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 non-empty 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 the preference lists. For the general case in which preference lists may contain ties, we give an $O(\sqrt{n}m)$ time algorithm, and show that the problem has equivalent time complexity to the maximum-cardinality bipartite matching problem.

Key words. matchings, one-sided preference lists

AMS subject classifications. 68W40

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 endpoint. 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.*

Figure 1.1 contains 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 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

[*] A preliminary version of this work appeared in the Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pages 424-432.

[†] Computer Science Department, Carnegie Mellon University, USA; dabraham@cs.cmu.edu

[‡] Department of Computing Science, University of Glasgow, UK; rwi@dcs.gla.ac.uk

[§] Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India kavitha@csa.iisc.ernet.in

[¶] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, Saarbrücken 66123, Germany;

$$\begin{array}{lclll} a_1 & : & p_1 & p_2 & p_3 \\ a_2 & : & p_1 & p_2 & p_3 \\ a_3 & : & p_1 & p_2 & p_3 \end{array}$$

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

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, 17, 19]. It models some important real-world problems, including the allocation of graduates to training positions [10], and families to government-owned housing [18]. The notion of a popular matching, also known as a *majority assignment*, was first introduced by Gardenfors [6] in the context of the stable marriage problem[1] [5, 7]. Gardenfors proved that every popular matching is stable, but also showed that popular matchings need not exist.

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, 17] 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 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 smaller ranks 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, giving an $O(\sqrt{n}m)$ time algorithm for the maximum-cardinality popular matching problem. 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 matching. 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

[1] A stable marriage instance is the same as a popular matching instance, except that *both* applicants and posts rank each other in order of preference.

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 $f(a) = p$. 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) $M(p) \in f(p)$.*

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. □

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. 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, 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 $f(a)$, the other to $s(a)$. We remark that G' need not admit an applicant-complete matching, since $l(a)$ is now isolated whenever $s(a) \neq l(a)$. Then G has no popular matching. If G admits a popular matching, the matching is contained in G', is applicant-complete, and matches every f-post by Lemma 2.4.

THEOREM 2.5. *G admits a popular matching if G' contains an applicant-complete matching that matches every f-post. Moreover, any such matching is popular in G.*

2.2. Algorithmic Results.. Figure 2.1 contains an algorithm for solving the popular matching problem. The correctness of this algorithm follows immediately from the characterization in Theorem 2.5. We only remark 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.

```
Popular-Matching($G = (\mathcal{A} \cup \mathcal{P}, E)$)
    $G'$ := 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. 2.1. *Linear-time popular matching algorithm for instances with strictly-ordered preference lists*

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 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 super-linear, whenever m is $o(n^{3/2})$. Consider then the algorithm in Figure 2.2.

```
Applicant-Complete-Matching($G' = (\mathcal{A} \cup \mathcal{P}, E')$)
    $M := \emptyset$;
    while some post $p$ has degree 1
        $a$ := unique applicant adjacent to $p$;
        $M := M \cup \{(a,p)\}$;
        $G' := G' - \{a,p\}$;    // 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 $|\mathcal{P}| < |\mathcal{A}|$ then
        return "no applicant-complete matching";
    else
        // $G'$ decomposes into a family of disjoint cycles
        $M'$ := any maximum-cardinality matching of $G'$;
        return $M \cup M'$;
```

FIG. 2.2. *Linear-time algorithm for finding an applicant-complete matching in G'*

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 only need 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.*

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 2.2). 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 2.1. It is clear that the resulting matching is a maximum-cardinality popular matching, and so we only comment 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, who 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 a $O(\sqrt{n}m)$ 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. We also remark that maximum-cardinality bipartite matching is at least as hard as popular matching, since we can simply assign all edges in the graph to have rank one. The two problems therefore have equivalent time complexity.

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. 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 non-zero degree in G_1). But since applicants have a unique first choice in 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 p_k is either 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.

□

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 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 gives some fundamental relationships between maximum matchings and this type of node partition. We include its proof for completeness.

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 $\mathcal{U}$. And there is no edge in G_1 connecting a node in $\mathcal{E}$ with a node in $\mathcal{U}$.

Proof. Assume a node v is reachable by an even length alternating path from the free node a and by an odd length alternating path from the free node b. Note that $a \neq b$ since G_1 is a bipartite graph. Then v is on the same side as a and the composition of the paths is an augmenting path from a to b. Thus M_1 is not maximum, a contradiction.

Consider any maximum matching N in G_1. Then $M_1 \oplus N$ consists of a set of alternating cycles and paths. Augmenting any such paths and cycles to M_1 leaves the odd and the unreachable nodes matched and also does not change the even / odd / unreachable status of any node.

This proves part (a).

Every node not reachable by an alternating path must be matched (otherwise it would be reachable by a path of length zero) and hence must be matched with a node which is also unreachable. Thus M_1 pairs nodes in $\mathcal{U}$ and matches the nodes in $\mathcal{O}$ with nodes in $\mathcal{E}$. (Note that there is no edge in G_1 between two nodes of $\mathcal{E}$ because nodes in $\mathcal{E}$ are reachable by alternating paths ending in a matching edge. An edge between two nodes of $\mathcal{E}$ has to be a non-matching edge and that can be used to construct an augmenting path, which contradicts the maximality of M_1.)

Thus the cardinality of M_1 is $|\mathcal{O}| + |\mathcal{U}|/2$.

This proves part (b).

Since any maximum matching pairs the nodes in $\mathcal{U}$ and matches nodes in $\mathcal{O}$ with nodes in $\mathcal{E}$, no maximum matching uses an edge connecting two odd nodes or an odd node with an unreachable node.

Since nodes in $\mathcal{E}$ are reachable by even length alternating paths from a free node, such paths end in a matching edge. An edge connecting a node in $\mathcal{E}$ to a node in $\mathcal{U}$ is non-matching and hence could be used to extend the alternating path, a contradiction to the definition of $\mathcal{U}$.

This proves part (c). □

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 non-zero degree (i.e. f-posts) are odd or unreachable.

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.

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 $Q' = \langle p_1, a_2, ..., p_k \rangle$, where p_k is unmatched in $M \cap E_1$ (note that p_k may be matched in M though). Now, let $Q = \langle a_1, p_1, a_2, ..., p_k \rangle$ (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 then 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 *non-empty 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 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 non-empty path Q of H.

Now, one endpoint 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 non-empty 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 endpoint 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 $s(a) \neq \{l(a)\}$.

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'.

3.2. Algorithmic Results..

In this section, we present algorithm Popular-Matching (see Figure 3.1) 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].

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

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 the nodes in $\mathcal{O}$ with nodes in $\mathcal{E}$ and all 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.*

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'). Call the resulting subgraph of G' as H. 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 is

(i) 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.

Table 4.1 contains 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 to 0.8. For

		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

		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

TABLE 4.1

Proportion of instances with a popular matching for $n = 10$ and $n = 100$.

$n = 10$ we allowed k to take all possible values $(1, ..., 10)$, and for $n = 100$ we investigated the cases $k = 1, ..., 10$ and $k = 20, 30, ..., 100$. We generated 1000 random instances in each case. In both cases, the table shows the number of instances admitting a popular matching.

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 [13] 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 then, McCutchen [15] 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 [14] 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 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 [16] 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 and a weighted generalized of the rank-maximal matching problem. Their popular matching algorithm runs in time $O(n^{\omega})$, where $\omega < 2.376$ is the best exponent for matrix multiplication.

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 non-popular 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 $\langle M_0, M_1, ..., M_k\rangle$ such that M_i is more popular than M_{i-1} for all $1 \leq i \leq k$, where M_k is popular.

Even though the *more popular than* relation is not acyclic, we are 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 is 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 2, and have given a linear time algorithm to find one.

Acknowledgment: We would like to thank David Manlove for directing us to previous work in the area, and commenting on an early draft.

REFERENCES

[1] A. Abdulkadiroğlu and T. Sönmez. *Random serial dictatorship and the core from random endowments in house allocation problems.* Econometrica, 66(3):689–701, 1998.

[2] D.J. Abraham, K. Cechlárová, D.F. Manlove, K. Mehlhorn. *Pareto-optimality in house allocation problems.* In *Proceedings of ISAAC 2004: the 15th Annual International Symposium on Algorithms and Computation,* (LNCS 3341), pages 3–15, Springer-Verlag, 2004.

[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*, pages 424–432, 2005.

[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,* (LNCS 4059), pages 65–76, Springer-Verlag, 2006.

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

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

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

[8] P. Hall *On representatives of subsets.* Journal of the London Mathematical Society, 10:26–30, 1935.

[9] J.E. Hopcroft and R.M. Karp. *A $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs.* SIAM Journal on Computing, 2:225–231, 1973.

[10] A. Hylland and R. Zeckhauser. *The efficient allocation of individuals to positions.* Journal of Political Economy, 87(2):293–314, 1979.

[11] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. *Rank-maximal matchings.* In Proceedings of SODA '04: the 15th ACM-SIAM Symposium on Discrete Algorithms, pages 68–75. ACM-SIAM, 2004.

[12] T. Kavitha and C. Shah. *Efficient Algorithms for Weighted Rank-Maximal Matchings and Related Problems.* To appear in ISAAC '06: the 17th International Symposium on Algorithms and Computation, 2006.

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

[14] 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*, (LNCS 4168), pages 492-503, Springer-Verlag, 2006.

[15] M. McCutchen. Least-Unpopularity-Factor Matching. Manuscript, 2006.

[16] J. Mestre. *Weighted popular matchings.* In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, (LNCS 4051), pages 715–726, 2006.

[17] A.E. Roth and A. Postlewaite. *Weak versus strong domination in a market with indivisible goods.* Journal of Mathematical Economics, 4:131–137, 1977.

[18] Y. Yuan. *Residence exchange wanted: a stable residence exchange problem.* European Journal of Operational Research, 90:536–546, 1996.

[19] L. Zhou. *On a conjecture by Gale about one-sided matching problems.* Journal of Economic Theory, 52(1):123–135, 1990.