Maximum Matchings and Short Voting Paths

Telikepalli Kavitha
Tata Institute of Fundamental Research, Mumbai, India

Abstract

Our input is a marriage instance $G = (A \cup B, E)$, i.e., it is a bipartite graph where every vertex has strict preferences over its neighbors. The preferences that a vertex has on its neighbors extend naturally to preferences over matchings. A maximum matching M that does not lose an election against any maximum matching (where vertices cast votes) is a *popular maximum-matching.* These matchings are useful in practice – they always exist and can be efficiently computed [Kavitha, SICOMP 2014]. Suppose preferences change; then the problem is to update the current matching M_0 via a short *voting path* to a popular maximum-matching, where a length-ℓ voting path from M_0 to M_ℓ is a sequence of maximum matchings $\langle M_0, M_1, \ldots, M_\ell \rangle$ such that each matching is more popular than its predecessor. There are maximum matchings from which there is no voting path (of any length) to a popular maximum-matching [Bhattacharya et al., ICALP 2015].

We show a polynomial-time algorithm to decide if there exists a *short* voting path, i.e. one of length ≤ 2, from a given maximum matching to a popular maximum-matching and find one, if so. Voting paths motivate natural relaxations of popularity: *pseudo-popular* maximum-matchings and *mostly-popular* maximum-matchings; these yield more egalitarian or optimal solutions than popular maximum-matchings. We show polynomial-time algorithms to compute such optimal solutions that go beyond popularity. In particular, we give a combinatorial characterization of pseudo-popular maximum-matchings in terms of forced vertices and forbidden edges. We also show a polynomial-time algorithm to compute at most $|E| = m$ popular maximum-matchings such that any maximum matching that loses to some popular maximum-matching loses to at least one of these m matchings.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Bipartite graphs, Fractional matchings, Polytopes, LP duality

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.56

Funding Supported by the Department of Atomic Energy, Government of India, under project no. RTI4014.

Acknowledgements Thanks to the reviewers of this paper for their helpful comments and suggestions that improved the presentation of the paper.

1 Introduction

Our input is a bipartite graph $G = (A \cup B, E)$ on n vertices and m edges where every vertex has a strict ranking of its neighbors. Such a graph is called a *marriage* instance. The problem is to find a "best" matching in G as per a suitable notion of optimality. The well-studied notion of optimality in this model is stability. A matching M is stable if no edge blocks it where an edge (a, b) blocks M if a and b prefer each other to their respective assignments in M. Stable matchings always exist in a marriage instance and the Gale-Shapley algorithm [12] finds one in linear time.

The Gale-Shapley algorithm and its variants are used in many real-world applications, e.g., its many-to-one variant [12] is used to match students to schools/colleges [1, 4] and medical residents to hospitals [8, 24]. Stability is a very strong notion – all stable matchings have the same size [13], which could be only half the size of a maximum matching. There are applications, e.g., assigning teaching assistants to courses or students to projects, where we seek larger matchings; this has motivated relaxations of stability. The notion of *popularity* is a well-studied relaxation of stability that captures collective welfare.

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 56; pp. 56:1–56:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Popularity. Any pair of matchings, say M and N, can be compared by holding an election between them where every vertex casts a vote for the matching in $\{M, N\}$ that it prefers[1] or it abstains from voting if it has the same assignment in M and N. Let $\phi(M, N)$ (resp., $\phi(N, M)$) be the number of votes for M (resp., N) in the M-vs-N election. We say N is more popular than M or "N *defeats* M" if $\phi(N, M) > \phi(M, N)$. This is denoted by $N \succ M$.

▶ **Definition 1.** *A matching M is popular if there is no matching more popular than M, i.e., $\phi(M, N) \geq \phi(N, M)$ for all matchings N in G.*

Thus no matching defeats a popular matching. Hence popular matchings are *weak Condorcet winners* [10, 23] in elections where matchings are candidates and vertices are voters. Gärdenfors [14] introduced the notion of popularity in 1975 where he observed that every stable matching is popular. Stable matchings are min-size popular matchings [16]. Max-size popular matchings have size at least 2/3·(size of a maximum matching) and there are instances where this bound is tight [17]. So there need not be any *maximum* matching that is popular. Hence in applications – such as assigning teaching assistants to courses – where we cannot compromise on the size of the matching, i.e., when it is only maximum matchings that are feasible, a useful class of matchings is the set of popular maximum-matchings.

▶ **Definition 2.** *A maximum matching M in $G = (A \cup B, E)$ is a popular maximum-matching if $\phi(M, N) \geq \phi(N, M)$ for all maximum matchings N.*

A popular maximum-matching need not be popular among all matchings, but it is popular within the set of all maximum matchings. Thus popular maximum-matchings prioritize cardinality over popularity. So it is only maximum matchings that are candidates in this election and popular maximum-matchings are weak Condorcet winners here. It was shown in [17] that popular maximum-matchings always exist in any marriage instance. Furthermore, popular maximum-matchings have been used in real-world applications, e.g., from July 2022 onward, the Computer Science and Engineering Department at IIT Madras has been using a popular maximum-matching to assign teaching assistants to courses as mentioned in [21, Section 1].

Our problem. Given a marriage instance G, we can find a popular maximum-matching M in polynomial time [17] (see Appendix B). Consider a dynamic matching market where vertices are allowed to change preferences. Then M may stop being popular among maximum-matchings at some point in time – the goal is to update M to a popular maximum-matching. One way to update M to a popular maximum-matching would be to ask for a popular maximum-matching P that minimizes the symmetric difference $M \oplus P$. Such a matching P can be computed in polynomial time via the characterization of popular maximum-matchings in G as stable matchings in an auxiliary multigraph G^* [18] – so we have to find a stable matching in G^* that shares the maximum possible number of edges with M; such a matching can be found in polynomial time by optimizing over the stable matching polytope of G^* [25].

The model of popularity mandates that we can switch from one maximum matching M to another maximum matching P only if there is consensus among the vertices to agree to the switch, i.e., only if P is more popular than M. But the popular maximum-matching P, which was computed in the paragraph above, need not be more popular than M. This motivates the notion of "reachability" defined below.

[1] A vertex prefers the matching where it gets a better partner and being unmatched is its worst choice.

▶ **Definition 3** ([2]). *A maximum matching M_ℓ is* reachable *from maximum matching M_0 if there is a sequence of maximum matchings $\langle M_0, M_1, \ldots, M_\ell \rangle$ such that $M_\ell \succ M_{\ell-1} \cdots \succ M_1 \succ M_0$. Such a sequence is called a length-ℓ* voting path *from M_0 to M_ℓ.*

Given a marriage instance G and a maximum matching M, is there always a popular maximum-matching reachable from M? The answer is "no" as shown by the following instance from [5] where $A = \{a_1, a_2, a_3\}, B = \{b_1, b_2, b_3\}$, and the underlying graph is the complete bipartite graph. Vertex preferences are as follows.

$$a_1\colon\ b_2 \succ b_1 \succ b_3 \qquad b_1\colon a_3 \succ a_1 \succ a_2$$
$$a_2\colon\ b_1 \succ b_3 \succ b_2 \qquad b_2\colon a_2 \succ a_3 \succ a_1$$
$$a_3\colon\ b_3 \succ b_2 \succ b_1 \qquad b_3\colon a_1 \succ a_2 \succ a_3$$

Agent a_1 regards b_2 as its top choice, b_1 as its second choice, b_3 as its third choice and other preference lists are analogous. Let $M = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}, M' = \{(a_1, b_3), (a_2, b_1), (a_3, b_2)\}$, and $M'' = \{(a_1, b_2), (a_2, b_3), (a_3, b_1)\}$. We have $M \succ M' \succ M'' \succ M$. The remaining three perfect matchings here are the stable matchings $S = \{(a_1, b_2), (a_2, b_1), (a_3, b_3)\}$, $S' = \{(a_1, b_1), (a_2, b_3), (a_3, b_2)\}$, and $S'' = \{(a_1, b_3), (a_2, b_2), (a_3, b_1)\}$.

Observe that none of S, S', S'' is more popular than any of M, M', M''. Thus no popular maximum-matching is reachable from any of M, M', M''. This gives rise to the following algorithmic question: given a marriage instance G and a maximum matching M, is there a polynomial-time algorithm to decide if a popular maximum-matching is reachable from M? Unfortunately, we do not know the answer to this question.

Previous work. The problem of constructing voting paths to popular matchings was originally posed in [2] in the setting of *1-sided* preferences where vertices in A are agents with preferences (weak rankings) while vertices in B are houses and have no preferences. Popular matchings need not always exist in this setting – it was shown in [3] that whenever the instance admits a popular matching, there is a voting path of length at most 2 from any matching to a popular matching; moreover, a shortest such path can be computed efficiently.

The above result in [3] motivates the following simpler question in our setting:

- Given a maximum matching M in a marriage instance G, is there a polynomial-time algorithm to decide if a popular maximum-matching is reachable from M by a voting path of length at most 2?

Indeed, long voting paths are difficult to implement in real-world applications because it would involve transiting through several intermediate matchings. So it makes sense to ask if there is a *short* voting path, i.e., one of length ≤ 2 (thus at most one intermediate matching), from a given matching to a popular maximum-matching. We show the following result.

▶ **Theorem 4.** *Let M be a maximum matching in a marriage instance $G = (A \cup B, E)$. There is a polynomial-time algorithm to decide if a popular maximum-matching is reachable from M by a voting path of length at most 2 and if so, to find a shortest such path.*

When all matchings (not just maximum matchings) are feasible in a marriage instance, it is NP-complete to decide if there exists a popular matching that defeats a given matching M [19]. In other words, it is NP-hard to decide if there exists a length-1 voting path from a given matching to a popular matching. However when it is only *maximum* matchings that are feasible, the length-1 and length ≤ 2 voting path problems become tractable (by Theorem 4).

Beyond popularity. Voting paths are relevant in the *static* setting as well (i.e., when preferences remain fixed) because they motivate the following natural relaxation of popularity.

▶ **Definition 5.** *A maximum matching M is a* pseudo-popular *maximum-matching in a marriage instance $G = (A \cup B, E)$ if M is undefeated by all popular maximum-matchings.*

A pseudo-popular maximum-matching M has *no* length-1 voting path to a popular maximum-matching, i.e., no popular maximum-matching defeats M. Thus M may be unpopular, but there is no single step switch from M to a popular maximum-matching. So in models where we are allowed only single step switches, i.e., only voting paths of length 1, it is not easy to replace M as it has no good replacement. Thus M is "more or less popular".

Min-cost solutions. Relaxing popularity to pseudo-popularity expands our pool of admissible solutions. In applications such as assigning TAs to courses, *egalitarian* matchings are most desirable; these are min-cost matchings where the cost of any edge (a, b) is the sum of ranks of a and b for each other. There is a polynomial-time algorithm to find a min-cost popular maximum-matching in a marriage instance [18], thus we can find an egalitarian popular maximum-matching in G in polynomial time.

By expanding our pool of admissible matchings to pseudo-popular maximum-matchings, we can get more egalitarian matchings as shown by the following example. Let $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $B = \{b_1, b_2, b_3, b_4\}$. Vertex preferences are as follows.

$a_1\colon\ b_1 \succ b_2$ $\qquad$ $b_1 : a_1 \succ a_2$

$a_2\colon\ b_1 \succ b_3 \succ b_2$ $\qquad$ $b_2 : a_1 \succ a_3 \succ a_2 \succ a_5$

$a_3\colon\ b_3 \succ b_4 \succ b_2$ $\qquad$ $b_3 : a_3 \succ a_4 \succ a_2$

$a_4\colon\ b_3 \succ b_4$ $\qquad$ $b_4 : a_3 \succ a_4$

$a_5\colon\ b_2$

This instance has two popular maximum-matchings: $S = \{(a_i, b_i) : 1 \le i \le 4\}$ and $P = \{(a_1, b_1), (a_2, b_2), (a_3, b_4), (a_4, b_3)\}$. Both S and P have egalitarian cost 14. Consider $M = \{(a_1, b_2), (a_2, b_1), (a_3, b_3), (a_4, b_4)\}$: its egalitarian cost is 12.

- Observe that neither S nor P defeats M. But $N_0 = \{(a_1, b_1), (a_5, b_2), (a_3, b_3), (a_4, b_4)\}$ and $N_1 = \{(a_1, b_1), (a_5, b_2), (a_3, b_4), (a_4, b_3)\}$ defeat M.

Since neither N_0 nor N_1 is a popular maximum-matching, M is a pseudo-popular maximum-matching. The following result follows from the proof of Theorem 4.

▶ **Corollary 6.** *A min-cost pseudo-popular maximum-matching in a marriage instance $G = (A \cup B, E)$ with* $\mathsf{cost} : E \to \mathbb{R}$ *can be computed in polynomial time.*

Once a min-cost pseudo-popular maximum-matching M is announced for some application, queries may be raised that some other maximum matching N is more popular than M. A maximum matching that defeats N can be returned as an answer, thereby showing that N is also not a popular maximum-matching. So relaxing popularity to pseudo-popularity entails such a query answering algorithm to "defend" M. In order to avoid running a query answering algorithm, let us consider the following stronger relaxation of popularity.

▶ **Definition 7.** *A maximum matching M is a* mostly-popular *maximum-matching if M is undefeated by all pseudo-popular maximum-matchings, i.e., $\phi(M, N) \ge \phi(N, M)$ for all pseudo-popular maximum-matchings N.*

In our example where both the popular maximum-matchings S and P have egalitarian cost 14, observe that $S \succ N_0$ and $P \succ N_1$. Thus neither N_0 nor N_1 is a pseudo-popular maximum-matching. So M (with egalitarian cost 12) is a *mostly-popular* maximum-matching.

Observe that M, M', M'' in the example from [5] described earlier are pseudo-popular maximum-matchings, but they are *not* mostly-popular since $M \succ M' \succ M'' \succ M$. A quick overview of the three notions of optimality considered here (see Figure 1).

- **Popular maximum-matchings**: These are maximum matchings undefeated by all maximum matchings.
- **Pseudo-popular maximum-matchings**: These are maximum matchings with *no* length 1 voting path to any popular maximum-matching. Thus these are maximum matchings undefeated by all popular maximum-matchings.
- **Mostly-popular maximum-matchings**: These are maximum matchings undefeated by all pseudo-popular maximum-matchings.

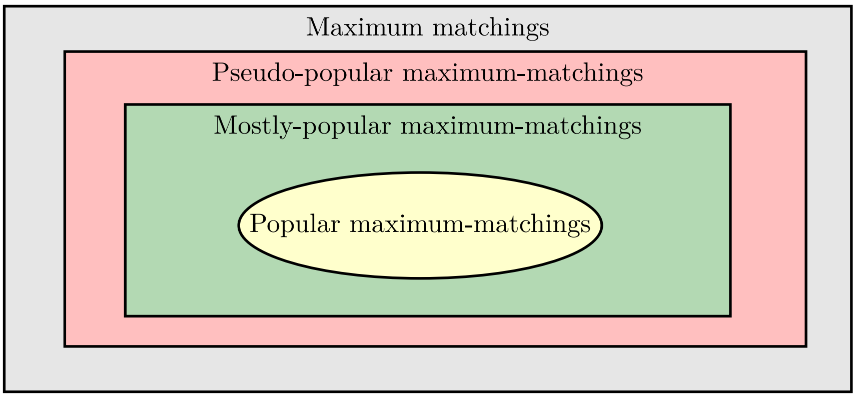

Figure 1 Mostly-popular maximum-matchings are popular within the set of pseudo-popular maximum-matchings. So no matching in the pink set defeats a mostly-popular maximum-matching.

Consider the following instance on $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, b_2, b_3\}$. Vertex preferences are as follows.

$$\begin{array}{llll} a_1: & b_1 \succ b_2 \succ b_3 & \qquad b_1: & a_1 \succ a_2 \succ a_3 \\ a_2: & b_1 \succ b_2 & \qquad b_2: & a_1 \succ a_2 \succ a_4 \\ a_3: & b_1 \succ b_3 & \qquad b_3: & a_1 \succ a_3 \\ a_4: & b_2 & & \end{array}$$

Let $M_0 = \{(a_1, b_3), (a_2, b_2), (a_3, b_1)\}, M_1 = \{(a_1, b_2), (a_2, b_1), (a_3, b_3)\}, M_2 = \{(a_1, b_1), (a_4, b_2), (a_3, b_3)\}$, and $M_3 = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$. Observe that $M_3 \succ M_2 \succ M_1 \succ M_0$.

M_3 is the only popular maximum-matching here, so it belongs to the innermost set in Figure 1. Since $M_3 \succ M_2$, the matching M_2 is not pseudo-popular. So M_2 belongs to the outermost set (and no inner set) in Figure 1. Note that M_2 is the only matching that defeats M_1, so M_1 is mostly-popular and belongs to the green set. The matching M_0 is undefeated by M_3, but $M_1 \succ M_0$, hence M_0 belongs to the pink set, but it is not in the green set. Observe that M_0 has a length-3 voting path (and no shorter voting path) to M_3.

▶ **Theorem 8.** *A min-cost mostly-popular maximum-matching M in G can be computed in polynomial time.*

We prove Theorem 8 on our way to proving Theorem 4. As before, once M is announced for some application, queries may be raised that some maximum matching N is more popular than M. By definition, a popular maximum-matching P defeats N. Rather than compute P in response to query N, our algorithm can declare upfront at most $|E| = m$ popular maximum-matchings $P_1, \ldots, P_m$ such that any *non-pseudo-popular* maximum-matching – one defeated by a popular maximum-matching – is defeated by one of $P_1, \ldots, P_m$.

▶ **Theorem 9.** *A set $\mathcal{P}$ of at most m popular maximum-matchings can be computed in polynomial time such that any maximum matching that defeats M (see Theorem 8) is defeated by a matching in $\mathcal{P}$.*

There may be exponentially many popular maximum-matchings in G [26]. So it is pleasing to see that we can efficiently compute a set $\mathcal{P}$ of at most m popular maximum-matchings such that any maximum matching that is defeated by some popular maximum-matching is defeated by one in $\mathcal{P}$. Thus the few matchings in $\mathcal{P}$ defeat all non-pseudo-popular maximum-matchings. Such a set $\mathcal{P}$ can be called a *winning set* of popular maximum-matchings.

Background and Related Results. Algorithmic questions in the domain of popular matchings have been well-studied during the last 15-20 years; see [9] for a survey. Popular matching algorithms were first studied in the setting of 1-sided preferences [2]. Voting paths were defined in [2] where the problem of efficiently finding a shortest voting path from a given matching to a popular matching was posed. This question was answered in [3].

Popular matchings always exist in the setting of 2-sided strict preferences, i.e., when the input is a marriage instance. There are polynomial-time algorithms to check if a given matching is popular or not [6, 16]. In contrast to the 1-sided setting, explicit examples were given in [5] to show that there are marriage instances and matchings from which there is no voting path to a popular matching. Since the underlying graph in this example is a complete bipartite graph, it follows that there are maximum matchings from which there is no voting path to any popular maximum-matching.

In applications where it is only maximum matchings that are feasible, a natural relaxation of stability is a maximum matching with the least number of blocking edges. Finding such a matching is NP-hard; moreover, it is NP-hard to approximate within $n^{1-\varepsilon}$, for any $\varepsilon > 0$ [7]. In contrast to this, the notion of popularity nicely extends to the set of maximum matchings – popular maximum-matchings always exist in a marriage instance and there is an efficient algorithm to find one [17] (we recap this in Appendix B). Popular maximum-matchings have also been studied in the 1-sided setting. Such a matching need not exist and a polynomial-time algorithm was given in [20] to decide if the given instance admits one.

There is a polynomial-time algorithm to find a max-weight/min-cost popular maximum-matching in a marriage instance [18]. But computing a max-weight/min-cost popular matching in a marriage instance is NP-hard [11]. The hard instance from [11] was used in [19] to show it is NP-complete to decide if a given matching M is defeated by any popular matching, i.e., it is coNP-complete to decide if M is undefeated by all popular matchings.

Our Techniques. Given a maximum matching M, it is easy to use the max-weight popular maximum-matching algorithm from [18] to solve the problem of deciding if there exists a popular maximum-matching that defeats M: either every popular maximum-matching defeats M (see Lemma 10) or there is an edge weight function wt_M such that $\mathsf{wt}_M(P) = \phi(P, M) - \phi(M, P)$ for any popular-maximum matching P (see Section 3). In the latter case, computing a max-weight popular maximum-matching w.r.t. the edge weight function wt_M allows us to determine if there exists a popular maximum-matching $P \succ M$.

On the other hand, deciding if there exists a maximum matching N, along with a popular maximum-matching P, such that $P \succ N \succ M$ looks tricky. In the 1-sided setting, there is a combinatorial characterization of popular matchings [2] and this structure of popular matchings plays a crucial role in the voting paths result from [3]. However there is no combinatorial characterization of popular maximum-matchings in the 2-sided setting. We will show a combinatorial characterization of *pseudo-popular* maximum-matchings.

It is easy to show a rural hospitals theorem for pseudo-popular maximum-matchings, i.e., every pseudo-popular maximum-matching in G has to match the same subset $S \subseteq A \cup B$. A simple argument shows that the pseudo-popular maximum-matching polytope is a face of the S-perfect matching polytope of G. Thus pseudo-popular maximum-matchings can be characterized as S-perfect matchings in a subgraph of G. This subgraph can be determined in polynomial time by linear programming. Section 2 has these details. Our combinatorial characterization of pseudo-popular maximum-matchings plays a crucial role in proving Theorem 4 on length-2 voting paths (see Section 3).

We will use LP duality to construct at most m popular maximum-matchings that together defeat every *non-pseudo-popular* maximum-matching. There is a forbidden set of edges for pseudo-popular maximum-matchings. We will find a subset K of forbidden edges such that (i) every maximum matching that is *not* pseudo-popular contains an edge in K, and (ii) for each $e \in K$, there is a popular maximum-matching P_e that defeats all maximum matchings with edge e. This leads to Theorem 9 whose proof is given in Section 4.

Appendix A contains a proof missing from the main part of the paper. For the sake of completeness, we describe the popular maximum-matching algorithm [17] in Appendix B.

2 Pseudo-Popular Maximum-Matchings

Our input is a marriage instance $G = (A \cup B, E)$ on n vertices and m edges. Lemma 10 below implies that all popular maximum-matchings match the same set of vertices. Moreover, the same conclusion also holds for pseudo-popular maximum-matchings. Let P be any popular maximum-matching in G, and let $S \subseteq A \cup B$ denote the set of vertices matched in P.

▶ **Lemma 10.** *If N is a maximum matching that leaves one or more vertices of S unmatched, then P defeats N.*

Proof. Let N be any maximum matching in G. Since both P and N are maximum matchings, the symmetric difference $P \oplus N$ consists of alternating cycles and even length alternating paths. We have:

$$\phi(P, N) - \phi(N, P) \ = \sum_{\rho \in P \oplus N} \phi(P, P \oplus \rho) - \phi(P \oplus \rho, P),$$

where the sum is over all alternating paths/cycles ρ in $P \oplus N$. Because P is a popular maximum-matching, $\phi(P, P \oplus \rho) \geq \phi(P \oplus \rho, P)$ for all alternating paths/cycles $\rho \in P \oplus N$.

If N leaves one or more vertices of S unmatched, then there is at least one alternating path (call it q) in $P \oplus N$. Consider the election between P and $P \oplus q$. All vertices, other than those on q, are indifferent between P and $P \oplus q$. Because preferences are strict, no vertex on q is indifferent between P and $P \oplus q$. Moreover, the number of vertices on q is odd because q has even length. So $\phi(P, P \oplus q) \neq \phi(P \oplus q, P)$. Hence it has to be the case that $\phi(P, P \oplus q) > \phi(P \oplus q, P)$. Thus $\phi(P, N) > \phi(N, P)$, in other words, P defeats N. ◀

For any vertex u and neighbors v, v' of u, let $\mathsf{vote}_u(v, v')$ be u's vote for v versus v'. So:

$$\mathsf{vote}_u(v,v') = \begin{cases} 1 & \text{if } u \text{ prefers } v \text{ to } v'; \\ -1 & \text{if } u \text{ prefers } v' \text{ to } v; \\ 0 & \text{otherwise (so } v = v'). \end{cases}$$

For any two matchings M and N, let $\mathsf{vote}_u(M,N) = \mathsf{vote}_u(M(u), N(u))$, where $M(u)$ denotes u's partner when u is matched in M and $M(u) = \mathsf{null}$ if u is unmatched in M. Let $\Delta(M,N) = \sum_{u \in A \cup B} \mathsf{vote}_u(M,N)$. Equivalently, $\Delta(M,N) = \phi(M,N) - \phi(N,M)$.

Fractional maximum matchings. A fractional maximum matching $\vec{x}$ of G is a point in the maximum matching polytope of G. The function $\Delta(M,N)$ extends to $\Delta(M,\vec{x})$ as follows [22]. Since $\vec{x}$ belongs to the maximum matching polytope, $\vec{x}$ is a convex combination of certain maximum matchings $N_1, \ldots, N_t$, i.e., $\vec{x} = \sum_{i=1}^{t} p_i N_i$ where each $p_i \geq 0$ and $\sum_i p_i = 1$. Let $\Delta(M,\vec{x}) = \sum_{i=1}^{t} p_i \cdot \Delta(M, N_i)$.

Recall the set $S \subseteq A \cup B$ of vertices matched in every pseudo-popular maximum-matching (see Lemma 10). Let G_S be the subgraph of G induced on the vertex set S. So $G_S = (S, E_S)$ where $E_S = E \cap (S \times S)$. Claim 11 follows from [22] and its proof is given in Appendix A.

▷ Claim 11. Let M be a perfect matching and $\vec{x}$ be a perfect fractional matching in G_S. Then $\Delta(M,\vec{x}) = \mathsf{wt}_x(M) = \sum_{e \in M} \mathsf{wt}_x(e)$ for the following weight function wt_x on E_S:

$$\mathsf{wt}_x(a,b) \;=\; \left(\sum_{b' \prec_a b} x_{(a,b')} - \sum_{b' \succ_a b} x_{(a,b')} \right) + \left(\sum_{a' \prec_b a} x_{(a',b)} - \sum_{a' \succ_b a} x_{(a',b)} \right) \quad \forall\, (a,b) \in E_S,$$

where for any vertex u and its neighbor v, the set $\{v' : v' \prec_u v\}$ is the set of neighbors worse than v in u's preference order and the set $\{v' : v' \succ_u v\}$ is the set of neighbors better than v.

2.1 The Pseudo-Popular Maximum-Matching Polytope

Let $\mathcal{Q}_G$ be the pseudo-popular maximum-matching polytope of G, i.e., $\mathcal{Q}_G$ is the convex hull of the 0-1 edge incidence vectors of pseudo-popular maximum-matchings in G. Let $\vec{x} \in \mathcal{Q}_G$ and let P be any popular maximum-matching in G. Then $\Delta(P,\vec{x})$ is a convex combination of $\Delta(P, N_1), \ldots, \Delta(P, N_t)$ for some pseudo-popular maximum-matchings $N_1, \ldots, N_t$.

Since $\Delta(P, N_i) = 0$ for all $i = 1, \ldots, t$ and each N_i matches all vertices of S (by Lemma 10), we have the following two properties:

1. The fractional matching $\vec{x} \in \mathcal{Q}_G$ fully matches all vertices in S.
2. $\Delta(P,\vec{x}) = 0$ for every popular maximum-matching P in G.

Let $\mathcal{S}_G$ be the convex hull of the 0-1 edge incidence vectors of all S-*perfect* matchings in G, i.e., those that match all vertices of S. Since $|S|/2$ is the cardinality of a maximum matching in G, all vertices outside S have to be left unmatched in every S-perfect matching in G. So $\mathcal{S}_G$ is described by the following constraints:

$$\sum_{e \in \delta(u)} x_e \;=\; 1 \;\; \forall u \in S \qquad \text{and} \qquad x_e \;\geq\; 0 \;\; \forall e \in E, \tag{1}$$

where $\delta(u)$ is the set of edges incident to vertex u in G. For any popular maximum-matching P and any S-perfect matching N (so N is a maximum matching), we have $\Delta(P,N) \geq 0$.

Hence $\Delta(P,\vec{x}) \geq 0$ is a valid inequality for $\mathcal{S}_G$. Recall that the constraint $\Delta(P,\vec{x}) \geq 0$ is linear in the variables x_e (by Claim 11). By properties 1-2 given above, it follows that the pseudo-popular maximum-matching polytope $\mathcal{Q}_G$ can be expressed as follows:

$$\mathcal{Q}_G = \{\vec{x} \in \mathcal{S}_G : \Delta(P,\vec{x}) = 0 \text{ for all popular maximum-matchings } P\}.$$

Thus $\mathcal{Q}_G$ is the set of points $\vec{x} \in \mathcal{S}_G$ that satisfy all the valid inequalities $\Delta(P,\vec{x}) \geq 0$ tightly. The following observation follows from the definition of a face [15, Definition 3.11]. This is similar to [19, Observation 4.1].

▶ **Observation 12.** *The polytope $\mathcal{Q}_G$ is a face of the polytope $\mathcal{S}_G$.*

2.2 Forbidden Edges

The polytope $\mathcal{Q}_G$ is a non-empty face of $\mathcal{S}_G$ because popular maximum-matchings always exist in G. Any non-empty face of a polytope can be described by making some of the constraints of the polytope tight (see [15, Theorem 3.5]). So $\mathcal{Q}_G$ can be described by making some of the constraints in (1) tight. Thus $\mathcal{Q}_G$ is described by (1) and (2) for some $F \subset E_S$ where $E_S = E \cap (S \times S)$.

$$x_e \;=\; 0 \;\; \forall e \in F. \tag{2}$$

What is unknown in constraints (1)-(2) is the set $F \subset E_S$ of forbidden edges for pseudo-popular maximum-matchings in G. Observe that we have $x_e = 0$ for all $e \in E \setminus E_S$ because all vertices outside S are left unmatched in any S-perfect matching. For any edge $f \in E_S$, we can decide if $f \in F$ by solving linear program (LP1) given below, where $\delta_S(u) = \delta(u) \cap E_S$.

$$\text{maximize } x_f \tag{LP1}$$

$$\text{subject to}$$

$$\sum_{e \in \delta_S(u)} x_e \;=\; 1 \;\; \forall\, u \in S, \qquad x_e \;\geq\; 0 \;\; \forall\, e \in E_S, \quad \text{and} \tag{3}$$

$$\Delta(M,\vec{x}) \;=\; 0 \;\; \forall \text{ popular maximum-matchings } M \text{ in } G. \tag{4}$$

The variables are x_e for $e \in E_S$. The feasible region of (LP1) is the pseudo-popular maximum-matching polytope $\mathcal{Q}_G$. There are exponentially many constraints in (4).

▶ **Lemma 13.** *Linear program* (LP1) *admits a polynomial-time separation oracle.*

Proof. For a given perfect fractional matching $\vec{x}$ in G_S, consider the problem of finding $\max_M \Delta(M,\vec{x}) = \max_M \mathsf{wt}_x(M)$ where the max is over all popular maximum-matchings M in G. To solve this, we need to find a max-weight popular maximum-matching in G with the function wt_x extended to all edges in E. So we set $\mathsf{wt}_x(e) = 0$ for all $e \in E \setminus E_S$ since no edge outside E_S belongs to any popular maximum-matching (by Lemma 10). For any edge $(a,b) \in E_S$, we set: (see Claim 11)

$$\mathsf{wt}_x(a,b) \;=\; \left(\sum_{b' \prec_a b} x_{(a,b')} - \sum_{b' \succ_a b} x_{(a,b')}\right) + \left(\sum_{a' \prec_b a} x_{(a',b)} - \sum_{a' \succ_b a} x_{(a',b)}\right).$$

There is a polynomial-time algorithm to compute a max-weight popular maximum-matching in a marriage instance [18], so we have a desired matching P in polynomial time.

- If $\mathsf{wt}_x(P) > 0$ then we have a violating constraint $\Delta(P, \vec{x}) > 0$ in (4) for $\vec{x}$.
- Else $\mathsf{wt}_x(M) = \Delta(M, \vec{x}) = 0$ for all popular maximum-matchings M in G, so $\vec{x}$ satisfies all constraints in (4).

Hence (LP1) admits a polynomial-time separation oracle. ◀

Summarizing, given an edge $f \in E_S$, we can determine if $f \in F$ or not by solving (LP1). Since the feasible region of this linear program is the pseudo-popular maximum-matching polytope $\mathcal{Q}_G$, the optimal value of (LP1) is either 0 or 1. The optimal value is 1 if and only if there is a pseudo-popular maximum-matching with the edge f, which means $f \notin F$.

By solving (LP1) for every edge in E_S, we can determine the set F of forbidden edges for pseudo-popular maximum-matchings in polynomial time. Thus we have the following lemma.

▶ **Lemma 14.** *There is a polynomial-time algorithm to compute $F \subset E_S$ such that the set of perfect matchings in the subgraph $G'_S = (S, E_S \setminus F)$ is the set of pseudo-popular maximum-matchings in the marriage instance $G = (A \cup B, E)$.*

Lemma 14 reduces the problem of computing a min-cost pseudo-popular maximum-matching in G to computing a min-cost perfect matching in the graph $G'_S = (S, E_S \setminus F)$. The min-cost perfect matching problem in bipartite graphs is a classical polynomial-time solvable problem [15], thus a min-cost pseudo-popular maximum-matching in G can be computed in polynomial time. Hence we can conclude Corollary 6 stated in Section 1.

Since a matching is a pseudo-popular maximum-matching if and only if it is a perfect matching in $G'_S = (S, E_S \setminus F)$ (by Lemma 14), M is a *mostly-popular* maximum-matching in G (see Definition 7) if and only if M is a popular perfect-matching in G'_S; in other words, M is a perfect matching in G'_S that is popular within the set of all perfect matchings in G'_S. Thus the min-cost popular maximum-matching algorithm [18] in G'_S finds a min-cost mostly-popular maximum-matching in G in polynomial time. Hence we can conclude Theorem 8.

3 Short Voting Paths

Let M be a maximum matching in G. There is a length-1 voting path $\langle M, P\rangle$ where P is a popular maximum-matching if and only if either (1) M is not S-perfect or (2) M is S-perfect and $M \cap F \neq \emptyset$ (by Lemma 14). In case (1), for any popular maximum-matching P, we have $P \succ M$ (by Lemma 10).

In case (2), we need to compute $\max_P \Delta(P, M)$ where max is over all popular maximum-matchings P in G. Consider the function $\mathsf{wt}_M : E \to \{0, \pm 2\}$ defined as follows. Let $\mathsf{wt}_M(e) = 0$ if $e \in E \setminus E_S$ and for any edge $(a, b) \in E_S$:

$$\mathsf{wt}_M(a,b) = \begin{cases} 2 & \text{if } (a,b) \text{ is a blocking edge to } M; \\ -2 & \text{if both } a \text{ and } b \text{ prefer their respective partners in } M \text{ to each other;} \\ 0 & \text{otherwise.} \end{cases}$$

The function wt_M is the same as wt_x (from Claim 11/proof of Lemma 13) where $\vec{x}$ is the edge incidence vector of M. So for any S-perfect matching P, we have $\mathsf{wt}_M(P) = \Delta(P, M)$. Since $M \cap F \neq \emptyset$, we know there is a popular maximum-matching P that defeats M, i.e., $\Delta(P, M) > 0$. So the max-weight popular maximum-matching algorithm in G with the edge weight function wt_M finds a desired matching P. Hence the length-1 voting path problem to a popular maximum-matching can be solved in polynomial time.

Length-2 voting path problem. Here we are given a pseudo-popular maximum-matching M and the problem is to decide if there exists a "non-pseudo-popular" maximum-matching N, i.e., N has to be defeated by some popular maximum-matching P, such that N defeats M. Then we will have $P \succ N \succ M$.

▶ **Lemma 15.** *Given any pseudo-popular maximum-matching M, we can decide in polynomial time if there exists a non-pseudo-popular maximum-matching N such that $N \succ M$.*

Proof. Let M be the given matching. We need to check if there is a maximum matching N such that $\Delta(N, M) > 0$ where either (i) N is not S-perfect or (ii) N is S-perfect and $N \cap F \neq \emptyset$. It is straightforward to check (ii) as shown below.

First, we compute the forbidden set $F \subset E_S$ for pseudo-popular maximum-matchings by solving (LP1) for every edge in E_S. Consider the linear program (LP2) where $f \in F$ and the function $\mathsf{wt}_M : E_S \to \{0, \pm 2\}$ defined earlier (restricted to E_S) is being used here.

$$\text{maximize} \sum_{e \in E_S} \mathsf{wt}_M(e) \cdot x_e \tag{LP2}$$

subject to

$$\vec{x} \in \mathcal{S}_G \qquad \text{and} \qquad x_f = 1.$$

The feasible region of (LP2) is a face (perhaps empty) of the S-perfect matching polytope $\mathcal{S}_G$; it is the convex hull of all S-perfect matchings in G that contain the edge f. The optimal value of (LP2) is positive if and only if there is an S-perfect matching N containing f such that $\mathsf{wt}_M(N) > 0$. Since $\Delta(N, M) = \mathsf{wt}_M(N) > 0$, it follows that N defeats M.

Hence by solving (LP2) for each $f \in F$, we can determine if there exists an S-perfect matching N with $N \cap F \neq \emptyset$ that defeats M. If there is no such S-perfect matching, then we need to check condition (i), i.e., N is not S-perfect. We do this below.

Suppose there is a blocking edge (u, v) between $u \notin S$ and $v \in S$. Then the matching $N = M \oplus q$, where q is the alternating path $u - v - M(v)$, is a *non-S-perfect* maximum matching that defeats M. Thus we get a desired matching N in this case. So let us assume there is no blocking edge incident to any vertex outside S. The following claim is helpful.

▷ Claim 16. If there exists a *non-S-perfect* maximum matching that defeats M, then there is a matching N that defeats M where N leaves exactly one vertex in S unmatched.

Proof. Let T be a non-S-perfect maximum matching that defeats M. Consider $T \oplus M$: this is a collection of augmenting cycles and even length alternating paths. If there is an alternating path $\rho \in T \oplus M$ such that $\Delta(M \oplus \rho, M) > 0$ then $N = M \oplus \rho$ is our desired matching. This is because $M \oplus \rho$ matches $|S| - 1$ vertices of S and it defeats M, so we are done. So assume that for every alternating path $\rho \in T \oplus M$, we have $\Delta(M \oplus \rho, M) \leq 0$.

Let $C_1, \ldots, C_k$ be all the alternating cycles in $T \oplus M$ such that $\Delta(M \oplus C_i, M) > 0$. Note that there have to be such alternating cycles since $\Delta(T, M) > 0$ while $\Delta(M \oplus \rho, M) \leq 0$ for all alternating paths in $T \oplus M$. Let $N = M \oplus C_1 \oplus \cdots \oplus C_k \oplus q$ where q is any alternating path in $T \oplus M$. Observe that N matches $|S| - 1$ vertices of S and it defeats M. ◁

For any matching N that matches exactly $|S| - 1$ vertices of S, we have $\Delta(N, M) = \mathsf{wt}_M(N) - 1$, where $\mathsf{wt}_M : E \to \{0, \pm 2\}$ is as defined at the start of this section and the "-1" counts the vote of the vertex in S matched in M, but not in N. Observe that $\Delta(N, M) > 0$ whenever $\mathsf{wt}_M(N) > 0$ because $\mathsf{wt}_M(N) > 0$ means $\mathsf{wt}_M(N) \geq 2$ due to *even* edge weights.

Consider the linear program (LP3) where $s \in S$.

$$\text{maximize} \sum_{e\in E} \mathsf{wt}_M(e)\cdot x_e \tag{LP3}$$

subject to

$$\sum_{e\in\delta(v)} x_e = 1 \quad \forall v \in S\setminus\{s\}, \qquad \sum_{e\in\delta(v)} x_e \le 1 \quad \forall v\in A\cup B,$$

$$\sum_{e\in\delta(s)} x_e = 0, \qquad \text{and} \qquad x_e \ge 0 \quad \forall e\in E.$$

The feasible region of (LP3) is the convex hull of edge incidence vectors of all matchings that match every vertex in $S \setminus \{s\}$. Observe that any such matching N has size at least $\lceil(|S|-1)/2\rceil = |S|/2$ because $|S|$ is even, so N is a maximum matching. Thus the feasible region of (LP3) is a face (perhaps empty) of the maximum matching polytope of G.

By solving (LP3) for each $s \in S$, we can determine in polynomial time if there is a matching N – which matches all but one vertex of S – that defeats M. Claim 16 tells us this check is sufficient to determine if there is any non-S-perfect matching that defeats M. Thus by solving (LP2) for all $f \in F$ and (LP3) for all $s \in S$, we find a non-pseudo-popular maximum-matching N that defeats M, if such a matching exists. ◀

Once we have a non-pseudo-popular maximum-matching N such that $N \succ M$, as discussed earlier in this section, we can easily find a popular maximum-matching P such that $P \succ N$. Thus if there is a length-2 voting path $\langle M, N, P\rangle$, then our algorithm finds it in polynomial time. Hence we can conclude Theorem 4 stated in Section 1.

4 A Winning Set of Popular Maximum-Matchings

We prove Theorem 9 in this section. We have to find popular maximum-matchings $P_1,\ldots,P_k$ (where $k \le m$) such that any *non-pseudo-popular* maximum-matching, i.e., one defeated by some popular maximum-matching, is defeated by at least one of $P_1,\ldots,P_k$.

Let P be any S-perfect matching in G. Consider the edge weight function wt_P whose definition is analogous to wt_M at the start of Section 3. So $\mathsf{wt}_P : E \to \{0,\pm 2\}$ where $\mathsf{wt}_P(e) = 0$ if $e \in E\setminus E_S$ and $\mathsf{wt}_P(e) = 2$ if $e \in E_S$ is a blocking edge to P and so on. As discussed at the start of Section 3, for any S-perfect matching M, we have $\mathsf{wt}_P(M) = \Delta(M,P)$. Let $c = |S|/2$. So c is the size of a maximum matching in G.

Consider the edge weight function $\mathsf{wt}'_P : E \to \mathbb{Z}$ where $\mathsf{wt}'_P(e) = \mathsf{wt}_P(e) + 2(c-1)$ for $e \in E$. The linear program (LP4) computes a max-weight matching (w.r.t. wt'_P) in G and (LP5) is the dual LP.

$$\max \sum_{e\in E} \mathsf{wt}'_P(e)\cdot x_e \tag{LP4}$$

$$\text{s.t.} \quad \sum_{e\in\delta(v)} x_e \le 1 \quad \forall v\in A\cup B$$

$$x_e \ge 0 \quad \forall e \in E.$$

$$\min \sum_{v\in A\cup B} \alpha_v \tag{LP5}$$

$$\text{s.t.} \quad \alpha_a + \alpha_b \ge \mathsf{wt}'_P(a,b) \quad \forall (a,b)\in E.$$

$$\alpha_v \ge 0 \quad \forall v \in A\cup B.$$

Let P be a popular maximum-matching in G. Then its edge incidence vector is an optimal solution to (LP4) [18, Lemma 3.1].[2] An optimal solution $\vec{\alpha}$ to (LP5) will be called a *dual certificate* of P. By LP duality, we have:

$$\sum_{v \in A \cup B} \alpha_v \;=\; \mathsf{wt}'_P(P) \;=\; \mathsf{wt}_P(P) + 2(c-1)|P| \;=\; 0 + 2c(c-1) = 2c(c-1). \tag{5}$$

Equation (5) will be useful in the analysis of our algorithm given below. Let us first compute the forbidden set $F \subset E_S$ for pseudo-popular maximum-matchings by solving (LP1) for every edge in E_S.

- For each edge $f \in F$, we would like to find a popular maximum-matching P_f that defeats *every* maximum matching that contains the edge f.

But we may not necessarily be able to find such a matching P_f for each $f \in F$. Nevertheless, Algorithm 1 (see below) finds enough such pairs (f, P_f) for our purpose. In other words, Algorithm 1 finds such a matching P_f for each $f \in K$ where K is a subset of F that satisfies $K \cap M \neq \emptyset$ for any S-perfect matching M that is not pseudo-popular.

The set $\mathcal{C}$ in our algorithm will store all pairs (f, P_f), where $f \in K \subseteq F$. To begin with, the set $\mathcal{C}$ is empty and so is the set Marked.

Algorithm 1 Computing certain helpful pairs (e, P).

1: Initialize $\mathcal{C} = \mathsf{Marked} = \emptyset$.
2: **while** Marked $\neq F$ **do**
3:     let $f \in F \setminus \mathsf{Marked}$.
4:     **if** there is an S-perfect matching N such that $f \in N$ **and** $N \cap \mathsf{Marked} = \emptyset$ **then**
5:         $*$ find a popular maximum-matching P that defeats N.
6:         $*$ find a dual certificate $\vec{\alpha}$ of P.
7:         **for** every edge $e = (a, b) \in N$ such that $\alpha_a + \alpha_b > \mathsf{wt}'_P(a, b)$ **do**
8:             add the pair (e, P) to $\mathcal{C}$ and add e to Marked.
9:     **else** add f to Marked.
10: return $\mathcal{C}$.

Initially all edges in F are unmarked in Algorithm 1. For each unmarked edge $f \in F$, step 4 of the algorithm checks if there is an S-perfect matching $N \subseteq \{\text{unmarked edges}\}$ that contains f. The problem of checking if such a matching N exists (see step 4) can be solved by the linear program (LP6). Recall that $\mathcal{S}_G$ is the S-perfect matching polytope of G.

$$\text{maximize } x_f \tag{LP6}$$
$$\text{subject to}$$
$$\vec{x} \in \mathcal{S}_G \qquad \text{and} \qquad x_e \;=\; 0 \quad \forall\, e \in \mathsf{Marked}.$$

The feasible region of (LP6) is a non-empty face of $\mathcal{S}_G$ since it contains all popular maximum-matchings; so the optimal value is 0 or 1. There exists a desired matching N if and only if the optimal value of (LP6) is 1. If N exists then step 4 of Algorithm 1 finds it. Observe that N is defeated by some popular maximum-matching since a *forbidden edge* $f \in N$ (recall that $f \in F$). Thus $\max_M \Delta(M, N) > 0$ where the max is over all popular maximum-matchings M in G.

[2] The intuition behind the weight modification $\mathsf{wt}'_P(e) = \mathsf{wt}_P(e) + 2(c-1)$ is that this shift ensures $\mathsf{wt}'_P(P) \geq \mathsf{wt}'_P(M)$ for any smaller matching M. Thus popularity does not override cardinality while computing a max-weight matching w.r.t. wt'_P in (LP4).

Step 5 computes a popular maximum-matching that defeats N. This step is implemented by computing a max-weight (with respect to wt_N) popular maximum-matching P. The edge weight function wt_N is analogous to the edge weight function $\mathsf{wt}_M : E \to \{0, \pm 2\}$ defined at the start of Section 3. Observe that P defeats N since $\mathsf{wt}_N(P) = \max_M \mathsf{wt}_N(M) = \max_M \Delta(M, N) > 0$, where the max is over all popular maximum-matchings M in G.

Step 7 of Algorithm 1 deals with *slack edges* with respect to P and its dual certificate $\vec{\alpha}$.

▶ **Definition 17.** *An edge $e = (a, b) \in E$ is* slack *with respect to P and its dual certificate $\vec{\alpha}$ if $\alpha_a + \alpha_b > \mathsf{wt}'_P(a, b)$.*

All *slack* edges in N (with respect to P and its dual certificate $\vec{\alpha}$) get marked in step 8.

▶ **Lemma 18.** *Let P be the popular maximum-matching computed in step 5 of Algorithm 1. Then there is at least one edge $e \in N$ that is slack with respect to P and any dual certificate of P.*

Proof. Since P defeats N, we have $\mathsf{wt}_P(N) = \Delta(N, P) < 0$. Let $\vec{\alpha}$ be a dual certificate of P. Since $\vec{\alpha}$ is a feasible solution to (LP5), we have $\alpha_a + \alpha_b \geq \mathsf{wt}'_P(a, b)$ for every edge (a, b). Adding up the constraints $\alpha_a + \alpha_b \geq \mathsf{wt}'_P(a, b)$ for all edges $(a, b) \in N$, we get $\sum_{v \in S} \alpha_v$ on the left hand side because N is an S-perfect matching. For any vertex u left unmatched in P, we have $\alpha_u = 0$ by complementary slackness. Hence $\sum_{v \in S} \alpha_v = \sum_{v \in A \cup B} \alpha_v$ and $\sum_{v \in A \cup B} \alpha_v = \mathsf{wt}'_P(P) = 2c(c - 1)$ (by Equation (5)).

On the right hand side of $\alpha_a + \alpha_b \geq \mathsf{wt}'_P(a, b)$ added up over all edges $(a, b) \in N$ is $\mathsf{wt}'_P(N)$. We have $\mathsf{wt}'_P(N) = 2c(c - 1) + \mathsf{wt}_P(N) < 2c(c - 1)$ since $\mathsf{wt}_P(N) = \Delta(N, P) < 0$. So $\sum_{v \in S} \alpha_v = 2c(c - 1) > \mathsf{wt}'_P(N)$. Thus not all constraints $\alpha_a + \alpha_b \geq \mathsf{wt}'_P(a, b)$ for edges $(a, b) \in N$ are tight. Hence there is an edge $(a, b) \in N$ such that $\alpha_a + \alpha_b > \mathsf{wt}'_P(a, b)$. ◀

Lemma 18 ensures that if there exists a matching N that satisfies the if-condition in step 4 then at least one pair (e, P) gets added to $\mathcal{C}$ in step 8. Lemma 19 shows that any entry in $\mathcal{C}$ satisfies the following desired property.

▶ **Lemma 19.** *If e is an edge that is slack with respect to P and some dual certificate of P then $\Delta(P, M) > 0$ for any S-perfect matching M that contains e.*

Proof. Let $e = (a, b)$ be an edge that is slack with respect to P and some dual certificate $\vec{\alpha}$ of P. So $\alpha_a + \alpha_b > \mathsf{wt}'_P(a, b)$. Let M be any S-perfect matching that contains (a, b). It follows from the constraints of (LP5) that $\alpha_u + \alpha_v \geq \mathsf{wt}'_P(u, v)$ for every $(u, v) \in M$.

Adding up the inequalities $\alpha_u + \alpha_v \geq \mathsf{wt}'_P(u, v)$ for all edges $(u, v) \in M$, we get $\sum_{v \in S} \alpha_v$ on the left and $\mathsf{wt}'_P(M)$ on the right. Because one of these inequalities is strict, i.e., $\alpha_a + \alpha_b > \mathsf{wt}'_P(a, b)$, we have $\sum_{v \in S} \alpha_v > \mathsf{wt}'_P(M)$. So $\mathsf{wt}'_P(M) < \sum_{v \in S} \alpha_v \leq \sum_{v \in A \cup B} \alpha_v = 2c(c - 1)$.

Thus $\mathsf{wt}_P(M) + 2c(c - 1) = \mathsf{wt}'_P(M) < 2c(c - 1)$. So $\Delta(M, P) = \mathsf{wt}_P(M) < 0$. ◀

Hence any S-perfect matching M that contains e is defeated by P, so M is not a pseudo-popular maximum-matching. Because no pseudo-popular maximum-matching contains e, we have $e \in F$ for every edge e that gets marked in step 8.

The above analysis deals with the case when a matching N that satisfies the if-condition in step 4 exists, i.e., when the optimal value of (LP6) is 1. Suppose the optimal value of (LP6) is 0. Then the edge f gets marked in step 9 and there is no entry $(f, *)$ in the set $\mathcal{C}$. Lemma 20 shows that for any S-perfect matching M that is defeated by some popular maximum-matching, there is an entry in $\mathcal{C}$ that proves M's non-pseudo-popularity.

▶ **Lemma 20.** *Let M be any S-perfect matching in G such that some popular maximum-matching defeats M. Then $\mathcal{C}$ has an entry $(e, *)$ for at least one edge $e \in M$.*

Proof. Since M is not a pseudo-popular maximum-matching and it is S-perfect, it follows from Lemma 14 that $M \cap F \neq \emptyset$. Note that at least one unmarked edge of F gets marked in each iteration (either in step 8 or in step 9) and at the end of the algorithm, we have $\mathsf{Marked} = F$. Among all edges in $M \cap F$, let f be the *first* edge that got added to Marked in our algorithm. We claim there has to be an entry $(f, *) \in \mathcal{C}$.

Suppose not. Consider the iteration when f is added to Marked. Since there is no entry $(f, *)$ in $\mathcal{C}$ by assumption, it follows that f was not added to Marked in step 8. Thus f was added to Marked in step 9. Because the "else" part of the if-statement (step 4) got executed, we can conclude:

- f was the edge in $F \setminus \mathsf{Marked}$ that was picked in step 3.
- No matching N in G satisfied (i) N is S-perfect (ii) $f \in N$ and (iii) $N \cap \mathsf{Marked} = \emptyset$ in step 4.

Since M is S-perfect and $f \in M$ (so M satisfies two of these three properties), it has to be the case that M includes at least one edge that was already in the set Marked at that point in time (during step 4). But this contradicts the fact that f is the *first* edge that got added to Marked among the edges in M. Hence there has to be an entry $(f, *) \in \mathcal{C}$. ◀

Proof of Theorem 9. We are now ready to finish the proof of Theorem 9. Let $\mathcal{C}$ be the set computed by Algorithm 1. We will show a set $\mathcal{P}$ of popular maximum-matchings such that every *non-pseudo-popular* maximum-matching M is defeated by a matching in $\mathcal{P}$.

- If $\mathcal{C} \neq \emptyset$ then let $\mathcal{P} = \{P : (*, P) \in \mathcal{C}\}$. By Lemma 14, one of these two cases has to hold.
  1. M is S-perfect and $M \cap F \neq \emptyset$: by Lemmas 19-20, there is some matching in $\mathcal{P}$ that defeats M.
  2. M is not S-perfect: Then any popular maximum-matching defeats M (by Lemma 10); so any matching in $\mathcal{P}$ defeats M.
- Else $\mathcal{C} = \emptyset$. This means there is no S-perfect matching M in G such that $M \cap F \neq \emptyset$. So every maximum matching that is defeated by a popular maximum-matching leaves at least one vertex in S unmatched. Then any popular maximum-matching defeats all non-pseudo-popular maximum-matchings (by Lemma 10). So let $\mathcal{P} = \{P\}$, where P is the matching in G computed by the popular maximum-matching algorithm from [17].

Since $|\mathcal{P}| \leq |F| + 1 \leq m$, the set $\mathcal{P}$ consists of at most m popular maximum-matchings. This finishes the proof of Theorem 9. ◀

References

1 A. Abdulkadiroğlu and T. Sönmez. School choice: a mechanism design approach. *American Economic Review*, 93(3):729–747, 2003. doi:10.1257/000282803322157061.

2 D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007. doi:10.1137/06067328X.

3 D. J. Abraham and T. Kavitha. Voting paths. *SIAM Journal on Discrete Mathematics*, 24(2):520–537, 2007. doi:10.1137/080724125.

4 S. Baswana, P. P. Chakrabarti, S. Chandran, Y. Kanoria, and U. Patange. Centralized admissions for engineering colleges in India. *INFORMS Journal on Applied Analytics*, 49(5):338–354, 2019. doi:10.1287/inte.2019.1007.

5 S. Bhattacharya, M. Hoefer, C.-C. Huang, T. Kavitha, and L. Wagner. Maintaining near-popular matchings. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP (2))*, pages 504–515, 2015. doi:10.1007/978-3-662-47666-6_40.

6 P. Biro, R. W. Irving, and D. F. Manlove. Popular matchings in the marriage and roommates problems. In *Proceedings of the 7th International Conference on Algorithms and Complexity (CIAC)*, pages 97–108, 2010. doi:10.1007/978-3-642-13073-1_10.

7 P. Biro, D. F. Manlove, and S. Mittal. Size versus stability in the marriage problem. *Theoretical Computer Science*, 411:1828–1841, 2010. doi:10.1016/j.tcs.2010.02.003.

8 Canadian Resident Matching Service (CIMS/CaRMS). The Match. https://www.carms.ca/the-match, 1970.

9 Á. Cseh. Popular matchings. Trends in Computational Social Choice, Ulle Endriss (ed.), 2017.

10 Nicolas de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie Royale, 1785.

11 Y. Faenza, T. Kavitha, V. Powers, and X. Zhang. Popular matchings and limits to tractability. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2790–2809, 2019. doi:10.1137/1.9781611975482.173.

12 D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962. doi:10.1080/00029890.1962.11989827.

13 D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11:223–232, 1985. doi:10.1016/0166-218X(85)90074-5.

14 P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Science*, 20:166–173, 1975. doi:10.1002/bs.3830200304.

15 M. Goemans. Combinatorial optimization. https://math.mit.edu/~goemans/18453S17/polyhedral.pdf, 2017.

16 C.-C. Huang and T. Kavitha. Popular matchings in the stable marriage problem. *Information and Computation*, 222:180–194, 2013. doi:10.1016/j.ic.2012.10.012.

17 T. Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43(1):52–71, 2014. doi:10.1137/120902562.

18 T. Kavitha. Maximum matchings and popularity. *SIAM Journal on Discrete Mathematics*, 38(2):1202–1221, 2024. doi:10.1137/22M1523248.

19 T. Kavitha. Matchings, relaxed popularity, and optimality. *SIAM Journal on Discrete Mathematics*, 39(2):881–911, 2025. doi:10.1137/24M163596X.

20 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 2022: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 103–123. SIAM, 2022. arXiv:2110.10984.

21 T. Kavitha and K. Makino. Min-cost popular matchings in a hospitals/residents instance with complete preferences. https://doi.org/10.1287/moor.2024.0634, 2025. Mathematics of Operations Research, Accepted in the Special Issue on Market Design.

22 T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. *Theoretical Computer Science*, 412:2679–2690, 2011. doi:10.1016/j.tcs.2010.03.028.

23 S. Merrill and B. Grofman. *A unified theory of voting: directional and proximity spatial models*. Cambridge University Press, 1999.

24 National Resident Matching Program. Intro to the Match. https://www.nrmp.org/intro-to-the-match/, 1952.

25 U. G. Rothblum. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54:57–67, 1992. doi:10.1007/BF01586041.

26 E. G. Thurber. Concerning the maximum number of stable matchings in the stable marriage problem. *Discrete Mathematics*, 248(1-3):195–219, 2002. doi:10.1016/S0012-365X(01)00194-7.

27 J. H. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153, 1989. doi:10.1016/0167-6377(89)90041-2.

A Proof of Claim 11

▷ Claim 11. Let M be a perfect matching and $\vec{x}$ be a perfect fractional matching in G_S. Then $\Delta(M, \vec{x}) = \mathsf{wt}_x(M) = \sum_{e \in M} \mathsf{wt}_x(e)$ for the following weight function wt_x on E_S:

$$\mathsf{wt}_x(a,b) \;=\; \left(\sum_{b' \prec_a b} x_{(a,b')} - \sum_{b' \succ_a b} x_{(a,b')}\right) + \left(\sum_{a' \prec_b a} x_{(a',b)} - \sum_{a' \succ_b a} x_{(a',b)}\right) \quad \forall\, (a,b) \in E_S,$$

where for any vertex u and its neighbor v, the set $\{v' : v' \prec_u v\}$ is the set of neighbors worse than v in u's preference order and the set $\{v' : v' \succ_u v\}$ is the set of neighbors better than v.

Proof. For any vertex $u \in S$, let $\mathsf{Nbr}(u)$ be the set of u's neighbors in G. Recall that $\vec{x}$ is a perfect fractional matching in G_S. For any neighbor v of u:

$$\text{let } \mathsf{vote}_u(v, \vec{x}) = \sum_{v' \in \mathsf{Nbr}(u)} x_{(u,v')} \cdot \mathsf{vote}_u(v, v') = \sum_{v' \prec_u v} x_{(u,v')} - \sum_{v' \succ_u v} x_{(u,v')}.$$

So for any $(a,b) \in E_S$, we have:

$$\mathsf{vote}_a(b, \vec{x}) = \sum_{b' \prec_a b} x_{(a,b')} - \sum_{b' \succ_a b} x_{(a,b')} \quad \text{and} \quad \mathsf{vote}_b(a, \vec{x}) = \sum_{a' \prec_b a} x_{(a',b)} - \sum_{a' \succ_b a} x_{(a',b)}.$$

Thus $\mathsf{wt}_x(a,b)$, as given in the claim statement, equals $\mathsf{vote}_a(b, \vec{x}) + \mathsf{vote}_b(a, \vec{x})$. Hence for any S-perfect matching M, we have:

$$\begin{aligned}
\mathsf{wt}_x(M) &= \sum_{e \in M} \mathsf{wt}_x(e) \;=\; \sum_{(a,b) \in M} \mathsf{wt}_x(a,b)\\
&= \sum_{(a,b) \in M} \mathsf{vote}_a(b, \vec{x}) \;+\; \mathsf{vote}_b(a, \vec{x})\\
&= \sum_{u \in S} \mathsf{vote}_u(M, \vec{x}) \qquad \text{(because the set of vertices matched in } M \text{ is } S)\\
&= \sum_{u \in S} \sum_{v \in \mathsf{Nbr}(u)} x_{(u,v)} \cdot \mathsf{vote}_u(M, v)\\
&= \sum_{i=1}^{t} p_i \cdot \Delta(M, N_i), \quad \text{where } \vec{x} \text{ is } \sum_{i=1}^{t} p_i N_i.
\end{aligned}$$

Note that the last equality was shown in [22, Lemma 1]. Thus $\mathsf{wt}_x(M) = \Delta(M, \vec{x})$. ◁

B The Popular Maximum-Matching Algorithm: A Recap

This algorithm constructs a multigraph $G^* = (A \cup B, E^*)$. For every $e \in E$, the edge set E^* of the multigraph G^* has c copies of e where c is the size of a maximum matching in G. Let $e_0, e_1, \ldots, e_{c-1}$ denote the c copies of edge e (see Figure 2). So $E^* = \cup_{e \in E}\{e_0, e_1, \ldots, e_{c-1}\}$.

Since G^* is a multigraph, vertices have preferences over edges (rather than neighbors). In order to compare two edges e_i and e'_j incident to it, any vertex first compares their indices i and j. Each vertex in A prefers lower index edges to higher index edges. On the other hand, each vertex in B prefers higher index edges to lower index edges. For any vertex v, among edges with the same index, the preference order is as per v's original preference order.

The popular maximum-matching algorithm in [17] runs the Gale-Shapley algorithm in G^* and returns its projection in G, i.e., it drops edge indices. Claim 21 follows from [17] and proves the correctness of the above algorithm.

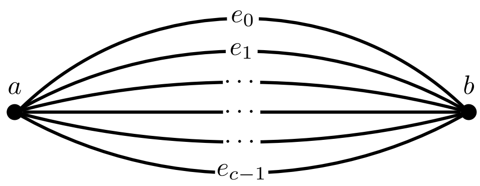

Figure 2 The graph G^* replaces every edge $e = (a, b)$ in G with c parallel edges $e_0, e_1, \ldots, e_{c-1}$.

▷ Claim 21. For any stable matching in G^*, its projection is a popular maximum-matching in G.

The converse to the above claim was shown in [18]: any popular maximum-matching M in G can be *realized* as a stable matching in G^*. That is, there exists a way of indexing the edges in M such that the resulting matching is stable in G^*. Equivalently, the projection map from {stable matchings in G^*} to {popular maximum-matchings in G} is surjective.

There are polynomial-time algorithms [25, 27] to optimize over the stable matching polytope of G^*. Thus an optimal (i.e., max-weight or min-cost) popular maximum-matching in G can be computed in polynomial time.