadgets. Note that the only vertices that do not (necessarily) obtain their top choice in M are the endpoints of the edges (s_i, a_{v_i}) and (b_{v_i}, t_i), both in M. Thus, either s_i prefers M' to M, or t_i prefers M' to M, or both. Furthermore, it also follows that at most four vertices in G'_i can be better off in M' when compared to M, and so at most three vertices of G'_i may be better off in M when compared to M'. We distinguish between two cases:

- Case A: both s_i and t_i prefer M' to M. Then $M'(s_i) = a_x$ and $M'(t_i) = b_y$ for some x and y with $v_i \notin \{x, y\}$. Since $a_{v_i} \prec_{s_i} a_x$, we get that x precedes v_i in π. By contrast, since $b_{v_i} \prec_{t_i} b_y$ we get that y does *not* precede v_i in π. Hence, $x \neq y$. This implies that vertices a_x, b_x, a_y, and b_y are four vertices that prefer M to M', a contradiction.
- Case B: exactly one of s_i and t_i prefer M' to M, say s_i (the case for t_i is symmetric). Let $M'(s_i) = a_x$ for some $x \neq v_i$. If $M'(t_i) = M(t_i)$, then the only vertices preferring M' to M can be s_i and a_{v_i}, but both a_x and b_x prefer M to M', so M' cannot beat M in G'_i, a contradiction. Otherwise, $M'(t_i) \neq M(t_i)$ and, by our assumption, t_i prefers M to M'. Then a_x, b_x, and t_i are three vertices in G'_i who prefer M to M', while at most three vertices in G'_i (namely, s_i, a_{v_i}, and b_{v_i}) may prefer M' to M, a contradiction.

This proves that M is indeed popular, as required, finishing our proof. $\qquad\square$

4.3 The extendibility of our algorithms

We next investigate whether the conditions of Theorems 9 and 10 can be lifted without leaving the field of polynomially tractable problems.

4.3.1 Free edges.

One may wonder if the restriction in Theorem 10 that all edge costs are at least 1 is necessary, or our algorithm can be extended to accommodate edges of cost 0. Such edges are called *free edges* in the literature, and a matching whose blocking edges are all free edges is called a *socially stable matching*. It is known that deciding whether there exists a complete socially stable matching in a bipartite instance with master lists on both sides is NP-complete [48, Theorem 5.3.4]; however, this does not imply the following theorem where we prove that it is NP-hard to find a complete *popular* socially stable matching. The proof of Theorem 12 is a slight modification of the reduction proving Theorem 11.

Theorem 12. *The* Max-Utility Popular Matching with Instability Costs *problem is* NP*-complete even if*
- *the cost function c is binary,*
- *the budget is $k = 0$, and*
- *all other restrictions of Theorem 11 hold.*

Proof. We prove the theorem by slightly modifying the reduction from Multicolored Clique given in the proof of Theorem 11; we re-use all definitions from that proof. Recall the instance $I = (G', \prec, \omega, c, t, k = \binom{q}{2} + q)$ of Max-Utility Popular Matching with Instability Costs constructed there. We define a modified instance $I' = (G', \prec, \omega, c', t, k' = 0)$ by setting $c'(e) = 1$ if e is a consistency edge, and setting $c'(e) = 0$ otherwise. Note that our modified reduction is a polynomial-time reduction.

We claim that there exists a feasible popular matching in I' if and only if the input graph G contains a clique of size q.

Direction "⇒". Suppose that M is a popular matching feasible for I'. The same arguments presented in the proof of Theorem 11 can be applied to define a clique in G, except for the argument to prove that M cannot be blocked by any consistency edge—the only place where properties of the original cost function c and budget k were used. In the case of I', however, this follows immediately from the fact that $c'(e) = 1$ for any consistency edge e and our budget is $k' = 0$.

Direction "⇐". Given a clique in G, the corresponding matching M defined as in the proof of Theorem 11 has the property that $\mathrm{bp}(M) \cap F = \emptyset$, so the cost of M with respect to the modified cost function c' is zero. Hence, M is feasible in I' as well. □

4.3.2 Ties in the master list.

Another possible direction to generalize Theorems 9 and 10 would be to extend their results to the case where preferences are not necessarily strict but may include ties. We define the Max-Utility Popular Matching with Instability Costs and Ties problem the same way as its strict variant, with the only difference that the input preference system may not be strict; we define Popularity with Fixed Instability and Ties analogously.

Our next result shows that there is no hope that the algorithm of Theorem 9 can be extended to the case where we allow ties in the preference lists, even if we further require severe restrictions on the input. The proof of Theorem 13 is based on the reduction proving Theorem 6, and its main idea is to circumvent the problem of cyclic preferences in the construction by using ties.

Theorem 13. *The* MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS AND TIES *problem is* NP-*complete even if*

- *the input graph $G = (A, B; E)$ is bipartite and $\Delta_G = 3$,*
- *preferences on both sides admit a master list,*
- *the cost function is $c \equiv 1$,*
- *the budget is $k = 1$,*
- *the utility function is $\omega(e) = \begin{cases} 1 & \text{if } e = f^\star, \\ 0 & \text{otherwise,} \end{cases}$ for some $f^\star \in E(G)$, and*
- *the objective value is $t = 1$.*

The POPULARITY WITH FIXED INSTABILITY AND TIES *problem is* NP-*complete even if*

- *the input graph G is bipartite and $\Delta_G = 3$,*
- *preferences on both sides admit a master list, and*
- *$|S| = 1$ for the set S of blocking edges.*

Proof. Again, we are going to present a modification of the reduction given in the proof of Theorem 6, re-using all definitions there. One can observe that there is a simple reason why preferences in the graph $\widetilde{G}$ created in that reduction do not admit a master list (on either side): vertices within a cycle $C^{c_j,\ell}$, $j \in [m]$ and $\ell \in [3]$, as well as within a cycle $\widetilde{C}^{x_i,\lambda}$, $i \in [n]$ and $\lambda \in \{\mathtt{t},\mathtt{f}\}$, are cyclic (in the sense that each vertex on such a cycle prefers the "next" vertex along the cycle to the "previous" vertex on the cycle, when traversing the cycle in one direction), and hence do not admit a master list on either side. We now present a modification that circumvents this problem with the help of ties, by simply replacing an edge in each of these cycles with a path of length 5 (i.e., making each of them longer by four edges).

Construction. For each $j \in [m]$ and $\ell \in [3]$ we subdivide the edge $(a_1^{c_j,\ell}, b_0^{c_j,\ell})$ of $\widetilde{G}$ with newly introduced vertices $b_2^{c_j,\ell}$, $a_2^{c_j,\ell}$, $b_3^{c_j,\ell}$, and $a_3^{c_j,\ell}$, in this order. Similarly, for each $i \in [n]$ and $\lambda \in \{\mathtt{t},\mathtt{f}\}$ we subdivide the edge $(b_3^{x_i,\lambda}, a_0^{x_i,\lambda})$ with newly introduced vertices $a_4^{x_i,\lambda}$, $b_4^{x_i,\lambda}$, $a_5^{x_i,\lambda}$, and $b_5^{x_i,\lambda}$. This way, for any cycle C of the form $\widetilde{C}^{x_i,\lambda}$ or $C^{c_j,\ell}$ in $\widetilde{G}$, we have created a corresponding cycle C^{ties} in the obtained graph $\widetilde{G}^{\text{ties}}$. We will call such cycles *base cycles* (both in $\widetilde{G}$ and in $\widetilde{G}^{\text{ties}}$).

We give the preferences of the newly introduced vertices and their neighbors below (see Figure 8); all other preferences remain unchanged. Ties are denoted by angle brackets $\langle \cdot \rangle$.

$$
\begin{array}{ll}
a_1^{c_j,\ell}: \langle b_1^{c_j,\ell}, b_2^{c_j,\ell} \rangle & \text{where } j \in [m] \text{ and } \ell \in [3]; \\
b_2^{c_j,\ell}: \langle a_1^{c_j,\ell}, a_2^{c_j,\ell} \rangle & \text{where } j \in [m] \text{ and } \ell \in [3]; \\
a_2^{c_j,\ell}: b_2^{c_j,\ell}, b_3^{c_j,\ell} & \text{where } j \in [m] \text{ and } \ell \in [3]; \\
b_3^{c_j,\ell}: a_2^{c_j,\ell}, a_3^{c_j,\ell} & \text{where } j \in [m] \text{ and } \ell \in [3]; \\
a_3^{c_j,\ell}: \langle b_0^{c_j,\ell}, b_3^{c_j,\ell} \rangle & \text{where } j \in [m] \text{ and } \ell \in [3]; \\
b_0^{c_j,\ell}: \langle a_0^{c_j,\ell}, a_3^{c_j,\ell} \rangle, a_0^{c_j,\ell+1} & \text{where } j \in [m] \text{ and } \ell \in [2]; \\
b_0^{c_j,3}: \langle a_0^{c_j,3}, a_3^{c_j,3} \rangle, a^{c_j} & \text{where } j \in [m]; \\
b_3^{x_i,\lambda}: \langle a_3^{x_i,\lambda}, a_4^{x_i,\lambda} \rangle & \text{where } i \in [n] \text{ and } \lambda \in \{\mathtt{t},\mathtt{f}\}; \\
a_4^{x_i,\lambda}: \langle b_3^{x_i,\lambda}, b_4^{x_i,\lambda} \rangle & \text{where } i \in [n] \text{ and } \lambda \in \{\mathtt{t},\mathtt{f}\}; \\
b_4^{x_i,\lambda}: a_4^{x_i,\lambda}, a_5^{x_i,\lambda} & \text{where } i \in [n] \text{ and } \lambda \in \{\mathtt{t},\mathtt{f}\}; \\
a_5^{x_i,\lambda}: b_4^{x_i,\lambda}, b_5^{x_i,\lambda} & \text{where } i \in [n] \text{ and } \lambda \in \{\mathtt{t},\mathtt{f}\}; \\
b_5^{x_i,\lambda}: \langle a_0^{x_i,\lambda}, a_5^{x_i,\lambda} \rangle & \text{where } i \in [n] \text{ and } \lambda \in \{\mathtt{t},\mathtt{f}\}; \\
a_0^{x_i,\mathtt{t}}: \langle b_0^{x_i,\mathtt{t}}, b_5^{x_i,\mathtt{t}} \rangle, b_0^{x_i,\mathtt{f}} & \text{where } i \in [n]; \\
a_0^{x_i,\mathtt{f}}: \langle b_0^{x_i,\mathtt{f}}, b_5^{x_i,\mathtt{f}} \rangle, b^{x_i} & \text{where } i \in [n].
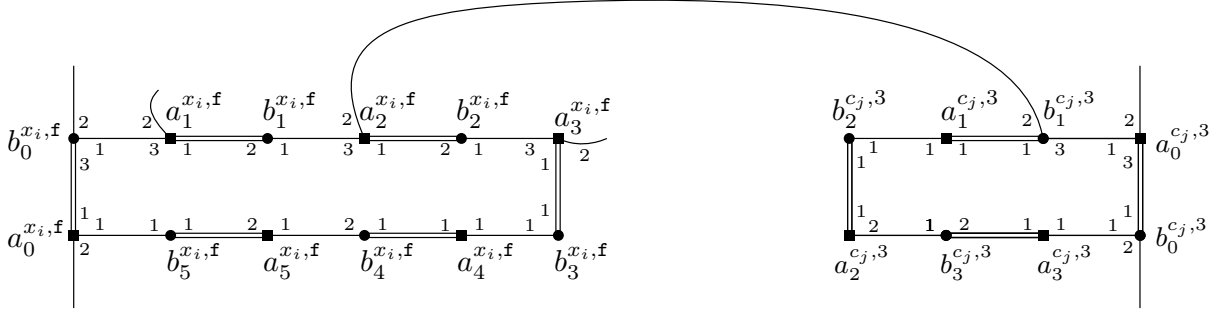\end{array}
$$

Figure 8: Illustration of the reduction in the proof of Theorem 13, depicting how the graph $\widetilde{G}$ is modified to obtain graph $\widetilde{G}^{\text{ties}}$. The figure assumes that the third literal in c_j is x_i as a positive literal. Edges leaving a given vertex a that are labelled with the same rank lead to vertices that are equally preferred by a and hence are contained in a tie in a's preference list.

We keep all remaining parameters for both problems unchanged: $S = \{e^\star\}$ for POPULARITY WITH FIXED INSTABILITY, and for MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS we set c, k, ω and t as required in the statement of the theorem, with $f^\star = (u, v_1')$ being the only edge with utility 1.

Further definitions. We extend the matching $\widetilde{M}_0$ in the straightforward way into a matching $\widetilde{M}_0^{\text{ties}}$: for any $i \in [n]$ and $\lambda \in \{\mathtt{t}, \mathtt{f}\}$, we let $\widetilde{M}_0^{\text{ties}}(a_h^{x_i,\lambda}) = b_h^{x_i,\lambda}$ for each $h \in \{0, 1, \ldots, 5\}$, and similarly, for each $j \in [m]$ and $\ell \in [3]$, we let $\widetilde{M}_0^{\text{ties}}(a_h^{c_j,\ell}) = b_h^{c_j,\ell}$ for each $h \in \{0, 1, 2, 3\}$; we set $\widetilde{M}_0^{\text{ties}}(a) = \widetilde{M}_0(a)$ for each remaining vertex a.

Recall from the proof of Theorem 6 that any popular matching $\widetilde{M}$ in $\widetilde{G}$ with $\text{bp}(\widetilde{M}) = \{e^\star\}$ has the following property: for any base cycle C either $\widetilde{M}$ contains $C \cap M_0$, or $\widetilde{M}$ contains $C \setminus M_0$. Based on this fact, we can define a matching $\widetilde{M}^{\text{ties}}$ in $\widetilde{G}^{\text{ties}}$ corresponding to $\widetilde{M}$ as follows: for each base cycle C in $\widetilde{G}$, if $\widetilde{M} \supseteq \widetilde{M}_0 \cap C$, then we add the edges $C^{\text{ties}} \cap \widetilde{M}_0^{\text{ties}}$ to $\widetilde{M}^{\text{ties}}$, otherwise we add the edges $C^{\text{ties}} \setminus \widetilde{M}_0^{\text{ties}}$; we further add all edges of $\widetilde{M}_0^{\text{ties}}$ not incident to any base cycle to $\widetilde{M}^{\text{ties}}$.

Correctness. It is easy to see that if $\widetilde{M}$ is a popular matching in $\widetilde{G}$ with $\text{bp}(\widetilde{M}) = \{e^\star\}$, then $\widetilde{M}^{\text{ties}}$ is a popular matching in $\widetilde{G}^{\text{ties}}$ with $\text{bp}(\widetilde{M}^{\text{ties}}) = \{e^\star\}$. We are going to prove that the converse is true as well, i.e., any popular matching in $\widetilde{G}^{\text{ties}}$ blocked only by $e^\star$ is of the form $\widetilde{M}^{\text{ties}}$ for some popular matching $\widetilde{M}$ in $\widetilde{G}$ with $\text{bp}(\widetilde{M}) = \{e^\star\}$. Notice that this suffices to prove that our modified reduction is correct (for both problems).

So let M be a popular matching in $\widetilde{G}^{\text{ties}}$ blocked only by $e^\star$. It is clear that $(v, u_1') \in M$. Using for $i = 1, 2, \ldots, n-1$ that the edge (v_i, u_{i+1}') cannot block M, we get that $(v_i, u_{i+1}') \in M$ for each $i \in [n-1]$. In particular, no edge $(u_i', b_0^{x_i,\mathtt{t}})$ is in M. This implies that $(a_0^{x_i,\mathtt{t}}, b_0^{x_i,\mathtt{f}}) \notin M$: indeed, $(a_0^{x_i,\mathtt{t}}, b_0^{x_i,\mathtt{f}}) \in M$ would imply $(a_5^{x_i,\mathtt{t}}, b_5^{x_i,\mathtt{t}}) \in M$ since $(a_0^{x_i,\mathtt{t}}, b_5^{x_i,\mathtt{t}})$ cannot block M, and applying the same argument iteratively it follows that $(a_h^{x_i,\mathtt{t}}, b_h^{x_i,\mathtt{t}}) \in M$ for each $h \in [5]$. That would, however, leave $b_0^{x_i,\mathtt{t}}$ unmatched in M and hence $(b_0^{x_i,\mathtt{t}}, a_0^{x_i,\mathtt{t}})$ would block M, a contradiction. Therefore we know $(a_0^{x_i,\mathtt{t}}, b_0^{x_i,\mathtt{f}}) \notin M$. Using the same argument once more, we also get that $(a_0^{x_i,\mathtt{f}}, b^{x_i}) \notin M$.

Note that analogous arguments can be used to show that M contains (u, v_1') together with all edges (u_j, v_{j+1}') for $j \in [m-1]$, and that M does not match any vertex of the form $b_0^{c_j,\ell}$ to its worst choice. To prove our claim, it therefore remains to show that M cannot contain any consistency edge.
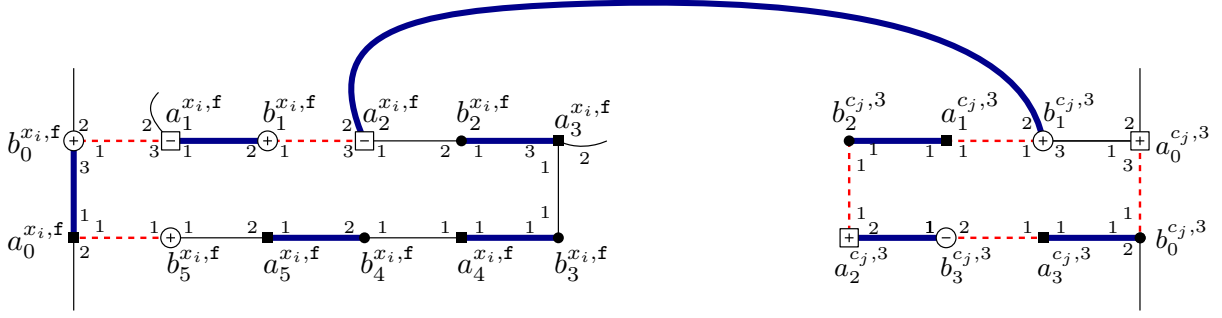
Figure 9: Illustration of the matching M' in the proof of Theorem 13 in the case where the matching M contains the consistency edge $(a_2^{x_i,\mathtt{f}}, b_1^{c_j,3})$. The figure assumes $(a_0^{x_i,\mathtt{f}}, b_0^{x_i,\mathtt{f}}) \in M$ but $(a_0^{c_j,3}, b_0^{c_j,3}) \notin M$. The edges of M are depicted with bold blue lines, while edges of $M' \setminus M$ are dashed red lines. Vertices that prefer M' to M have been marked by a plus sign, while those preferring M to M' are marked by a minus sign.

Assume for the sake of contradiction that M contains some consistency edge $(a_h^{x_i,\lambda}, b_1^{c_j,\ell})$. We will define a matching M' so that M' is more popular than M. Since $(a_h^{x_i,\lambda}, b_h^{x_i,\lambda})$ cannot block M, we know $(a_{h+1}^{x_i,\lambda}, b_h^{x_i,\lambda}) \in M$, and applying the same argument iteratively we obtain that $(a_{h'+1}^{x_i,\lambda}, b_{h'}^{x_i,\lambda}) \in M$ for each $h' = h, h+1, \ldots, 3$. We also get $(a_5^{x_i,\lambda}, b_4^{x_i,\lambda}) \in M$, as otherwise $(a_5^{x_i,\lambda}, b_4^{x_i,\lambda})$ would block M. By contrast, if $h > 1$, then since $(a_{h-1}^{x_i,\lambda}, b_{h-1}^{x_i,\lambda})$ cannot block M, it must be contained in M; arguing iteratively again, we obtain that $(a_{h'}^{x_i,\lambda}, b_{h'}^{x_i,\lambda}) \in M$ for each $h' \in [h-1]$. Thus, M assigns either $b_0^{x_i,\lambda}$ or $b_5^{x_i,\lambda}$ to $a_0^{x_i,\lambda}$, leaving the other vertex unmatched. Now, we let M' contain the edges $\{(a_h^{x_i,\lambda}, b_h^{x_i,\lambda}) : h \in \{0, 1, \ldots, 5\}\}$ if $(a_0^{x_i,\lambda}, b_0^{x_i,\lambda}) \notin M$, otherwise we let M' to contain the edges $\{(a_{h+1 \bmod 6}^{x_i,\lambda}, b_h^{x_i,\lambda}) : h \in \{0, 1, \ldots, 5\}\}$. We use the same reasoning for the cycle $C^{c_j,\ell}$ and define M' analogously on the vertices of $C^{c_j,\ell}$. For each remaining vertex a we let $M'(a) = M(a)$. It is not hard to see by simply checking all vertices in $V(M \triangle M')$ that there are at least two more vertices preferring M' to M than vice versa (see Figure 9), showing that M' is indeed more popular than M, a contradiction.

Master lists. To finish the proof of our theorem, it remains to show that the reduction has the promised properties, in particular that preferences on both sides admit a master list. To this end, for $h = 1, 2, 3$ we define I_h as the set of those pairs (c_j, ℓ) where the ℓ-th literal in clause c_j is the h-th occurrence of some variable in the input formula. We define the two master lists as follows.

$\mathcal{L}_A$: $u, \left\{\left\langle a_1^{c_j,\ell}, a_2^{c_j,\ell}\right\rangle\right\}_{j\in[m],\ell\in[3]}, \left\{\left\langle a_3^{x_i,\lambda}, a_4^{x_i,\lambda}\right\rangle\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}}, \left\{a_2^{x_i,\lambda}\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}}, \left\{a_1^{x_i,\lambda}\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}},$

$u_1', u_2', \ldots, u_n', \left\langle a_0^{c_1,1}, a_3^{c_1,1}\right\rangle, u_1, \left\langle a_0^{c_2,1}, a_3^{c_2,1}\right\rangle, u_2, \ldots, u_{m-1}, \left\langle a_0^{c_m,1}, a_3^{c_m,1}\right\rangle, \left\{\left\langle a_0^{c_j,2}, a_3^{c_j,2}\right\rangle\right\}_{j\in[m]},$

$\left\{\left\langle a_0^{c_j,3}, a_3^{c_j,3}\right\rangle\right\}_{j\in[m]}, \left\{a^{c_j}\right\}_{j\in[m]}, \left\{\left\langle a_0^{x_i,\mathtt{t}}, a_5^{x_i,\mathtt{t}}\right\rangle\right\}_{i\in[n]}, \left\{\left\langle a_0^{x_i,\mathtt{f}}, a_5^{x_i,\mathtt{f}}\right\rangle\right\}_{i\in[n]}.$

$\mathcal{L}_B$: $v, \left\{\left\langle b_3^{x_i,\lambda}, b_4^{x_i,\lambda}\right\rangle\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}}, \left\{\left\langle b_1^{c_j,\ell}, b_2^{c_j,\ell}\right\rangle\right\}_{(c_j,\ell)\in I_3}, \left\{b_2^{x_i,\lambda}\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}}, \left\{\left\langle b_1^{c_j,\ell}, b_2^{c_j,\ell}\right\rangle\right\}_{(c_j,\ell)\in I_2},$

$\left\{b_1^{x_i,\lambda}\right\}_{i\in[n],\lambda\in\{\mathtt{t},\mathtt{f}\}}, \left\{\left\langle b_1^{c_j,\ell}, b_2^{c_j,\ell}\right\rangle\right\}_{(c_j,\ell)\in I_1}, \left\langle b_0^{x_1,\mathtt{t}}, b_5^{x_1,\mathtt{t}}\right\rangle, v_1, \left\langle b_0^{x_2,\mathtt{t}}, b_5^{x_2,\mathtt{t}}\right\rangle, v_2, \ldots, v_{n-1},$

$\left\langle b_0^{x_n,\mathtt{t}}, b_5^{x_n,\mathtt{t}}\right\rangle, \left\{\left\langle b_0^{x_i,\mathtt{f}}, b_5^{x_i,\mathtt{f}}\right\rangle\right\}_{i\in[n]}, \left\{b^{x_i}\right\}_{i\in[n]}, v_1', v_2', \ldots, v_m', \left\{\left\langle b_0^{c_j,1}, b_3^{c_j,1}\right\rangle\right\}_{j\in[m]},$

$$\left\{\langle b_0^{c_j,2}, b_3^{c_j,2}\rangle\right\}_{j\in[m]}, \left\{\langle b_0^{c_j,3}, b_3^{c_j,3}\rangle\right\}_{j\in[m]}.$$

It is not hard (though tedious) to verify that preferences in the constructed instance indeed admit the master lists given above. $\qquad\square$

5 Pareto-optimal matchings with bounded instability

In this section we shift our attention to the following problem.

MAX-UTILITY PARETO-OPTIMAL MATCHING WITH INSTABILITY COSTS:

Input: A strict preference system $(G, \prec)$, 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 $k \in \mathbb{N}$.

Question: Is there a Pareto-optimal matching in G whose utility is at least t and whose blocking edges have total cost at most k?

A natural approach to solve this problem is to guess the set S of blocking edges in G, and find a stable matching M of maximum utility in the graph $G - S$. But even though M is Pareto-optimal in $G - S$, it may not be Pareto-optimal in G, as Example 2 demonstrates.

Example 2. Instead of a formal definition, we set the structure and preferences of our example as depicted in Figure 10. Assume that $k = 2$ and we are looking for a matching that is blocked by exactly the edges contained in $S = \{(a_1, b_2), (a_2, b_1)\}$. Assume that edges in S have utility 1, while all other edges have utility 2. There are two stable matchings in $G - S$, both of them having the same utility. Observe that the matching $M_a = \{(a_i, b_i) : i \in [3]\}$ highlighted in part (a) of Figure 10 is Pareto-optimal in $G - S$; however, it is not Pareto-optimal in G: switching edges along the cycle (a_1, b_1, a_2, b_2) yields a matching that is a Pareto-improvement for M_a, but has less utility than M_a. By contrast, the matching $M_b = \{(a_1, b_1), (a_2, b_3), (a_3, b_2)\}$ highlighted in part (b) of Figure 10 is Pareto-optimal in $G - S$ as well as in G.
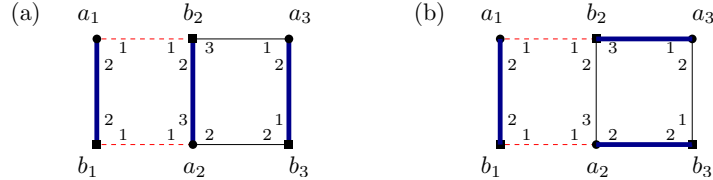

Figure 10: An example illustrating why the simplest approach for finding a Pareto-optimal matching with at most k blocking edges and maximal utility does not work. Bold blue edges denote the matching M_a and M_b in parts (a) and (b) of the figure, respectively, while dashed red edges denote blocking edges.

In Section 5.1 we present our algorithm that solves MAX-UTILITY PARETO-OPTIMAL MATCHING WITH INSTABILITY COSTS, and then prove that its running time is asymptotically optimal in Section 5.2.

5.1 Our algorithm

Let M be a solution for our input instance $I = (G, \prec, \omega, c, t, k)$. Consider the set F_M of edges that run between vertices in $V(\mathrm{bp}_G(M))$ but do *not* belong to M. We will refer to the pair $(\mathrm{bp}_G(M), F_M)$ as the *hint* for M.

Let $\mathcal{H}_I$ denote the set of all pairs (S, F) where $S \subseteq E$ with $|S| \le c(S) \le k$, and F is a subset of the edges in $G[V(S)] - S$ such that $G[V(S)] - (S \cup F)$ is a matching. Our algorithm iterates over each $H \in \mathcal{H}_I$, and searches for a matching M_H whose hint is H. With each hint H, we associate a weight function $w_H : E \setminus S \to \mathbb{N}$ as follows:

$$w_H(e) = \begin{cases} 0 & \text{if } e \in F, \\ \omega(e) + w_0 & \text{if } e \in E \setminus (S \cup F), \end{cases}$$

where w_0 is large enough to ensure that any maximum-weight stable matching in $G - S$ is disjoint from F; setting $w_0 = |M_s| \cdot \max_{e \in E} \{\omega(e)\}$ suffices for our purpose, where M_s is any stable matching in $G - S$.

Algorithm 1   Input: $I = (G, \prec, \omega, c, t, k)$.

1: **for all** $H = (S, F) \in \mathcal{H}_I$ **do**
2:      Compute a maximum-weight stable matching M_H w.r.t. w_H in $G - S$.
3:        **if** M_H is feasible and Pareto-optimal in G **then return** M_H.
4: **return** "No feasible Pareto-optimal matching exists for I."

Theorem 14. *Algorithm 1 solves an instance $(G, \prec, \omega, c, t, k)$ of* MAX-UTILITY PARETO-OPTIMAL MATCHING WITH INSTABILITY COSTS *where $G = (V, E)$ is bipartite and $c(e) \ge 1$ for all edges $e \in E$ in $2^{O(k \log k)} |E|^{k+2}$ time.*

Proof. Note that whenever Algorithm 1 outputs a matching, it is clearly Pareto-optimal and feasible in G (since it checks both). So it remains to prove that whenever there exists a feasible Pareto-optimal matching M in G, Algorithm 1 finds one. Let $H = (S_M, F_M)$ be the hint of M; clearly, $H \in \mathcal{H}_I$. We claim that M_H is also feasible and Pareto-optimal in G (though $M \ne M_H$ is possible).

First, since M_H is stable in $G - S_M$, we get $\mathrm{bp}_G(M_H) \subseteq \mathrm{bp}_G(M) = S_M$ and therefore $c(\mathrm{bp}_G(M_H)) \le c(\mathrm{bp}_G(M)) \le k$.

Second, to see $\omega(M_H) \ge \omega(M) \ge t$, recall that $S_M = \mathrm{bp}(M)$ by definition, so M is stable in $G - S_M$. As M_H is a maximum-weight stable matching in $G - S_M$ under the weight function w_H, we get that $w_H(M_H) \ge w_H(M)$. Since M has weight $|M_s| \cdot w_0 + \omega(M)$, by our choice of w_0 it follows that M_H must also contain $|M_s|$ edges whose weight (according to w_H) is at least w_0, i.e., $|M_s|$ edges in $E \setminus (S_M \cup F_M)$. Since $|M_s| = |M| = |M_H|$, this implies $M_H \cap F_M = \emptyset$. Thus, we get

$$w_H(M_H) = |M_s| \cdot w_0 + \omega(M_H) \ge w_H(M) = |M_s| \cdot w_0 + \omega(M),$$

implying $\omega(M_H) \ge \omega(M)$. Hence, M_H is feasible.

Third, we claim that M_H is Pareto-optimal in G. Let us assume otherwise for the sake of contradiction. Then there exists a matching M' in G that is a Pareto-improvement over M_H. By the definition of Pareto-improvements, any edge in $M' \setminus M_H$ blocks M_H, so $M' \setminus M_H \subseteq S_M$. Let $D = M' \triangle M_H$. Note that D is the disjoint union of cycles and paths whose endpoints are both unmatched in M_H but matched in M'. This implies that each edge in $D \cap M_H$ connects two vertices incident to some edge in $D \cap M' \subseteq M' \setminus M_H \subseteq S_M$, i.e., edges of $D \cap M_H$ are in $G[V(S_M)]$. Recall that every edge of $G[V(S_M)]$ that is not in $S_M \cup F_M$ is in M (by the definition of F_M). Therefore, as $M_H \cap F_M = \emptyset$ and M_H is a matching in $G - S_M$, we obtain $D \cap M_H \subseteq M$.

We claim that $M \triangle D$ is a matching that is a Pareto-improvement over M in G, contradicting the Pareto-optimality of M. That $M \triangle D$ is a matching follows from the facts that $D \cap M_H \subseteq M$,

that $D \setminus M_H \subseteq S_M$ is disjoint from M, and that M_H and M match the same set of vertices in G (as they are both stable in $G - S_M$). To see that $M \triangle D$ is a Pareto-improvement over M, it suffices to observe that all edges in $D \setminus M \subseteq S_M = \mathsf{bp}_G(M)$ block M.

Hence, we can conclude that Algorithm 1 will find at least one matching, namely M_H, that is feasible and Pareto-optimal in G.

Regarding the running time, first observe that $|\mathcal{H}_I| \leq |E|^k \cdot k!$ because there are at most $|E|^k$ ways to select some $S \subseteq E$ of size at most k, and given S, there are at most $k!$ possibilities to select F so that $(S, F) \in \mathcal{H}_I$. The latter follows from the observation that selecting F from the edges of $G[V(S)] - S$ is equivalent to selecting the remaining edges of $G[V(S)] - S$, required to form a matching; note that there are at most $k!$ matchings in $G[V(S)] - S$.

Given some hint $H = (S, F) \in \mathcal{H}_I$, we use the algorithm of Irving, Leather, and Gusfield [41] to find a maximum-weight stable matching in $G - S$ with respect to w_H. Their method constructs a flow network based on the so-called rotation digraph of $G - S$; this network has $N = O(|E|)$ vertices and $M = O(|E|)$ arcs, as can be noted by inspecting the proof of Lemma 3.3.2 in [36]. A detailed analysis of Irving et al.'s algorithm [14] shows that their method runs in $O(NM)$ time plus the time necessary for computing a maximum flow in the constructed network. The latter can be done in $O(NM)$ time as well, applying Orlin's algorithm [54] (not yet available when [41] and [36] were published). Thus, computing M_H takes $O(NM) = O(|E|^2)$ time, implying that the total running time of Algorithm 1 is $k!O(|E|^{k+2}) = 2^{O(k \log k)}|E|^{k+2}$. $\qquad\square$

5.2 The optimality of our algorithm

Theorem 15 generalizes a result in [34] which shows that finding a matching in a bipartite graph that is larger by t' than a stable matching and has at most k blocking edges is W[1]-hard with parameter $t' + k$; we show that the parameterized hardness holds even if t' is a constant, namely $t' = 1$, and preferences admit a master list. As a consequence we obtain Corollary 16, showing that we cannot hope to solve MAX-UTILITY PARETO-OPTIMAL MATCHING WITH INSTABILITY COSTS in FPT time with parameter k. Thus, Algorithm 1 is roughly optimal.

Theorem 15. *Given a bipartite preference system $(G, \prec)$ with a stable matching M_s, finding a matching in $(G, \prec)$ with at most k blocking edges that is larger than M_s is* W[1]-*hard with parameter k, even if preferences in $(G, \prec)$ admit a master list on both sides and $|M_s| = |V(G)|/2 - 1$.*

Proof. Recall the reduction presented in the proof Theorem 11: given an instance $(G = (V, E), q)$ of MULTICOLORED CLIQUE with V partitioned into q sets $V_1, \ldots, V_q$, we constructed an equivalent instance $I = (G', \prec, \omega, c, t, k = q + \binom{q}{2})$ of MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS. We claim that this construction is correct here as well in the sense that $(G', \prec)$ admits a matching M of size $|M_s| + 1 = |V(G')|/2$ and with $\mathsf{bp}_{G'}(M) \leq k$ if and only if G has a clique $v_1, \ldots, v_q$ in G with $v_i \in V_i$ for each $i \in [q]$.

On the one hand, given such a clique in G, the corresponding matching M defined in the proof of Theorem 11 is clearly a matching that is larger than the unique stable matching M_s in $(G', \prec)$ and has exactly k blocking pairs.

On the other hand, let M be a matching in $(G', \prec)$ that is larger than M_s and has at most k blocking edges. Note that M is feasible in instance I of MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS, but we do not know whether it is popular in $(G', \prec)$ or not. Nevertheless, observe that in the proof of Theorem 11 we only use the popularity of M in order to show that certain vertices must be matched by M. This is, however, obvious now, since M is a complete matching. Therefore, we can apply the same arguments we used in the proof of Theorem 11, making use of the completeness of M instead of its popularity, to obtain a clique in G as required.

Thus, the reduction is indeed correct and proves the theorem. $\qquad\square$

		input settings			result	
problem	graph	preferences	costs	utilities	complexity	theorem
popular	general	general	$c \equiv 1, k = 1$	ω fixed, $t = 0$	NP-c	Cor. 2
popular	bipartite, $\Delta_G = 3$	SP, SC	$c \equiv 1, k = 1$	ω almost always 0, $t = 1$	NP-c	Thm. 6
popular	bipartite, $\Delta_G = 3$	SP, SC	$c \equiv 1, k = 1$	$\omega \equiv 1, t = |M_s| + 1$	NP-c	Thm. 7
popular	bipartite	1-ML	$c \geq 1$		$O(|E|^{k+1})$ alg.	Thm 10
popular	bipartite	2-ML	$c \equiv 1$	ω almost always 0, $t = 1$	W[1]-h wrt k	Thm. 11
popular	bipartite	2-ML	$c \equiv 1$	$\omega \equiv 1, t = |M_s| + 1$	W[1]-h wrt k	Thm. 11
popular	bipartite	2-ML	c binary, $k = 0$	ω almost always 0, $t = 1$	NP-c	Thm. 12
popular	bipartite	2-ML	c binary, $k = 0$	$\omega \equiv 1, t = |M_s| + 1$	NP-c	Thm. 12
popular w/ ties	bipartite, $\Delta_G = 3$	2-ML	$c \equiv 1, k = 1$	ω almost always 0, $t = 1$	NP-c	Thm. 13
Pareto-opt	bipartite	general	$c \geq 1$		$k! \cdot O(|E|^{k+2})$ alg.	Thm. 14
Pareto-opt	bipartite	2-ML	$c \equiv 1$	$\omega \equiv 1, t = |M_s| + 1$	W[1]-h wrt k	Cor. 16

Table 1: Summary of our results. We simply write "popular" ("Pareto-opt") to denote the problem MAX-UTILITY POPULAR (PARETO-OPTIMAL) MATCHING WITH INSTABILITY COSTS, and add "w/ ties" for the variant with ties. SP / SC / 1-ML / 2-ML mean preferences that are single-peaked /single-crossing / admit a master list on one side / on both sides, resp. We describe ω as "almost always 0" when ω is binary and takes value 1 only on a single edge. We let M_s denote a stable matching in the input.

Corollary 16. *The* MAX-UTILITY PARETO-OPTIMAL MATCHING WITH INSTABILITY COSTS *problem is* W[1]*-hard with budget k as parameter, if*

- *the input graph G is bipartite,*
- *preferences admit a master list on both sides,*
- *the cost function is $c \equiv 1$,*
- *the utility function is $\omega \equiv 1$, and*
- *the objective value is $t = |M_s| + 1 = \frac{|V(G)|}{2}$ for a stable matching M_s of G.*

Proof. The corollary is an immediate consequence of Theorem 15 and the following observation: G contains a matching of size t with at most k blocking edges if and only if G contains a Pareto-optimal matching of size t with at most k blocking edges. To see the non-trivial direction of this observation, consider any matching M of size t for which $|\mathrm{bp}(M)| \leq k$. By applying Pareto-improvements to M as long as possible, we obtain a Pareto-optimal matching M' where no vertex is worse off than in M. This implies that an edge that blocks M' also blocks M (note that no edge in $M \setminus M'$ blocks M'). It follows that $|\mathrm{bp}_G(M')| \leq k$, and it is also clear that $|M'| \geq |M| = t$, proving the observation. $\square$

6 Conclusion

We studied the MAX-UTILITY POPULAR MATCHING WITH INSTABILITY COSTS problem, and painted a detailed landscape of its computational complexity by showing how the utility and cost functions ω and c, the budget k, our objective value t, and various restrictions on the preferences affect the tractability of the problem. We also made a brief detour into a relaxation where the requirement of popularity is replaced by Pareto-optimality. Table 1 summarizes our results.

An interesting open question is whether the algorithm of Theorem 10 can be extended to preference domains that are, in some sense, close to admitting a master list. For example, can we efficiently solve instances where preferences of almost all vertices admit a master list?

References

[1] D. J. Abraham, P. Biró, and D. F. Manlove. "Almost stable" matchings in the roommates problem. In T. Erlebach and G. Persiano, editors, *WAOA '05: Proceedings of the 3rd Workshop on Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.

[2] H. Aziz, P. Biró, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with uncertain linear preferences. *Algorithmica*, 82(5):1410–1433, 2020.

[3] S. Bade. Matching with single-peaked preferences. *Journal of Economic Theory*, 180:81–99, 2019.

[4] M. Balinski and T. Sönmez. A tale of two mechanisms: Student placement. *Journal of Economic Theory*, 84(1):73–94, 1999.

[5] J. Bartholdi III and M. A. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.

[6] P. Biró, R. W. Irving, and D. F. Manlove. Popular matchings in the marriage and roommates problems. In *CIAC '10: Proceedings of the 7th International Conference on Algorithms and Complexity*, volume 6078 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2010.

[7] P. Biró, D. F. Manlove, and E. J. McDermid. "Almost stable" matchings in the roommates problem with bounded preference lists. *Theoretical Computer Science*, 432:10–20, 2012.

[8] P. Biró, D. F. Manlove, and S. Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411:1828–1841, 2010.

[9] S. Braun, N. Dwenger, and D. Kübler. Telling the truth may not pay off: an empirical study of centralized university admissions in Germany. *The B.E. Journal of Economic Analysis and Policy*, 10, 2010. Article 22.

[10] R. Bredereck, J. Chen, U. P. Finnendahl, and R. Niedermeier. Stable roommates with narcissistic, single-peaked, and single-crossing preferences. *Autonomous Agents and Multi-agent Systems*, 34(2):53–53, 2020.

[11] R. Bredereck, K. Heeger, D. Knop, and R. Niedermeier. Multidimensional stable roommates with master list. In *WINE '20: Proceedings of the 16th International Conference on Web and Internet Economics*, pages 59–73. Springer, 2020.

[12] J. Chen. Computational complexity of stable marriage and stable roommates and their variants. *arXiv preprint arXiv:1904.08196*, 2019.

[13] J. Chen, D. Hermelin, M. Sorge, and H. Yedidsion. How hard is it to satisfy (almost) all roommates? In *ICALP 2018: Proceedings of the 45th International Colloquium on Automata, Languages, and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[14] X. Chen, G. Ding, X. Hu, and W. Zang. The maximum-weight stable matching problem: duality and efficiency. *SIAM Journal on Discrete Mathematics*, 26(3):1346–1360, 2012.

[15] Y. Chen and T. Sönmez. Improving efficiency of on-campus housing: an experimental study. *American Economic Review*, 92:1669–1686, 2002.

[16] Á. Cseh, Y. Faenza, T. Kavitha, and V. Powers. Understanding popular matchings via stable matchings. *SIAM Journal on Discrete Mathematics*, 36(1):188–213, 2022.

[17] Á. Cseh, T. Friedrich, and J. Peters. Pareto optimal and popular house allocation with lower and upper quotas. In *AAMAS '22: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 300–308, 2022.

[18] Á. Cseh, R. W. Irving, and D. F. Manlove. The stable roommates problem with short lists. *Theory of Computing Systems*, 63(1):128–149, 2019.

[19] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, Cham, 2015.

[20] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer, New York, 1999.

[21] E. Elkind, P. Faliszewski, M. Lackner, and S. Obraztsova. The complexity of recognizing incomplete single-crossing preferences. In *AAAI 2015: Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 865–871. AAAI Press, 2015.

[22] D. Epple, R. Romano, and H. Sieg. Admission, tuition, and financial aid policies in the market for higher education. *Econometrica*, 74(4):885–928, 2006.

[23] Y. Faenza and T. Kavitha. Quasi-popular matchings, optimality, and extended formulations. *Mathematics of Operations Research*, 47(1):427–457, 2022.

[24] Y. Faenza, T. Kavitha, V. Powers, and X. Zhang. Popular matchings and limits to tractability. *SIAM Journal on Discrete Mathematics*, 36(1):188–213, 2022.

[25] T. Feder. A new fixed point approach for stable networks and stable marriages. *Journal of Computer and System Sciences*, 45:233–284, 1992.

[26] T. Feder. Network flow and 2-satisfiability. *Algorithmica*, 11:291–319, 1994.

[27] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410:53–61, 2009.

[28] Z. Fitzsimmons and M. Lackner. Incomplete preferences in single-peaked electorates. *Journal of Artificial Intelligence Research*, 67:797–833, 2020.

[29] P. Floréen, P. Kaski, V. Polishchuk, and J. Suomela. Almost stable matchings by truncating the Gale-Shapley algorithm. *Algorithmica*, 58(1):102–118, 2010.

[30] A.-T. Gai, D. Lebedev, F. Mathieu, F. De Montgolfier, J. Reynier, and L. Viennot. Acyclic preference systems in P2P networks. In A. Kermarrec, L. Bougé, and T. Priol, editors, *Euro-Par '07: Proceedings of the 13th International Euro-Par Conference (European Conference on Parallel and Distributed Computing)*, volume 4641 of *Lecture Notes in Computer Science*, pages 825–834. Springer, 2007.

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

[32] P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Science*, 20:166–173, 1975.

[33] M. R. Garey and D. S. Johnson. *Computers and Intractability*. Freeman, San Francisco, CA., 1979.

[34] S. Gupta, P. Jain, S. Roy, S. Saurabh, and M. Zehavi. On the (Parameterized) Complexity of Almost Stable Marriage. In *FSTTCS 2020: Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 182 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[35] S. Gupta, P. Misra, S. Saurabh, and M. Zehavi. Popular matching in roommates setting is NP-hard. *ACM Transactions on Computation Theory*, 13(2):1–20, 2021.

[36] D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[37] K. Hamada, K. Iwama, and S. Miyazaki. An improved approximation lower bound for finding almost stable maximum matchings. *Information Processing Letters*, 109:1036–1040, 2009.

[38] C.-C. Huang and T. Kavitha. Popular matchings in the stable marriage problem. *Information and Computation*, 222:180–194, 2013.

[39] C.-C. Huang and T. Kavitha. Popularity, mixed matchings, and self-duality. *Mathematics of Operations Research*, 46(2):405–427, 2021.

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

[41] R. W. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM*, 34:532–543, 1987.

[42] R. W. Irving, D. F. Manlove, and S. Scott. The stable marriage problem with master preference lists. *Discrete Applied Mathematics*, 156(15):2959–2977, 2008.

[43] N. Kamiyama. Many-to-many stable matchings with ties, master preference lists, and matroid constraints. In *AAMAS '19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 583–591, 2019.

[44] T. Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43:52–71, 2014.

[45] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, and U. Schmidt-Kraepelin. The popular assignment problem: when cardinality is more important than popularity. In *SODA '22: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 103–123. SIAM, 2022.

[46] T. Kavitha, M. Nasre, and P. Nimbhorkar. Popularity at minimum cost. *Journal of Combinatorial Optimization*, 27(3):574–596, 2014.

[47] S. Khuller, S. G. Mitchell, and V. V. Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127:255–267, 1994.

[48] A. Kwanashie. *Efficient algorithms for optimal matching problems under preferences*. PhD thesis, University of Glasgow, 2015.

[49] D. Lebedev, F. Mathieu, L. Viennot, A.-T. Gai, J. Reynier, and F. De Montgolfier. On using matching theory to understand P2P network design. In *INOC '07: Proceedings of the 3rd International Network Optimization Conference*, 2007.

[50] D. F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

[51] K. Meeks and B. Rastegari. Solving hard stable matching problems involving groups of similar agents. *Theoretical Computer Science*, 844:171–194, 2020.

[52] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

[53] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *STACS 2010: Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science*, pages 17–32, 2010.

[54] J. B. Orlin. Max flows in $O(nm)$ time, or better. In *STOC '13: Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, page 765–774, 2013.

[55] N. Perach, J. Polak, and U. G. Rothblum. A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the Technion. *International Journal of Game Theory*, 36:519–535, 2008.

[56] A. E. Roth and M. A. O. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, 1990.

[57] U. G. Rothblum. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54:57–67, 1992.

[58] A. Saporiti. Strategic voting on single-crossing domains. Technical report, Economics, The University of Manchester, 2006.

[59] C.-P. Teo and J. Sethuraman. LP based approach to optimal stable matchings. In M. E. Saks, editor, *SODA '97: Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*, pages 710–719. ACM-SIAM, 1997.

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