SIAM J. COMPUT.
Vol. 37, No. 4, pp. 1030–1045
© 2007 Society for Industrial and Applied Mathematics

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

POPULAR MATCHINGS*

DAVID J. ABRAHAM†, ROBERT W. IRVING‡, TELIKEPALLI KAVITHA§, AND KURT MEHLHORN¶

Abstract. We consider the problem of matching a set of *applicants* to a set of *posts*, where each applicant has a *preference list*, ranking a nonempty subset of posts in order of preference, possibly involving ties. We say that a matching M is *popular* if there is no matching M' such that the number of applicants preferring M' to M exceeds the number of applicants preferring M to M'. In this paper, we give the first polynomial-time algorithms to determine if an instance admits a popular matching and to find a largest such matching, if one exists. For the special case in which every preference list is strictly ordered (i.e., contains no ties), we give an $O(n+m)$ time algorithm, where n is the total number of applicants and posts and m is the total length of all of the preference lists. For the general case in which preference lists may contain ties, we give an $O(\sqrt{n}m)$ time algorithm.

Key words. matchings, bipartite graphs, one-sided preference lists

AMS subject classification. 68W40

DOI. 10.1137/06067328X

1. Introduction. An instance of the *popular matching problem* is a bipartite graph $G = (\mathcal{A} \cup \mathcal{P}, E)$ and a partition $E = E_1 \dot\cup E_2 \ldots \dot\cup E_r$ of the edge set. We call the nodes in $\mathcal{A}$ *applicants*, the nodes in $\mathcal{P}$ *posts*, and the edges in E_i the edges of rank i. If $(a,p) \in E_i$ and $(a,p') \in E_j$, with $i < j$, we say that *a prefers p to p'*. If $i = j$, we say that a is *indifferent* between p and p'. This ordering of posts adjacent to a is called a's *preference list*. We say that preference lists are *strictly ordered* if no applicant is indifferent between any two posts on his/her preference list. More generally, if applicants can be indifferent between posts, we say that preference lists contain *ties*.

A *matching* M of G is a set of edges, no two of which share an end point. A node $u \in \mathcal{A} \cup \mathcal{P}$ is either *unmatched* in M or *matched* to some node, denoted by $M(u)$ (i.e., $(u, M(u)) \in M$). We say that an applicant *a prefers* matching M' to M if (i) a is matched in M' and unmatched in M or (ii) a is matched in both M' and M, and a prefers $M'(a)$ to $M(a)$. M' is *more popular than* M, denoted by $M' \succ M$, if the number of applicants that prefer M' to M exceeds the number of applicants that prefer M to M'.

DEFINITION 1.1. *A matching M is* popular *if and only if there is no matching M' that is more popular than M.*

Example 1.1. Figure 1.1 shows the preference lists for an example instance in which $\mathcal{A} = \{a_1, a_2, a_3\}$, $\mathcal{P} = \{p_1, p_2, p_3\}$, and each applicant prefers p_1 to p_2 and p_2

*Received by the editors October 25, 2006; accepted for publication (in revised form) June 5, 2007; published electronically October 5, 2007. A preliminary version of this work appeared in *Proceedings of the* 16*th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 424–432.
http://www.siam.org/journals/sicomp/37-4/67328.html

†Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 (dabraham@cs.cmu.edu).
‡Department of Computing Science, University of Glasgow, Glasgow G12 8RZ, Scotland, UK (rwi@dcs.gla.ac.uk).
§Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India (kavitha@csa.iisc.ernet.in).
¶Department of Algorithms and Complexity, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany (mehlhorn@mpi-sb.mpg.de).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

a_1	:	p_1	p_2	p_3
a_2	:	p_1	p_2	p_3
a_3	:	p_1	p_2	p_3

FIG. 1.1. *An instance for which there is no popular matching.*

to p_3. Consider the three symmetrical matchings $M_1 = \{(a_1, p_1), (a_2, p_2), (a_3, p_3)\}$, $M_2 = \{(a_1, p_3), (a_2, p_1), (a_3, p_2)\}$, and $M_3 = \{(a_1, p_2), (a_2, p_3), (a_3, p_1)\}$. It is easy to verify that none of these matchings is popular, since $M_1 \prec M_2$, $M_2 \prec M_3$, and $M_3 \prec M_1$. In fact, this instance admits no popular matching, the problem being, of course, that the *more popular than* relation is not acyclic.

The *popular matching problem* is to determine if a given instance admits a popular matching and to find such a matching, if one exists. We remark that popular matchings may have different sizes, and a largest such matching may be smaller than a maximum-cardinality matching. The *maximum-cardinality popular matching problem* then is to determine if a given instance admits a popular matching and to find a *largest* such matching, if one exists.

In this paper, we use a novel characterization of popular matchings to give an $O(\sqrt{n}m)$ time algorithm for the maximum-cardinality popular matching problem, where n is the number of nodes, and m is the number of edges. For instances with strictly ordered preference lists, we give an $O(n+m)$ time algorithm. No polynomial time algorithms were known previously.

Related previous work. The bipartite matching problem with a graded edge set is well-studied in the economics literature; see, for example, [1, 19, 21]. It models some important real-world markets, including the allocation of graduates to training positions [10] and families to government-owned housing [20]. Instances of these markets are restrictions of stable marriage instances [5, 7], in which members of one side of the market (posts) are indifferent between members of the other side of the market (applicants).

The notion of popular matching was originally introduced by Gardenfors [6] in the context of the full stable marriage problem. It is well known that every stable marriage instance admits a weakly stable matching (one for which there is no pair who strictly prefer each other to their partners in the matching). In fact, there can be an exponential number of weakly stable matchings, and so Gardenfors considered the problem of finding one with additional desirable properties, such as popularity. Gardenfors showed that, when preference lists are strictly ordered, every stable matching is popular. He also showed that, when preference lists contain ties, there may be no popular matching.

For the problem setup considered in this paper, various other definitions of optimality have been studied. For example, a matching M is *Pareto optimal* [2, 1, 19] if there is no matching M' such that (i) some applicant prefers M' to M and (ii) no applicant prefers M to M'. In particular, such a matching has the property that no coalition of applicants can collectively improve their allocation (say, by exchanging posts with one another) without requiring some other applicant to be worse off. This is the weakest reasonable definition of optimality—see [2] for an algorithmically oriented exposition. Stronger definitions exist: A matching is *rank-maximal* [11] if it allocates the maximum number of applicants to their first choice and then, subject to this, the maximum number to their second choice, and so on. Rank-maximal matchings always exist and may be found in time $O(\min(n, C\sqrt{n})m)$ [11], where C is the maximum edge rank used in the matching. Finally, we mention *maximum-utility* matchings, which

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

maximize $\sum_{(a,p)\in M} u_{a,p}$, where $u_{a,p}$ is the utility of allocating post p to applicant a. Maximum-utility matchings can be found through an obvious transformation to the maximum-weight matching problem. Neither rank-maximal nor maximum-utility matchings are necessarily popular.

Preliminaries. For exposition purposes, we create a unique *last resort* post $l(a)$ for each applicant a and assign the edge $(a, l(a))$ higher rank than any edge incident on a. In this way, we can assume that every applicant is matched, since any unmatched applicant can be allocated to his/her last resort. From now on then, matchings are *applicant-complete*, and the size of a matching is just the number of applicants not matched to their last resort. We may also assume that instances have no gaps—so if an applicant a is incident to a rank i edge, then a is also incident to edges of all ranks smaller than i.

Organization of the paper. In section 2 we develop an alternative characterization of popular matchings, under the assumption that preference lists are strictly ordered. We then use this characterization as the basis of a linear-time algorithm to solve the maximum-cardinality popular matching problem. In section 3 we consider preference lists with ties and give an $O(\sqrt{n}m)$ time algorithm for the maximum-cardinality popular matching prblem. In section 4 we give some empirical results on the probability that a popular matching exists. Finally, the preliminary version of this paper motivated the study of several other questions related to popular matchings. We end by summarizing this recent work.

2. Strictly ordered preference lists.

In this section, we restrict our attention to strictly ordered preference lists, both to provide some intuition for the more general case and because we can solve the popular matching problem in only linear time. This last claim is not immediately clear, since Definition 1.1 potentially requires an exponential number of comparisons to even check that a given matching is popular. We begin this section then by developing an equivalent (though efficiently checkable) characterization of popular matchings.

2.1. Characterizing popular matchings.

For each applicant a, let $f(a)$ denote the first-ranked post on a's preference list (i.e., $(a, f(a)) \in E_1$). We call any such post p an *f-post* and denote by $f(p)$ the set of applicants a for which $f(a) = p$.

Example 2.1. Figure 2.1 gives the preference lists for an instance with six applicants and six posts that we shall use to illustrate the results in this section. The f-posts for this instance are p_1, p_2, and p_3, and $f(p_1) = \{a_1, a_2\}$, $f(p_2) = \{a_3, a_4, a_5\}$, and $f(p_3) = \{a_6\}$. Note that we use l_i as an abbreviation for $l(a_i)$.

The following lemma gives the first of three conditions necessarily satisfied by a popular matching.

LEMMA 2.1. *Let M be any popular matching. Then for every f-post p,* (i) *p is matched in M, and* (ii) *$M(p) \in f(p)$.*

a_1	:	p_1	p_2	p_3	l_1
a_2	:	p_1	p_5	p_4	l_2
a_3	:	p_2	p_1	p_3	l_3
a_4	:	p_2	p_3	p_6	l_4
a_5	:	p_2	p_6	p_4	l_5
a_6	:	p_3	p_2	p_5	l_6

FIG. 2.1. *An illustrative example.*

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

a_1	:	$\mathbf{p_1}$	p_2	p_3	$\underline{l_1}$
a_2	:	$\mathbf{p_1}$	$\underline{p_5}$	p_4	l_2
a_3	:	$\mathbf{p_2}$	p_1	p_3	$\underline{l_3}$
a_4	:	$\mathbf{p_2}$	p_3	$\underline{p_6}$	l_4
a_5	:	$\mathbf{p_2}$	$\underline{p_6}$	p_4	l_5
a_6	:	$\mathbf{p_3}$	p_2	$\underline{p_5}$	l_6

FIG. 2.2. *The f-posts and s-posts for the example instance.*

Proof. Every f-post p must be matched in M, for otherwise we can promote any $a \in f(p)$ to p, thereby constructing a matching more popular than M. Suppose for a contradiction then that p is matched to some $M(p) \notin f(p)$. Select any $a_1 \in f(p)$, let $a_2 = M(p)$, and since all f-posts are matched in M, let $a_3 = M(f(a_2))$. We can again construct a matching more popular than M, this time by (i) demoting a_3 to $l(a_3)$, (ii) promoting a_2 to $f(a_2)$, and then (iii) promoting a_1 to p. ∎

Example 2.2. According to Lemma 2.1, we can be sure that, if a popular matching exists for our example instance, then posts p_1, p_2, and p_3 are matched, and $M(p_1) \in \{a_1, a_2\}$, $M(p_2) \in \{a_3, a_4, a_5\}$, and $M(p_3) = a_6$.

For each applicant a, let $s(a)$ denote the first non-f-post on a's preference list (note that $s(a)$ must exist, due to the introduction of $l(a)$). We call any such post p an *s-post* and remark that f-posts are disjoint from s-posts.

Example 2.3. Figure 2.2 shows the preference lists for our example instance with the f-posts and s-posts highlighted. The bold entry in each preference list is the f-post and the underlined entry is the s-post.

In the next two lemmas, we show that a popular matching can only allocate an applicant a to either $f(a)$ or $s(a)$.

LEMMA 2.2. *Let M be any popular matching. Then for every applicant a, $M(a)$ can never be strictly between $f(a)$ and $s(a)$ on a's preference list.*

Proof. Suppose for a contradiction that $M(a)$ is strictly between $f(a)$ and $s(a)$. Since a prefers $M(a)$ to $s(a)$, we have that $M(a)$ is an f-post. Furthermore, M is a popular matching, so a belongs to $f(M(a))$ (by Lemma 2.1), thereby contradicting the assumption that a prefers $f(a)$ to $M(a)$. ∎

LEMMA 2.3. *Let M be a popular matching. Then for every applicant a, $M(a)$ is never worse than $s(a)$ on a's preference list.*

Proof. Suppose for a contradiction that a_1 prefers $s(a_1)$ to $M(a_1)$. If $s(a_1)$ is unmatched in M, we can promote a_1 to $s(a_1)$, thereby constructing a matching more popular than M. Otherwise, let $a_2 = M(s(a_1))$, and let $a_3 = M(f(a_2))$ (note that $a_2 \neq a_3$, since f-posts and s-posts are disjoint). We can again construct a matching more popular than M, this time by (i) demoting a_3 to $l(a_3)$, (ii) promoting a_2 to $f(a_2)$, and then (iii) promoting a_1 to $s(a_1)$. ∎

The three necessary conditions we have just derived form the basis of the following preliminary characterization.

LEMMA 2.4. *A matching M is popular if and only if*

(i) *every f-post is matched in M, and*

(ii) *for each applicant a, $M(a) \in \{f(a), s(a)\}$.*

Proof. Any popular matching necessarily satisfies conditions (i) and (ii) (by Lemmas 2.1–2.3). It remains to show that, together, these conditions are sufficient.

Let M be any matching satisfying (i) and (ii), and suppose for a contradiction that there is some matching M' that is more popular than M. Let a be any applicant that prefers M' to M, and let $p' = M'(a)$ (note that p' is distinct for each such a). Now, since a prefers p' to $M(a)$, it follows from condition (ii) that $M(a) = s(a)$. So,

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

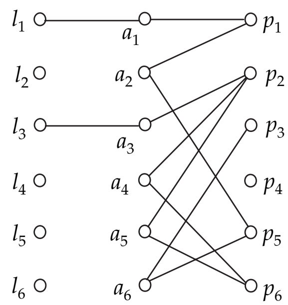

FIG. 2.3. *The reduced graph G' for the example instance.*

p' is an f-post, which by condition (i) must be matched in M, say, to a'. But then $p' = f(a')$ (by condition (ii) and since f-posts and s-posts are disjoint), and so a' prefers M to M'.

Therefore, for every applicant a that prefers M' to M, there is a distinct corresponding applicant a' that prefers M to M'. Hence, M' is not more popular than M, giving the required contradiction. ∎

Given an instance graph $G = (\mathcal{A} \cup \mathcal{P}, E)$, we define the *reduced graph* $G' = (\mathcal{A} \cup \mathcal{P}, E')$ as the subgraph of G containing two edges for each applicant a: one to $f(a)$ and the other to $s(a)$. We remark that G' need not admit an applicant-complete matching, since $l(a)$ is now isolated whenever $s(a) \neq l(a)$.

Example 2.4. Figure 2.3 shows the reduced graph for our example instance.

Lemma 2.4 gives us that M is a popular matching of G if and only if every f-post is matched in M, and M belongs to the graph G'. Recall that all popular matchings are applicant-complete through the introduction of last resorts. Hence, the following characterization is immediate.

THEOREM 2.5. *M is a popular matching of G if and only if*

(i) *every f-post is matched in M, and*

(ii) *M is an applicant-complete matching of the reduced graph G'.*

Example 2.5. By applying Theorem 2.5 to the reduced graph of Figure 2.3, it may be verified that our example instance admits four popular matchings, two of size 5 and two of size 4, as listed below. (Clearly, in the matchings of size 5, a_3 is matched with his last resort in the reduced graph, and in those of size 4, a_1 is also matched with his last resort.)

$$M_1 = \{(a_1, p_1), (a_2, p_5), (a_4, p_2), (a_5, p_6), (a_6, p_3)\},$$
$$M_2 = \{(a_1, p_1), (a_2, p_5), (a_4, p_6), (a_5, p_2), (a_6, p_3)\},$$
$$M_3 = \{(a_2, p_1), (a_4, p_2), (a_5, p_6), (a_6, p_3)\},$$
$$M_4 = \{(a_2, p_1), (a_4, p_6), (a_5, p_2), (a_6, p_3)\}.$$

2.2. Algorithmic results. Figure 2.4 contains an algorithm for solving the popular matching problem. The correctness of this algorithm follows immediately from the characterization in Theorem 2.5. We remark only that at the termination of the loop, every f-post must be matched, since $f(a)$ is unique for each applicant a, and f-posts are disjoint from s-posts. We now show a linear-time implementation of this algorithm.

It is clear that the reduced graph G' of G can be constructed in $O(n + m)$ time. G' has $O(n)$ edges, since each applicant has degree 2, and so it is also clear that the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

```
Popular matching (G = (A ∪ P, E))
    G' := reduced graph of G;
    if G' admits an applicant-complete matching M, then
        for each f-post p unmatched in M
            let a be any applicant in f(p);
            promote a to p in M;
        return M;
    else
        return "no popular matching".
```

FIG. 2.4. *Linear-time popular matching algorithm for instances with strictly ordered preference lists.*

```
Applicant-complete matching (G' = (A ∪ P, E'))
    M := ∅;
    while some post p has degree 1
        a := unique applicant adjacent to p;
        M := M ∪ {(a, p)};
        G' := G' − {a, p};    // remove a and p from G'
    while some post p has degree 0
        G' := G' − {p};
    // Every post now has degree at least 2
    // Every applicant still has degree 2
    if |P| < |A| then
        return "no applicant-complete matching";
    else
        // G' decomposes into a family of disjoint cycles
        M' := any maximum-cardinality matching of G';
        return M ∪ M'.
```

FIG. 2.5. *Linear-time algorithm for finding an applicant-complete matching in G'.*

loop phase requires only $O(n)$ time. It remains to show how we can efficiently find an applicant-complete matching of G' or determine that no such matching exists.

One approach involves computing a maximum-cardinality matching M of G' and then testing if M is applicant-complete. However, using the Hopcroft–Karp algorithm for maximum-cardinality matching [9], this would take $O(n^{3/2})$ time, which is superlinear, whenever m is $o(n^{3/2})$. Consider then the algorithm in Figure 2.5.

This algorithm begins by repeatedly matching a degree 1 post p with the unique applicant a adjacent to p. No subsequent augmenting path can include p (since it is matched and has degree 1), so we can remove both a and p from consideration. It is not hard to see that this loop can be implemented to run in $O(n)$ time, using, for example, degree counters and lazy deletion. After this, we remove any degree 0 posts, so that all remaining posts have degree at least 2, while all remaining applicants still have degree exactly 2. Now, if $|\mathcal{P}| < |\mathcal{A}|$, G' cannot admit an applicant-complete matching by Hall's marriage theorem [8]. Otherwise, we have that $|\mathcal{P}| \geq |\mathcal{A}|$, and $2|\mathcal{P}| \leq \sum_{p \in \mathcal{P}} \deg(p) = 2|\mathcal{A}|$. Hence, it must be the case that $|\mathcal{A}| = |\mathcal{P}|$, and every post has degree exactly 2. G' therefore decomposes into a family of disjoint cycles, and we need only to walk over these cycles, choosing every second edge.

We summarize the preceding discussion in the following lemma.

LEMMA 2.6. *We can find a popular matching, or determine that no such matching exists, in $O(n + m)$ time.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We now consider the maximum-cardinality popular matching problem. Let $\mathcal{A}_1$ be the set of all applicants a with $s(a) = l(a)$, and let $\mathcal{A}_2 = \mathcal{A} - \mathcal{A}_1$. Our target matching must satisfy conditions (i) and (ii) of Theorem 2.5 and, among all such matchings, allocate the fewest $\mathcal{A}_1$-applicants to their last resort.

We begin by constructing G' and testing for the existence of an applicant-complete matching M of $\mathcal{A}_2$-applicants to posts (using the applicant-complete matching algorithm in Figure 2.5). If no such M exists, then G admits no popular matching by Theorem 2.5. Otherwise, we remove all edges from G' that are incident on a last resort post and exhaustively augment M, each time matching an additional $\mathcal{A}_1$-applicant with his/her first-ranked post. If any $\mathcal{A}_1$-applicants are unmatched at this point, we simply allocate them to their last resort. Finally, we ensure that every f-post is matched, as in the popular matching algorithm in Figure 2.4. It is clear that the resulting matching is a maximum-cardinality popular matching, and so we comment only on the time complexity of augmenting M.

Note that an alternating path Q from an unmatched applicant a is completely determined (since applicants have degree 2). If we are able to augment along this path, then no subsequent augmenting path can contain a node in Q, since such a path would necessarily terminate at a, which is already matched. Otherwise, if there is no augmenting path from a, then it is not hard to see that again no subsequent augmenting path can contain a node in Q. This means we only need to visit and mark each node at most once, leading to the following result.

THEOREM 2.7. *For instances with strictly ordered preference lists, we can find a maximum-cardinality popular matching, or determine that no such matching exists, in $O(n + m)$ time.*

3. Preference lists with ties. In this section, we relax our assumption that preference lists are strictly ordered and consider problem instances with ties. We begin by developing a generalization of the popular matching characterization, similar to Theorem 2.5. Using this characterization, we then go on to give an $O(\sqrt{n}m)$ time algorithm for solving the maximum-cardinality popular matching problem. Note that we cannot hope for a linear-time algorithm here, since, for the special case where all edges have rank one, the problem of finding a popular matching is equivalent to the problem of finding a maximum-cardinality bipartite matching. Thus the popular matching problem is at least as hard as the maximum-cardinality bipartite matching problem when preference lists contain ties.

3.1. Characterizing popular matchings. For each applicant a, let $f(a)$ denote the *set* of first-ranked posts on a's preference list. Again, we refer to all such posts p as *f-posts* and denote by $f(p)$ the set of applicants a for which $p \in f(a)$.

It may no longer be possible to match *every* f-post p with an applicant in $f(p)$ (as in Lemma 2.1), since, for example, there may now be more f-posts than applicants. Below then, we work towards generalizing this key lemma.

Let M be a popular matching of some instance graph $G = (\mathcal{A} \cup \mathcal{P}, E)$. We define the *first-choice graph* of G as $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$, where E_1 is the set of all rank-one edges.

Example 3.1. Figure 3.1 gives an example instance that we use as an illustration in this section. Ties in the preference lists are indicated by parentheses.

The graph G_1 for this instance is shown in Figure 3.2.

For instances with strictly ordered preference lists, Lemma 2.1 is equivalent to requiring that every f-post is matched in $M \cap E_1$ (note that f-posts are the only posts with nonzero degree in G_1). But since applicants have a unique first choice in

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

a_1	:	$(p_1$	$p_2)$	p_4	l_1
a_2	:	p_1	$(p_2$	$p_5)$	l_2
a_3	:	p_2	$(p_4$	$p_6)$	l_3
a_4	:	p_2	p_1	p_3	l_4
a_5	:	p_4	p_3	p_2	l_5
a_6	:	$(p_5$	$p_6)$	p_1	l_6

FIG. 3.1. *An example with ties in the preference lists.*

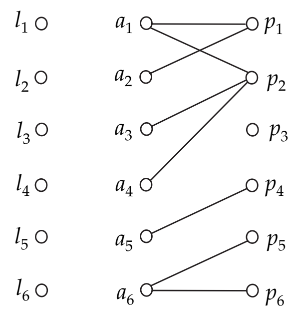

FIG. 3.2. *The graph G_1 for the example instance with ties.*

this context, Lemma 2.1 is also equivalent to the weaker condition that $M \cap E_1$ is a maximum matching of G_1. The next lemma shows that this weaker condition must also be satisfied when ties are permitted.

LEMMA 3.1. *Let M be a popular matching. Then $M \cap E_1$ is a maximum matching of G_1.*

Proof. Suppose for a contradiction that $M_1 = M \cap E_1$ is not a maximum matching of G_1. Then M_1 admits an augmenting path $Q = \langle a_1, p_1, \ldots, p_k \rangle$ with respect to G_1. It follows that $M(a_1) \notin f(a_1)$, and either p_k is unmatched in M, or $M(p_k) \notin f(p_k)$. We now show how to use Q to construct a matching M' that is more popular than M. Begin with $M' = M \setminus \{(a_1, M(a_1))\}$. There are two cases:

(i) *p_k is unmatched in M'.* Since both a_1 and p_k are unmatched in M', we augment M' with Q. In this new matching, a_1 is matched with p_1 (where $p_1 \in f(a_1)$), while all other applicants in Q remain matched to one of their first-ranked posts. Hence M' is more popular than M.

(ii) *p_k is matched in M'.* Let $a_{k+1} = M'(p_k)$, and note that $p_k \notin f(a_{k+1})$. Remove (a_{k+1}, p_k) from M', and then augment M' with Q. Select any $p_{k+1} \in f(a_{k+1})$. If p_{k+1} is unmatched in M', we promote a_{k+1} to p_{k+1}. Otherwise, we demote $a = M'(p_{k+1})$ to either $l(a)$ (if $a \neq a_1$) or back to $M(a_1)$ (if $a = a_1$), after which we can promote a_{k+1} to p_{k+1}. It is clear from this that at least one of a_1 and a_{k+1} prefers M' to M. Also, at most one applicant (that is, a) prefers M to M', though in this case *both* a_1 and a_{k+1} prefer M'. Hence, M' is more popular than M. ∎

Example 3.2. In our example, we see from Figure 3.2 and Lemma 3.1 that posts p_1, p_2, and p_4 and applicants a_5 and a_6 must be matched in any popular matching M. Furthermore, we deduce that $M(p_1) \in \{a_1, a_2\}$, $M(p_2) \in \{a_1, a_3, a_4\}$, $M(p_4) = a_5$, and $M(a_6) \in \{p_5, p_6\}$.

We now begin working towards a generalized definition of $s(a)$. For instances with strictly ordered preference lists, $s(a)$ is equivalent to the first post in a's preference

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

a_1	:	$(\mathbf{p_1}$	$\mathbf{p_2})$	p_4	$\underline{l_1}$
a_2	:	$\mathbf{p_1}$	$(p_2$	$\underline{p_5})$	l_2
a_3	:	$\mathbf{p_2}$	$(p_4$	$\underline{p_6})$	l_3
a_4	:	$\mathbf{p_2}$	p_1	$\underline{p_3}$	l_4
a_5	:	$\mathbf{p_4}$	$\underline{p_3}$	p_2	l_5
a_6	:	$(\underline{\mathbf{p_5}}$	$\underline{\mathbf{p_6}})$	p_1	l_6

FIG. 3.3. *An example with ties in the preference lists.*

list that has degree 0 in G_1. However, since Lemma 2.1 no longer holds, $s(a)$ may now contain any number of surplus f-posts. It will help us to know which f-posts *cannot* be included in $s(a)$, and for this we use the following well known ideas from bipartite matching theory.

Let M_1 be a maximum matching of some bipartite graph $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$. (Note that we are using notation that matches our use of this theory—so $M_1 = M \cap E_1$, and G_1 is the graph G restricted to rank-one edges.) Using M_1, we can partition $A \cup \mathcal{P}$ into three disjoint sets: A node v is *even* (respectively, *odd*) if there is an even (respectively, odd) length alternating path (with respect to M_1) from an unmatched node to v. Similarly, a node v is *unreachable* if there is no alternating path from an unmatched node to v. Denote by $\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ the sets of even, odd, and unreachable nodes, respectively. The Gallai–Edmonds decomposition lemma, covered in detail in [13], gives some fundamental relationships between maximum matchings and this type of node partition.

LEMMA 3.2 (Gallai–Edmonds decomposition). *Let $\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ be the node sets defined by G_1 and M_1 above. Then*

(a) *$\mathcal{E}$, $\mathcal{O}$, and $\mathcal{U}$ are pairwise disjoint. Every maximum matching in G_1 partitions the node set into the same partition of even, odd, and unreachable nodes.*

(b) *In any maximum-cardinality matching of G_1, every node in $\mathcal{O}$ is matched with some node in $\mathcal{E}$, and every node in $\mathcal{U}$ is matched with another node in $\mathcal{U}$. The size of a maximum-cardinality matching is $|\mathcal{O}| + |\mathcal{U}|/2$.*

(c) *No maximum-cardinality matching of G_1 contains an edge between two nodes in $\mathcal{O}$ or a node in $\mathcal{O}$ and a node in $\mathcal{U}$. Also, there is no edge in G_1 connecting a node in $\mathcal{E}$ with a node in $\mathcal{U}$.*

Example 3.3. In our example, it may be verified from a maximum matching, say, $\{(a_1, p_2), (a_2, p_1), (a_5, p_4), (a_6, p_5)\}$, in Figure 3.2, that $\mathcal{E} = \{a_1, a_2, a_3, a_4, p_3, p_5, p_6, l_1, l_2, l_3, l_4, l_5, l_6\}$, $\mathcal{O} = \{a_6, p_1, p_2\}$, and $\mathcal{U} = \{a_5, p_4\}$.

Now, since M_1 is a maximum-cardinality matching of G_1, Lemma 3.2(b) gives us that every odd or unreachable post p in G_1 must be matched in M to some applicant in $f(p)$. Such posts cannot be members of $s(a)$, and so we define $s(a)$ to be the set of top-ranked posts in a's preference list that are *even* in G_1 (note that $s(a) \neq \emptyset$, since $l(a)$ is always even in G_1). This definition coincides with the one in section 2, since degree 0 posts are even, and whenever every applicant has a unique first choice, posts with nonzero degree (i.e., f-posts) are odd or unreachable.

Example 3.4. Figure 3.3 displays the preference lists for our example instance, annotated as before, with the f-posts in bold and the s-posts underlined. Note that, when ties are present, f-posts and s-posts may coincide, as occurs here for applicant a_6.

Recall that our original definition of $s(a)$ led to Lemmas 2.2 and 2.3, which restrict the set of posts to which an applicant can be matched in a popular matching. We now show that the generalized definition leads to analogous results here.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

LEMMA 3.3. *Let M be a popular matching. Then for every applicant a, $M(a)$ can never be strictly between $f(a)$ and $s(a)$ on a's preference list.*

Proof. Suppose for a contradiction that $M(a)$ is strictly between $f(a)$ and $s(a)$. Then since a prefers $M(a)$ to any post in $s(a)$ and because posts in $s(a)$ are the top-ranked even nodes in G_1, it follows that $M(a)$ must be an odd or unreachable node of G_1. By Lemma 3.2(b), odd and unreachable nodes are matched in every maximum matching of G_1. But since $M(a) \notin f(a)$, $M(a)$ is unmatched in $M \cap E_1$. Hence M is not a maximum matching on rank-one edges, and so by Lemma 3.1, M is not a popular matching. ∎

LEMMA 3.4. *Let M be a popular matching. Then for every applicant a, $M(a)$ is never worse than $s(a)$ on a's preference list.*

Proof. Suppose for a contradiction that $M(a_1)$ is strictly worse than $s(a_1)$. Let p_1 be any post in $s(a_1)$. If p_1 is unmatched in M, we can promote a_1 to p_1, thereby constructing a matching more popular than M. Otherwise, let $a_2 = M(p_1)$. There are two cases:

(a) $p_1 \notin f(a_2)$. Select any post $p_2 \in f(a_2)$, and let $a_3 = M(p_2)$ (note that p_2 must be matched in M, for otherwise Lemma 3.1 is contradicted). We can again construct a matching more popular than M, this time by (i) demoting a_3 to $l(a_3)$, (ii) promoting a_2 to p_2, and then (iii) promoting a_1 to p_1.

(b) $p_1 \in f(a_2)$. Now, since $p_1 \in s(a_1)$ as well, it must be the case that p_1 is an even post in G_1. It follows then that G_1 contains (with respect to $M \cap E_1$) an even length alternating path $Q' = \langle p_1, a_2, \ldots, p_k \rangle$, where p_k is unmatched in $M \cap E_1$ (note that p_k may be matched in M though). Now, let $Q = \langle a_1, p_1, a_2, \ldots, p_k \rangle$ (i.e., a_1 followed by Q'), and let $M' = M \setminus \{(a_1, M(a_1)\}$. The remaining argument follows the proof of Lemma 3.1. If p_k is unmatched in M', $M' \oplus Q$ is more popular than M. Otherwise, p_k is matched in M'. Let $a_{k+1} = M'(p_k)$, and note that $p_k \notin f(a_{k+1})$. Remove (a_{k+1}, p_k) from M', and then augment M' with Q. Select any $p_{k+1} \in f(a_{k+1})$. If p_{k+1} is unmatched in M', we promote a_{k+1} to p_{k+1}. Otherwise, we demote $a = M'(p_{k+1})$ to either $l(a)$ (if $a \neq a_1$) or back to $M(a_1)$ (if $a = a_1$), after which we can promote a_{k+1} to p_{k+1}. It is clear from this that at least one of a_1 and a_{k+1} prefers M' to M. Also, at most one applicant (that is, a) prefers M to M', though in this case *both* a_1 and a_{k+1} prefer M'. Hence, M' is more popular than M. ∎

The three necessary conditions we have just derived form the basis of the following preliminary characterization.

LEMMA 3.5. *A matching M is popular in G if and only if*

(i) *$M \cap E_1$ is a maximum matching of G_1, and*

(ii) *for each applicant a, $M(a) \in f(a) \cup s(a)$.*

Proof. Any popular matching necessarily satisfies conditions (i) and (ii) (by Lemmas 3.1, 3.3, and 3.4). It remains to show that, together, these conditions are sufficient.

Let M be any matching satisfying conditions (i) and (ii), and suppose for a contradiction that there is some matching M' that is more popular than M. Let a be any applicant that prefers M' to M. We want to show that there is a distinct corresponding applicant a' that prefers M to M'.

The graph $H = (M \oplus M') \cap E_1$ consists of disjoint cycles and paths, each alternating between edges in $M \cap E_1$ and edges in $M' \cap E_1$. We claim that $M'(a)$ must be contained in a *nonempty path* Q of H. First, note that $M'(a)$ is an odd or unreachable node in G_1, since a prefers $M'(a)$ to $M(a)$, and $M(a) \in s(a)$ is a top-ranked even

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

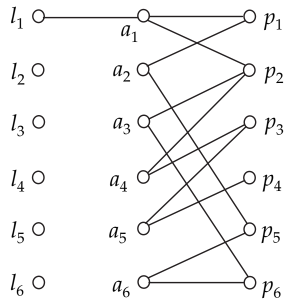

FIG. 3.4. *The reduced graph G' for the example instance with ties.*

node of G_1 in a's preference list. So by condition (i) and Lemma 3.2(b), $M'(a)$ is matched in $M \cap E_1$. However, $M'(a) \neq M(a)$, so $M'(a)$ is not isolated in H. Also, $M'(a)$ cannot be in a cycle, since a is unmatched in $M \cap E_1$. Therefore, $M'(a)$ belongs to some nonempty path Q of H.

Now, one end point of Q must be a (if $M'(a) \in f(a)$) or $M'(a)$ (otherwise). So for each such applicant a, there is a distinct nonempty path Q. Since $M'(a)$ is odd or unreachable, every post p in Q is also odd or unreachable. It follows from Lemma 3.1 that all such posts must be matched in $M \cap E_1$, and so the other end point of Q is an applicant, say, a'. It is easy to see then that a' prefers M to M', since $M(a') \in f(a')$, while $M'(a) \notin f(a')$.

Therefore, for every applicant a that prefers M' to M, there is a distinct corresponding applicant a' that prefers M to M'. Hence, M' is not more popular than M, giving the required contradiction. ∎

Given an instance graph $G = (\mathcal{A} \cup \mathcal{P}, E)$, we define the *reduced graph* $G' = (\mathcal{A} \cup \mathcal{P}, E')$ as the subgraph of G containing edges from each applicant a to posts in $f(a) \cup s(a)$. We remark that G' need not admit an applicant-complete matching, since $l(a)$ is now isolated whenever $s(a) \neq \{l(a)\}$.

Example 3.5. Figure 3.4 shows the reduced graph for our example instance.

Lemma 3.5 gives us that M is a popular matching of G if and only if M is a maximum matching on rank-one edges, and M belongs to the graph G'. Recall that all popular matchings are applicant-complete through the introduction of last resorts. Hence, the following characterization is immediate.

THEOREM 3.6. *M is a popular matching of G if and only if*

(i) *$M \cap E_1$ is a maximum matching of G_1, and*

(ii) *M is an applicant-complete matching of the reduced graph G'.*

Example 3.6. By applying Theorem 3.6 to the reduced graph of Figure 3.4, it may be verified that our example instance admits five popular matchings, two of size 6 and three of size 5, as listed below. (Clearly, in the three matchings of size 5, a_1 is matched with his last resort l_1 in the reduced graph.)

$$M_1 = \{(a_1, p_1), (a_2, p_5), (a_3, p_2), (a_4, p_3), (a_5, p_4), (a_6, p_6)\},$$

$$M_2 = \{(a_1, p_2), (a_2, p_1), (a_3, p_6), (a_4, p_3), (a_5, p_4), (a_6, p_5)\},$$

$$M_3 = \{(a_2, p_1), (a_3, p_2), (a_4, p_3), (a_5, p_4), (a_6, p_5)\},$$

$$M_4 = \{(a_2, p_1), (a_3, p_2), (a_4, p_3), (a_5, p_4), (a_6, p_6)\},$$

$$M_5 = \{(a_2, p_1), (a_3, p_6), (a_4, p_2), (a_5, p_4), (a_6, p_5)\}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Popular matching $(G = (\mathcal{A} \cup \mathcal{P}, E))$

1. Construct the graph $G' = (\mathcal{A} \cup \mathcal{P}, E')$, where $E' = \{(a, p) \mid p \in f(a) \cup s(a),\ a \in \mathcal{A}\}$.
2. Compute a maximum matching M_1 on rank-one edges; i.e., M_1 is a maximum matching in $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$.
   (M_1 is also a matching in G' because $E' \supseteq E_1$)
3. Delete all edges in G' connecting two nodes in the set $\mathcal{O}$ or a node in $\mathcal{O}$ with a node in $\mathcal{U}$, where $\mathcal{O}$ and $\mathcal{U}$ are the sets of odd and unreachable nodes of $G_1 = (\mathcal{A} \cup \mathcal{P}, E_1)$.
   Determine a maximum matching M in the modified graph G' by augmenting M_1.
4. If M is not applicant-complete, then declare that there is no popular matching in G.
   Else return M.

FIG. 3.5. *An $O(\sqrt{n}m)$ popular matching algorithm for preference lists with ties.*

3.2. Algorithmic results. In this section, we present the algorithm popular matching (see Figure 3.5) for solving the popular matching problem. This algorithm is based on the characterization given in Theorem 3.6 and is similar to the algorithm for computing a rank-maximal matching [11].

The following lemma is necessary for the correctness of our algorithm.

LEMMA 3.7. *Algorithm popular matching returns a maximum matching M on rank-one edges.*

Proof. Since M is obtained from M_1 by successive augmentations, every node matched by M_1 is also matched by M. Nodes in $\mathcal{O}$ and $\mathcal{U}$ are matched by M_1 (by Lemma 3.2(b)). Hence, nodes in $\mathcal{O}$ and $\mathcal{U}$ are matched in M.

First, we claim that G' has no edges of rank greater than one incident on nodes in $\mathcal{O}$ and nodes in $\mathcal{U} \cap \mathcal{P}$. Let us consider any odd or unreachable node in $\mathcal{P}$. This is never a candidate for $s(a)$, and hence no edge of the type $(a, p), p \in s(a)$, is incident on such a node. For odd nodes that belong to $\mathcal{A}$, it is the case that they have first-ranked posts that are even, and so $s(a) \subseteq f(a)$. This proves our claim.

So the edges that we removed in step 3 are rank-one edges, and these edges cannot be used by any maximum matching of G_1, by Lemma 3.2(c). (So no popular matching of G can use these edges.) Now the only neighbors of nodes in $\mathcal{O}$ are the even nodes of G_1 (call this set $\mathcal{E}$), and similarly, the only neighbors of nodes in $\mathcal{U} \cap \mathcal{P}$ are nodes in $\mathcal{U} \cap \mathcal{A}$ (by our edge deletions in step 3 and Lemma 3.2(c)). This means that M must match all of the nodes in $\mathcal{O}$ with nodes in $\mathcal{E}$ and all of the nodes in $\mathcal{U} \cap \mathcal{P}$ with nodes in $\mathcal{U} \cap \mathcal{A}$.

So M has at least $|\mathcal{O}| + |\mathcal{U} \cap \mathcal{P}| = |\mathcal{O}| + |\mathcal{U}|/2$ edges of rank one. So M is a maximum matching on rank-one edges (by Lemma 3.2(b)). ∎

Thus the matching returned by the algorithm popular matching is both an applicant-complete matching in G' and a maximum matching on rank-one edges. The correctness of the algorithm now follows from Theorem 3.6.

It is easy to see that the running time of our algorithm is $O(\sqrt{n}m)$: We use the algorithm of Hopcroft and Karp [9] to compute a maximum matching in G_1 and identify the set of edges E' and construct G' in $O(\sqrt{n}m)$ time. We then repeatedly augment M_1 (by the Hopcroft–Karp algorithm) to obtain M. This gives us the following result.

LEMMA 3.8. *We can find a popular matching, or determine that no such matching exists, in $O(\sqrt{n}m)$ time.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

It is now a simple matter to solve the maximum-cardinality popular matching problem. Let us assume that the instance $G = (\mathcal{A} \cup \mathcal{P}, E)$ admits a popular matching. (Otherwise, we are done.) We now want an applicant-complete matching in G' that is a maximum matching on rank-one edges and which maximizes the number of applicants not matched to their last resort.

Let M' be an arbitrary popular matching in G. We know that M' belongs to the graph G'. Remove all edges of the form $(a, l(a))$ from G' (and M'). Let H denote the resulting subgraph of G'. Note that M' is still a maximum matching on rank-one edges, since no rank-one edge has been deleted from M' or G', but M' need not be a maximum matching in the graph H. Determine a maximum matching N in H by augmenting M'. N is a matching in G' that

(i) is a maximum matching on rank-one edges and

(ii) matches the maximum number of non-last-resort posts.

N need not be a popular matching. Determine a maximum matching M in G' by augmenting N. The matching M will be applicant-complete. Since M is obtained from N by successive augmentations, all posts that are matched by N are still matched by M. Hence, it follows that M is a popular matching that maximizes the number of applicants not matched to their last resort.

The following theorem is therefore immediate.

THEOREM 3.9. *We can find a maximum-cardinality popular matching, or determine that no such matching exists, in* $O(\sqrt{n}m)$ *time.*

4. Concluding remarks. In order to obtain an idea of the probability that a popular matching exists, we performed some simulations. The factors that affect this probability are the number of applicants, the number of posts, the lengths of the preference lists, and the number, size, and position of ties in these lists.

To keep this empirical investigation manageable, we restricted our attention to cases where the numbers of applicants and posts are equal, represented by n, and all preference lists have the same length k. We characterized the ties by a single parameter t, the probability that an entry in a preference list is tied with its predecessor.

Tables 4.1 and 4.2 contain the results of simulations carried out on randomly generated instances with $n = 10$ and $n = 100$, respectively. We set t to a sequence of values in the range 0.0–0.8. For $n = 10$ we allowed k to take all possible values $(1, \ldots, 10)$, and for $n = 100$ we investigated the cases $k = 1, \ldots, 10$ and $k = 20, 30, \ldots, 100$. We generated 1000 random instances in each case. In both cases, the table shows the number of instances admitting a popular matching.

TABLE 4.1
Proportion of instances with a popular matching for $n = 10$.

		t				
		0.0	0.2	0.4	0.6	0.8
k	1	1000	1000	1000	1000	1000
	2	986	988	996	997	1000
	3	898	941	962	983	996
	4	759	846	929	979	999
	5	681	811	915	979	998
	6	636	786	888	976	1000
	7	578	737	893	978	1000
	8	565	738	909	985	1000
	9	553	759	906	980	1000
	10	556	725	890	979	1000

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Table 4.2
Proportion of instances with a popular matching for $n = 100$.

		t				
		0.0	0.2	0.4	0.6	0.8
k	1	1000	1000	1000	1000	1000
	2	997	1000	999	1000	1000
	3	884	956	985	990	1000
	4	519	807	925	946	974
	5	204	534	806	863	879
	6	64	346	685	782	798
	7	20	192	534	705	721
	8	8	90	436	628	672
	9	3	39	309	578	670
	10	2	28	243	531	675
	20	0	0	53	346	787
	30	0	0	37	302	776
	40	0	1	37	314	781
	50	0	0	44	291	791
	60	0	1	49	318	775
	70	0	2	36	304	780
	80	0	1	63	280	801
	90	0	0	38	306	776
	100	0	1	51	302	759

These results, and others not reported in detail here, give rise to the following observations:

- When $t = 0.0$, i.e., there are no ties, the likelihood of a popular matching declines rapidly as k increases and, for large n, is negligible except for very small values of k.
- Not surprisingly, increasing the value of t, and therefore the likely number and length of ties, increases the probability of a popular matching.
- For fixed n and t, increasing k initially reduces the likelihood of a popular matching, but beyond a certain range this effect all but disappears.

Thus popular matchings do exist with good probability when the chance of ties in the preference lists is high, which is likely to happen in real-world problems.

In fact, since the preliminary version of this paper [3] appeared, Mahdian [14] has shown that a popular matching exists with high probability, when (i) preference lists are randomly constructed and (ii) the number of posts is a small multiplicative factor larger than the number of applicants.

Of course, for a given instance, it still may be the case that a popular matching does not exist. Recently, McCutchen [16] considered the problem of finding a *least-unpopular* matching, where the unpopularity of a matching M is defined as the maximum ratio over all matchings M' of the number of applicants preferring M' to M to the number of applicants preferring M to M'. This definition of unpopularity makes the problem NP-hard; however, it is not clear if this is the case for other reasonable definitions.

The preliminary version also motivated the study of several other questions related to popular matchings. Manlove and Sng [15] have generalized the algorithms of sections 2.2 and 3.2 to the case where each post has an associated *capacity*, indicating the number of applicants that it can accommodate. (They described this in the equivalent context of the house allocation problem.) They gave an $O(\sqrt{C}n_1 + m)$ time algorithm for the no-ties case and a $O((\sqrt{C} + n_1)m)$ time algorithm when ties

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

are allowed, where n_1 is the number of applicants, m, as usual, is the total length of all preference lists, and C is the total capacity of all of the posts.

In [17] Mestre designed an efficient algorithm for the *weighted* popular matching problem, where each applicant is assigned a priority or weight, and the definition of popularity takes into account the priorities of the applicants. In this case the algorithm for the no-ties version has $O(n + m)$ complexity, and for the version that allows ties, the complexity is $O(\min(k\sqrt{n}, n)m)$, where k is the number of distinct weights assigned to applicants.

In [12], Kavitha and Shah give faster randomized algorithms for the popular matching problem (for problem instances where preference lists contain ties) and a weighted version of the rank-maximal matching problem. Their popular matching algorithm runs in expected time $O(n^\omega)$, where $\omega < 2.376$ is the best exponent for matrix multiplication—this algorithm reduces the popular matching problem to the bipartite perfect matching problem and uses the $O(n^\omega)$ algorithm for the latter problem [18]. The reduction works as follows: In the graph G' (the reduced graph, refer to section 3.2), let us first delete posts which are isolated; now each post in G' is either an odd or unreachable post in G_1 or it is a most preferred even post in G_1. Note that these two sets are disjoint. Let there be k_1 posts of the first type and k_2 posts of the second type. Add $k_1 + k_2 - |\mathcal{A}|$ new nodes to $\mathcal{A}$, and make each of these new nodes adjacent to each of the k_2-posts of the second type (that is, most preferred even posts in G_1). It is easy to see that there is a perfect matching in this resulting graph if and only if there is an applicant-complete matching in G' that is a maximum matching on rank-one edges. Thus it follows that the popular matching problem and the bipartite perfect matching problem have equivalent time complexities.

Finally, in the preliminary version of this paper, we described the following open problem. Suppose we have an instance that admits a popular matching, but we already have a nonpopular matching M_0 in place. Since the *more popular than* relation is not transitive, it may be that no *popular* matching is more popular than M_0. We define a *voting path* then as a sequence of matchings $\langle M_0, M_1, \ldots, M_k\rangle$ such that M_i is more popular than M_{i-1} for all $1 \le i \le k$, where M_k is popular.

Even though the *more popular than* relation is not acyclic, we were able to show that, for every matching M_0, (i) there is a voting path beginning at M_0 and (ii) the shortest such path has length at most 3. The open problem was to give an efficient algorithm for computing a shortest-length voting path from a given matching. Recently, Abraham and Kavitha [4] have shown that there is always such a voting path of length at most 2 and have given a linear-time algorithm to find one.

Acknowledgments. We thank David Manlove for directing us to previous work in the area and commenting on an early draft. We also thank Julian Mestre for correcting our description of Gardenfors' original results on popular matching.

REFERENCES

[1] A. Abdulkadiroğlu and T. Sönmez, *Random serial dictatorship and the core from random endowments in house allocation problems*, Econometrica, 66 (1998), pp. 689–701.

[2] D. J. Abraham, K. Cechlárová, D. F. Manlove, and K. Mehlhorn, *Pareto-optimality in house allocation problems*, in Proceedings of ISAAC 2004: The 15th Annual International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 3341, Springer-Verlag, Berlin, 2004, pp. 3–15.

[3] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn, *Popular matchings*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 424–432.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[4] D. J. Abraham and T. Kavitha, *Dynamic popular matchings and voting paths*, in Proceedings of SWAT 2006: The 10th Scandinavian Workshop on Algorithm Theory, Lecture Notes in Comput. Sci. 4059, Springer-Verlag, Berlin, 2006, pp. 65–76.

[5] D. Gale and L. S. Shapley, *College admissions and the stability of marriage*, Amer. Math. Monthly, 69 (1962), pp. 9–15.

[6] P. Gardenfors, *Match making: Assignments based on bilateral preferences*, Behavioural Sciences, 20 (1975), pp. 166–173.

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

[8] P. Hall, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.

[9] J. E. Hopcroft and R. M. Karp, *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.

[10] A. Hylland and R. Zeckhauser, *The efficient allocation of individuals to positions*, J. Political Economy, 87 (1979), pp. 293–314.

[11] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch, *Rank-maximal matchings*, ACM Transactions on Algorithms, 2 (2006), pp. 602–610.

[12] T. Kavitha and C. Shah, *Efficient algorithms for weighted rank-maximal matchings and related problems*, in ISAAC '06: The 17th International Symposium on Algorithms and Computation, 2006, to appear.

[13] L. Lovász and M. D. Plummer, *Matching Theory*, Ann. Discrete Math. 29, North-Holland, Amsterdam, 1986.

[14] M. Mahdian, *Random popular matchings*, in Proceedings of the 7th ACM Conference on Electronic-Commerce, 2006, pp. 238–242.

[15] D. F. Manlove and C. Sng, *Popular matchings in the capacitated house allocation problem*, in Proceedings of ESA 2006, the 14th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 4168, Springer-Verlag, Berlin, 2006, pp. 492–503.

[16] M. McCutchen, *Least-Unpopularity-Factor Matching*, manuscript, 2006.

[17] J. Mestre, *Weighted popular matchings*, in Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 4051, Springer-Verlag, Berlin, 2006, pp. 715–726.

[18] M. Mucha and P. Sankowski, *Maximum matchings via Gaussian elimination*, in Proceedings of the 45th Symposium on Foundations of Computer Science, IEEE, Piscataway, NJ, 2004, pp. 248–255.

[19] A. E. Roth and A. Postlewaite, *Weak versus strong domination in a market with indivisible goods*, J. Math. Econom., 4 (1977), pp. 131–137.

[20] Y. Yuan, *Residence exchange wanted: A stable residence exchange problem*, European J. Oper. Res., 90 (1996), pp. 536–546.

[21] L. Zhou, *On a conjecture by Gale about one-sided matching problems*, J. Econom. Theory, 52 (1990), pp. 123–135.

Downloaded 04/09/25 to 74.74.197.233 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.