Popular Matchings in the Stable Marriage Problem*

Chien-Chung Huang[1] and Telikepalli Kavitha[2]

[1] Humboldt-Universität zu Berlin, Germany. villars@informatik.hu-berlin.de
[2] Tata Institute of Fundamental Research, India. kavitha@tcs.tifr.res.in

Abstract. The input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex $u \in \mathcal{A} \cup \mathcal{B}$ ranks its neighbors in a strict order of preference. This is the same as an instance of the *stable marriage* problem with incomplete lists. A matching M^* is said to be popular if there is no matching M such that more vertices are better off in M than in M^*. Any stable matching of G is popular, however such a matching is a *minimum* cardinality popular matching. We consider the problem of computing a *maximum* cardinality popular matching in G.

It has very recently been shown that when preference lists have *ties*, the problem of determining if a given instance admits a popular matching or not is NP-complete. When preference lists are *strict*, popular matchings always exist, however the complexity of computing a maximum cardinality popular matching was unknown. In this paper we give a simple characterization of popular matchings when preference lists are strict and a sufficient condition for a maximum cardinality popular matching. We then show an $O(mn_0)$ algorithm for computing a maximum cardinality popular matching, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$.

1 Introduction

Our input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex ranks its neighbors in a strict order of preference. Each vertex $u \in \mathcal{A} \cup \mathcal{B}$ seeks to be assigned to one of its neighbors and u's preference is given by the ordering in u's preference list. Preference lists can be incomplete, which means that a vertex may be adjacent to only some of the vertices on the other side. (We assume without loss of generaility that a belongs to b's list if and only if b belongs to a's list, for any a and b.) Note that this is the same as an instance of the *stable marriage* problem with incomplete lists and it is customary to call the two sides of the graph *men* and *women* respectively. Let V denote the entire vertex set $\mathcal{A} \cup \mathcal{B}$ and let $|V| = n$ and $|E| = m$. We assume that no vertex is isolated, so $m \geq n/2$.

A matching M is a set of edges no two of which share an endpoint. An edge (u, v) is said to be a *blocking edge* for a matching M if by being matched to each other, both u and v are *better off* than their respective assignments in M: that is, u is either unmatched in M or prefers v to $M(u)$ and similarly, v is either unmatched in M or prefers u to $M(v)$. A matching that admits no blocking edges is called a stable matching. It is known that every instance G admits a stable matching [9] and such a matching can be computed in linear time by a straightforward generalization [5] of the Gale/Shapley algorithm [3] for complete lists.

1.1 Popular Matchings

For any two matchings M and M', we say that vertex u prefers M to M' if u is better off in M than in M' (i.e., u is either matched in M and unmatched in M' or matched in both and prefers $M(u)$ to $M'(u)$). We say that M is more popular than M', denoted by $M \succ M'$, if the number of vertices that prefer M to M' is more than the number of vertices that prefer M' to M.

* Work done when C.-C. Huang was at MPI Saarbrücken supported by a Humboldt fellowship and visited TIFR Mumbai under the IMPECS program.

Definition 1. *A matching M is popular if there is no matching that is more popular than M.*

Popularity is an attractive notion of optimality as a majority vote cannot force a migration from a popular matching. Gärdenfors [4] introduced the notion of popularity in the context of stable matchings. Popular matchings have been studied extensively during the last few years [1, 11, 10, 8, 13, 12, 7] in the case where only vertices of $\mathcal{A}$ have preferences while vertices of $\mathcal{B}$ have no preferences. Thus each edge $e = (a,b)$ in G has a rank associated with it (the rank that a assigns to b). There are simple examples in the one-sided preference lists domain that admit no popular matching. In the world of two-sided strict preference lists, popular matchings always exist since stable matchings always exist and every stable matching is popular, as we observe in the next paragraph.

When comparing a stable matching S to any matching M, note that for any edge $e \in M$, *both* the endpoints of e cannot prefer M to S - if they do, then it contradicts the stability of S. Hence if one endpoint of e prefers M to S, then the other has to prefer S to M. Thus the number of votes in favor of M is at most the number of votes in favor of S, hence M cannot be more popular than S. So popular matchings always exist in the world of two-sided strict preference lists. But not all popular matchings are stable as shown by this simple example: let $\mathcal{A} = \{a_1, a_2\}$ and $\mathcal{B} = \{b_1, b_2\}$ and let the preference lists be as shown in Fig. 1.

$a_1 : b_1 \;\; b_2$	$b_1 : a_1 \;\; a_2$
$a_2 : b_1$	$b_2 : a_1$

Fig. 1. Here a_1's top choice is b_1 and second choice is b_2 while b_1's top choice is a_1 and second choice is a_2; for a_2, the only neighbor is b_1 and for b_2, the only neighbor is a_1.

In this instance, the matching $\{(a_1, b_1)\}$ is the only stable matching, while $\{(a_1, b_2), (a_2, b_1)\}$ is popular but unstable. Thus the containment $\{$stable matchings$\} \subseteq \{$ popular matchings$\}$ could be strict.

1.2 Our problem

Given $G = (\mathcal{A} \cup \mathcal{B}, E)$ with two-sided preference lists, a stable matching has usually been considered the optimal way of matching the vertices. The fact that there can be *no* blocking edge in a stable matching is a very strong condition and it is known ([5], Section 4.5.2) that all stable matchings in $G = (\mathcal{A} \cup \mathcal{B}, E)$ have the same size and match exactly the same set of vertices, let U denote this subset of V. We show in Section 2 that every popular matching has to match all the vertices in U and a stable matching is a *minimum* cardinality popular matching.

There are many problems, where it is desirable to match more than just the vertices in U, for instance, in allocating training positions to trainees or projects to students, where the total absence of blocking edges is not necessary and a more relaxed definition of stability suffices. Thus at one end of the spectrum, we have stable matchings where *no* blocking edge is permitted and whose size is the minimum among all popular matchings and at the other end, we have maximum cardinality matchings that are not stable in any sense, since the preferences of vertices play no role here. What we seek is a matching that is somewhere in between these two extremes - we are willing to weaken to some extent the notion of stability for the sake of obtaining a larger matching.

The notion of popularity captures this slightly weakened notion of stability: blocking edges are permitted in a popular matching M, nevertheless M has *overall stability* since there is no matching where more vertices are better off than in M. Hence in problems where we are ready to substitute stability with popularity, for the sake of increasing the size of the resulting matching, what we seek is a *maximum* cardinality popular matching. In other words, we want a largest matching M in G such that there is no matching where more vertices are better off than in M. There are instances (as in our example in Fig. 1) where a maximum cardinality popular matching can be twice as large as a stable matching.

Our main result is that a maximum cardinality popular matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ can be computed in $O(mn_0)$ time, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$. We now give an overview of how we obtain this result and other results here. The following definition will be useful to us:

Definition 2. *For any $u \in \mathcal{A} \cup \mathcal{B}$ and neighbors x and y of u, define u's vote between x and y as:*

$$\mathsf{vote}_u(x,y) = \begin{cases} 1 & \text{if } u \text{ prefers } x \text{ to } y \\ -1 & \text{if } u \text{ prefers } y \text{ to } x \\ 0 & \text{otherwise (i.e., } x = y). \end{cases}$$

Let M be any matching in G. Label every edge $e = (u,v)$ in $E \setminus M$ by the pair (α_e, β_e), where $\alpha_e = \mathsf{vote}_u(v, M(u))$ and $\beta_e = \mathsf{vote}_v(u, M(v))$, i.e., α_e is u's vote for v vs. $M(u)$ and β_e is v's vote for u vs. $M(v)$. Note that for any vertex u, if u is unmatched in M, then $\mathsf{vote}_u(v, M(u)) = 1$ for any neighbor v of u, since every vertex prefers being matched with any of its neighbors to being unmatched.

Any path/cycle ρ in G where alternate edges in ρ belong to M is called an alternating path/cycle with respect to M. For an alternating path ρ, if the endpoints of ρ are unmatched in M, then ρ is called an augmenting path wrt M.

Theorem 1 gives a simple characterization of popular matchings in a graph $G = (V,E)$ with strict preference lists. Note that this theorem also holds for non-bipartite graphs with strict preference lists, referred to as the *roommates* problem; however popular matchings need not always exist in the roommates problem. We prove Theorem 1 in Section 2.

Theorem 1. *Let G_M denote the subgraph of G obtained by deleting all edges from G that are labeled $(-1,-1)$ wrt M. The matching M is popular in G if and only if the following conditions hold in G_M:*

(i) There is no alternating cycle with respect to M that contains a $(1,1)$ edge.
(ii) There is no alternating path starting from an unmatched vertex that contains a $(1,1)$ edge.
(iii) There is no alternating path with respect to M that contains two or more $(1,1)$ edges.

While a stable matching forbids all $(1,1)$ edges, it is condition (iii) that allows $(1,1)$ edges in a popular matching M - at most one $(1,1)$ edge can be allowed in certain alternating paths in G_M. In addition to conditions (i)-(iii), suppose M also satisfies the following condition:

(iv) There is no augmenting path with respect to M in G_M.

We will show in Section 2 that such a matching M has to be a maximum cardinality popular matching. Note that unlike conditions (i)-(iii) that are both sufficient and necessary for a popular matching, condition (iv) is not *necessary* for a maximum cardinality popular matching (Section 2 has such an example). In fact, it is not clear if there always exists a matching that satisfies conditions (i)-(iv). We show an algorithm in Section 3 that always constructs such a matching in a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ with strict preference lists.

Our approach. Suppose we partition the vertex set $V = \mathcal{A} \cup \mathcal{B}$ into L and R, i.e., $L \dot\cup R = G$, and reorganize the graph G by placing all the vertices of L on the left and all the vertices of R on the right. Note that L and R need not be independent sets. Let M be a matching in $L \times R$, i.e., every edge of M has one endpoint in L and the other endpoint in R.

Definition 3. *Call a matching $M \subseteq L \times R$* good *with respect to (L,R) if the following two properties are satisfied:*

(1) There is no edge marked $(1,1)$ in $L \times R$.
(2) Every edge in $L \times L$ is marked $(-1,-1)$.

Property (1) of goodness states that for every $(a,b) \in L \times R$ that is not in M, either a prefers $M(a)$ to b or b prefers $M(b)$ to a or both. Property (2) of goodness states that for every e in $L \times L$, each endpoint of e prefers its partner in M to the other endpoint of e. Theorem 2 proved in Section 2 establishes the link between a good matching and the matching that we seek.

Theorem 2. *If M is a matching that is good with respect to some partition (L,R) of V and M is R-perfect, then M satisfies conditions (i)-(iv).*

Given a partition (L,R) of V, to construct a matching that satisfies property (1) of goodness is easy: the Gale/Shapley algorithm on the edge set restricted to $E \cap (L \times R)$ where vertices in L propose and those in R dispose, yields a matching $M \subseteq L \times R$ that has no edge marked $(1,1)$ in $L \times R$. To ensure that M obeys property (2), we need to come up with a suitable $L \subset V$. Additionally, M needs to be R-perfect so that we can use Theorem 2. We show the following theorem in Section 3.

Theorem 3. *A matching M that is good wrt a partition (L,R) of V and which is R-perfect can be computed in $O(mn_0)$ time, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|,|\mathcal{B}|)$.*

Finally, we show a linear time algorithm that is based on Theorem 1 to test if a given matching M in $G = (\mathcal{A} \cup \mathcal{B}, E)$ is popular or not.

1.3 Related Results

Abraham et al. [1] considered the popular matchings problem in the domain of one-sided preference lists; they described efficient algorithms to determine if a given instance admits a popular matching or not and if so, to compute one with maximum cardinality. For one-sided preference lists (both for strict lists and for lists with ties), they gave a structural characterisation of instances that admit popular matchings. The work in [1] on one-sided popular matchings was generalized to the capacitated version by Manlove and Sng [11], the weighted version by Mestre [14], and Mahdian studied random popular matchings [10]. Kavitha and Nasre [8] as well as McDermind and Irving [13] independently studied the problem of computing an optimal popular matching for strict instances where the notion of optimality is specified as a part of the input. For instances that do not admit popular matchings, McCutchen [12] considered the problem of computing a least unpopular matching and showed this problem to be NP-hard, while Kavitha, Mestre, and Nasre [7] showed the existence of popular mixed matchings and efficient algorithms for computing them.

Gärdenfors [4], who originated the notion of popular matchings, considered this problem in the domain of two-sided preference lists. When ties are allowed in preference lists here, it has recently been shown by Biró, Irving, and Manlove [2] that the problem of computing an arbitrary popular matching in the stable marriage problem is NP-hard. The complexity of the maximum cardinality popular matching problem in the stable marriage problem when preference lists are strict (recall that the popular matchings always exist here) was not known so far and we answer this question here.

2 Structural Results

Characterization of Popular Matchings. In this section we first prove Theorem 1 and then show that conditions (i)-(iv) imply a maximum cardinality popular matching. First we show that conditions (i)-(iii) imply popularity and vice-versa in the proof of Theorem 1.

Proof of Theorem 1. Suppose M is any matching in G that satisfies conditions (i)-(iii) given in Theorem 1. Let M' be any matching in G. Define $\Delta(M',M)$ as follows:

$$\Delta(M',M) = \sum_{u \in \mathcal{A} \cup \mathcal{B}} \mathsf{vote}_u(M'(u), M(u)).$$

Thus $\Delta(M',M)$ is the difference between the votes that M' gets vs. M and the votes that M gets vs. M'. Note that $M(u)$ or $M'(u)$ can also be the state of being unmatched, which is the least preferred state for any u. We have $M' \succ M$ if and only if $\Delta(M',M) > 0$. We will now show that $\Delta(M',M) \leq 0$ for all matchings M'. This will imply that M is popular.

We need to compute $\sum_u \mathsf{vote}_u(M'(u), M(u))$ now. Mark each edge $e = (u,v)$ of M' by the pair (α_e, β_e) where $\alpha_e = \mathsf{vote}_u(v, M(u))$ and $\beta_e = \mathsf{vote}_v(u, M(v))$. Suppose $\alpha_e = \beta_e = -1$. That is, both u and v are happier with their partners in M than with each other. Then we can as well assume that M' leaves u and v unmatched, i.e., we can delete the edge (u,v) from M' since this makes no difference to $\mathsf{vote}_u(M'(u), M(u))$ or $\mathsf{vote}_v(M'(v), M(v))$ because both these values were -1 to begin with and they both remain -1 after assuming that u and v are unmatched in M'. Thus in order to evaluate $\sum_u \mathsf{vote}_u(M'(u), M(u))$, we can assume that M' is a matching in the subgraph G_M. Recall that G_M is the subgraph of G obtained by deleting all edges marked $(-1,-1)$ wrt M.

Let ρ be any connected component in $M \oplus M'$. We have $\Delta(M',M) = \sum_\rho \sum_{u \in \rho} \mathsf{vote}_u(M'(u), M(u))$, where the sum is over all the components $\rho \in M \oplus M'$. For vertices u that are isolated in $M \oplus M'$, $M(u) = M'(u)$, so we need to consider only those components ρ that contain two or more vertices. Each such ρ in $M \oplus M'$ is either a cycle or a path; also $\mathsf{vote}_u(M'(u), M(u)) = \pm 1$ for each vertex u in ρ.

Let ρ be a cycle. Since every vertex in ρ is matched by M', we have

$$\sum_{u \in \rho} \mathsf{vote}_u(M'(u), M(u)) = \sum_{e=(u,v) \in \rho \cap M'} \alpha_e + \beta_e \tag{1}$$

where $\alpha_e = \mathsf{vote}_u(v, M(u))$ and $\beta_e = \mathsf{vote}_v(u, M(v))$. Note that for every edge $e \in \rho$, (α_e, β_e) is either $(1,1)$ or $(-1,1)$ or $(1,-1)$. But we are given that M satisfies condition (i) of Theorem 1. Hence there is no $(1,1)$ edge in ρ. Thus for each edge $e \in \rho \cap M'$, $\alpha_e + \beta_e = 0$ and hence $\sum_{e \in \rho \cap M'} \alpha_e + \beta_e = 0$.

Let ρ be a path. Suppose both the endpoints of ρ are matched in M'. Then Eqn. (1) holds here. Since an endpoint of ρ is free in M, by condition (ii) of Theorem 1, we have no $(1,1)$ edge wrt M in ρ. Thus for each edge $e \in \rho \cap M'$, $\alpha_e + \beta_e = 0$ and hence $\sum_{e \in \rho \cap M'} \alpha_e + \beta_e = 0$.

– Suppose exactly one endpoint of ρ is matched in M'. Then

$$\sum_{u\in\rho} \mathsf{vote}_u(M'(u), M(u)) = -1 + \sum_{e=(u,v)\in\rho\cap M'} \alpha_e + \beta_e$$

since there is one vertex that is matched in M but not in M' and that vertex prefers M to M'. Here too an endpoint of ρ is free in M, and so by condition (ii), we have no $(1,1)$ edge wrt M in ρ. Thus for each edge $e \in \rho \cap M'$, $\alpha_e + \beta_e = 0$ and hence $\sum_{u\in\rho} \mathsf{vote}_u(M'(u), M(u)) = -1$ here.

– Suppose neither endpoint of ρ is matched in M'. Then

$$\sum_{u\in\rho} \mathsf{vote}_u(M'(u), M(u)) = -2 + \sum_{e=(u,v)\in\rho\cap M'} \alpha_e + \beta_e$$

since there are two vertices that are matched in M but not in M' and those two vertices prefer M to M'. We use condition (iii) here. There can be at most one $(1,1)$ edge wrt M in ρ. Thus except for at most one edge e in $\rho \cap M'$, we have $\alpha_e + \beta_e = 0$. So $\sum_{e=(u,v)\in\rho\cap M'} \alpha_e + \beta_e \leq 2$, thus $\sum_{u\in\rho} \mathsf{vote}_u(M'(u), M(u)) \leq 0$.

We have shown that for each component ρ of $M \oplus M'$, we have $\sum_{u\in\rho} \mathsf{vote}_u(M'(u), M(u)) \leq 0$. Thus it follows that $\Delta(M', M) \leq 0$. In other words, if M satisfies properties (i)-(iii), then M is popular.

We will now show the converse. That is, if M does not satisfy one or more of conditions (i)-(iii) of Theorem 1, then M is *not* popular.

- Suppose M does not satisfy property (i). So there is a cycle C in G_M that contains a $(1,1)$ edge wrt M. It is easy to see that the matching $M \oplus C$ is more popular than M.
- Suppose M does not satisfy property (ii). Then there is an alternating path p wrt M, one of whose endpoints is unmatched in M and p contains a $(1,1)$ edge. It is again easy to see that $M \oplus p$ is more popular than M.
- Suppose M does not satisfy property (iii). Then there is an alternating path p wrt M that contains two or more $(1,1)$ edges wrt M. It is again easy to see that $M \oplus p$ is more popular than M.

This finishes the proof of Theorem 1. □

Lemma 1. *Any stable matching is a minimum cardinality popular matching in* $G = (\mathcal{A} \cup \mathcal{B}, E)$.

Proof. Let S be a stable matching in G. We know that S is popular. Let R be any matching such that $|R| < |S|$. Then one of the components of $R \oplus S$ is a path p that is augmenting with respect to R, i.e., both the endpoints of p are unmatched in R. Since the endpoints of p prefer S to R, we have

$$\sum_{u\in p} \mathsf{vote}_u(R(u), S(u)) = -2 + \sum_{e=(u,v)\in p\cap R} \alpha_e + \beta_e$$

where $\alpha_e = \mathsf{vote}_u(v, S(u))$ and $\beta_e = \mathsf{vote}_v(u, S(v))$. The main observation here is that since S is *stable*, no edge of R can be a $(1,1)$ edge. Thus for each e in $p \cap R$, we have $\alpha_e + \beta_e \leq 0$. Hence $\sum_{u\in p} \mathsf{vote}_u(R(u), S(u)) \leq -2$, in other words, $R \oplus p$ is more popular than R.

Thus no matching of size smaller than S can be popular. So S is a minimum cardinality popular matching in G. □

Recall that it is known ([5], Section 4.5.2) that all stable matchings in $G = (\mathcal{A} \cup \mathcal{B}, E)$ have the same size and match exactly the same set of vertices, let U denote this subset of vertices.

Corollary 1. *Every popular matching in G has to match all vertices in U.*

Proof. Let R be a matching that does not match some $v \in U$. Then $R \oplus S$, where S is stable, contains a path p, where v is an endpoint of p. Since no edge of R can be a $(1,1)$ edge wrt S, it is easy to see that $\sum_{u \in p} \mathsf{vote}_u(R(u), S(u)) \le -1$, in other words, $R \oplus p$ is more popular than R. $\square$

The sufficient condition. Recall *condition (iv)* stated in Section 1: *There is no augmenting path with respect to M in G_M*. We now show that a matching that satisfies conditions (i)-(iv) is what we seek.

Theorem 4. *If a popular matching M satisfies condition (iv), then M is a maximum cardinality popular matching in G.*

Proof. Since M is a popular matching, we know that M satisfies conditions (i)-(iii) of Theorem 1. Let Q be another matching in G and let $|Q| > |M|$. So $Q \oplus M$ contains an augmenting path p wrt M. We will show using condition (iv) that $Q \oplus p$ is more popular than Q. Thus no matching of size larger than $|M|$ can be popular. Hence it follows that M is a maximum cardinality popular matching in G.

Condition (iv) states that there is no augmenting path with respect to M in G_M. So the path p has to use edges outside G_M, i.e., p contains $(-1,-1)$ edges wrt M. Split the path p into subpaths $p_1, p_2, \ldots, p_t$ by removing the $(-1,-1)$ edges from p. Each of the subpaths p_i belongs to G_M.

Each of the paths $p_2, \ldots, p_{t-1}$ can have at most one $(1,1)$ edge wrt M by condition (iii). By condition (ii), neither p_1 nor p_t can contain a $(1,1)$ edge. Thus we have

$$\sum_{u \in p} \mathsf{vote}_u(Q(u), M(u)) \le 2(t-2) - 2(t-1),$$

where the first term $2(t-2)$ counts the total number of $(1,1)$ edges possible over $p_1, \ldots, p_t$ and the second term $2(t-1)$ counts all the $(-1,-1)$ edges in p (one such edge between p_i and p_{i+1}, for $i = 1, \ldots t-1$. Thus $\sum_{u \in p} \mathsf{vote}_u(Q(u), M(u)) \le -2$. In other words, $Q \oplus p$ is more popular than Q. $\square$

Note that condition (iv) is not *necessary* for a popular matching to be one of maximum cardinality, as shown by the following example.

a_1 : b_1 b_2 b_3		$b_1 : a_1$ a_2 a_3
a_2 : b_1 b_2		$b_2 : a_1$ a_2
a_3 : b_1		$b_3 : a_1$

Fig. 2. Here a_1's top choice is b_1 and second choice is b_2 and third choice is b_3 and a_2's top choice is b_1 and second choice is b_2 while a_3's only choice is b_1. The preference lists of the b_i's are symmetric.

The matching $S = \{(a_1,b_1),(a_2,b_2)\}$ is the only stable matching in the instance in Fig. 2 and this is also a maximum cardinality popular matching. However there is an augmenting path a_3-b_1-a_1-b_3 wrt S in G_S. However, there is another maximum cardinality popular matching $M = \{(a_1,b_2),(a_2,b_1)\}$ that admits no augmenting path in G_M.

An example of a stable *roommates* instance where *no* popular matching satisfies condition (iv) is given in Fig. 3. Note that this instance is the same as the example in Fig. 2 with two extra edges: (a_2,a_3) and (b_2,b_3), where x_2 is x_3's second choice and x_3 is x_2's third choice, for $x = a, b$. This instance admits a stable matching $S = \{(a_1,b_1),(a_2,b_2)\}$ indicated in bold in Fig. 3. This is the only popular matching here.

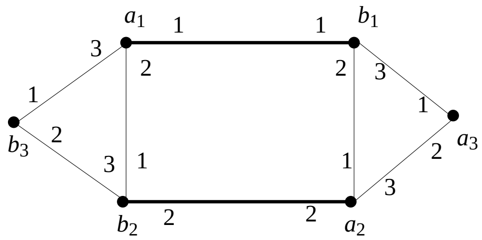

Fig. 3. The preferences of the vertices are indicated on the edges. The only popular matching here $\{(a_1,b_1),(a_2,b_2)\}$.

- The matching $M_1 = \{(a_1,b_2),(a_2,b_1)\}$ is not popular since there is an alternating path $p = b_3\text{-}b_2\text{-}a_1\text{-}b_1\text{-}a_2$ in G_{M_1} that has an unmatched vertex b_3 as an endpoint with a $(1,1)$ edge (the edge (a_1,b_1)) in it.
- The matching $M_2 = \{(a_1,b_1),(a_2,a_3),(b_2,b_3)\}$ is not popular since there is an alternating cycle $C = b_3\text{-}b_2\text{-}a_2\text{-}a_3\text{-}b_1\text{-}a_1\text{-}b_3$ in G_{M_2} that has a $(1,1)$ edge (the edge (a_2,b_2)) in it.
- The matching $M_3 = \{(a_1,b_3),(a_2,b_2),(a_3,b_1)\}$ is not popular since there is an alternating path $p' = a_1\text{-}b_2\text{-}a_2\text{-}b_1$ in G_{M_3} that has two $(1,1)$ edges (the edges (a_1,b_2) and (a_2,b_1)) in it.

In other words, we have $M_1 \prec M_2 \prec M_3 \prec M_1$ and S is the only popular matching here. However there is an augmenting path $b_3\text{-}a_1\text{-}\ b_1\text{-}a_3$ wrt S in G_S. Thus no popular matching in this instance satisfies condition (iv). However as we shall see in Section 3, in the stable *marriage* problem (i.e., G is a bipartite graph), there is always a matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ that satisfies conditions (i)-(iv).

We also show an instance $G = (\mathcal{A} \cup \mathcal{B}, E)$ below whose stable matching S has size 4 and the size of a maximum cardinality popular matching is 6. However there is no popular matching in G of size 5. The preference lists of the vertices are given below.

$a_1 : b_1$	$a_3 : b_4\ b_2\ b_3$	$a_5 : b_5\ b_4$
$b_1 : a_2\ a_1$	$b_3 : a_3$	$b_5 : a_2\ a_6\ a_5$
$a_2 : b_1\ b_5\ b_2$	$a_4 : b_4$	$a_6 : b_5\ b_6$
$b_2 : a_2\ a_3$	$b_4 : a_5\ a_3\ a_4$	$b_6 : a_6$

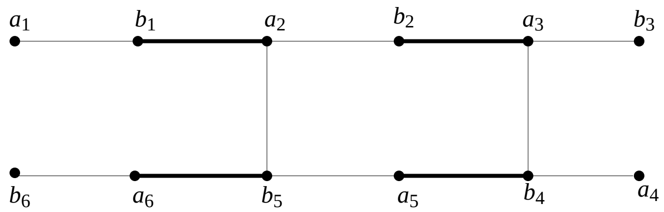

Fig. 4. The only stable marriage here is $S = \{(a_2,b_1),(b_2,a_3),(a_5,b_4),(a_6,b_5)\}$.

There are 4 augmenting paths wrt the stable matching S in G_S. These are:

– $p_1 = a_1\text{-}b_1\text{-}a_2\text{-}b_2\text{-}a_3\text{-}b_3$ – $p_2 = a_4\text{-}b_4\text{-}a_5\text{-}b_5\text{-}a_6\text{-}b_6$
– $p_3 = a_1\text{-}b_1\text{-}a_2\text{-}b_5\text{-}a_6\text{-}b_6$ – $p_4 = a_1\text{-}b_1\text{-}a_2\text{-}b_2\text{-}a_3\text{-}b_4\text{-}a_5\text{-}b_5\text{-}a_6\text{-}b_6$.

It is easy to see that none of $S \oplus p_i$ is popular, for $1 \le i \le 4$. However $S \oplus p_1 \oplus p_2$, which is $\{(a_1,b_1),(a_2,b_2),(a_3,b_3),(a_4,b_4),(a_5,b_5),(a_6,b_6)\}$, is popular. Thus there is a popular matching of size 6, however there is no popular matching of size 5, but there is one (the matching S) of size 4. That is, there are several augmenting paths with respect to S in G_S and for every augmenting path ρ, the matching $S \oplus \rho$ is unpopular. So an *augmenting path*-type technique to find a matching that satisfies conditions (i)-(iv) does not look promising and we need a new idea.

2.1 Good Matchings

Recall the definition of a good matching (Definition 3 from Section 1). Let $M \subseteq L \times R$ be a matching that is *good* with respect to (L, R), where $L \dot{\cup} R = V$. We will now prove Theorem 2 that a good matching that is R-perfect satisfies conditions (i)-(iv).

Proof of Theorem 2. Let M be a matching that is good with respect to some partition (L, R) of V and suppose M is R-perfect. Consider the graph G_M. By property (2) of goodness, the set L of vertices is independent in G_M. We now show that conditions (i)-(iv) are obeyed by M.

Condition (i). Let C be an alternating cycle with respect to M in G_M. Since $M \subseteq L \times R$, every edge in $C \cap M$ is an edge of $L \times R$. Thus the number of vertices of L that are in C equals the number of vertices of R that are in C. Since there is no edge in G_M between any pair of vertices in L, the only way an alternating cycle C can exist in G_M is that $C \subseteq L \times R$. By property (1) of goodness of M, there is no $(1,1)$ edge in $L \times R$. Hence C has no $(1,1)$ edge wrt M. Thus condition (i) is satisfied.

Condition (ii). Let $p = \langle u_0, u_1, \ldots, u_k \rangle$ be an alternating path with respect to M in G_M such that u_0 is unmatched in M. Since M is R-perfect, the vertex $u_0 \in L$. Since there are no $L \times L$ edges in G_M, the next vertex u_1 in p is in R. Since M uses only $L \times R$ edges, it follows that $u_2 = M(u_1)$ has to be in L, and u_3 is in R since u_2 has no neighbor in L and so on. Thus $p \subseteq L \times R$. Hence by property (1) of goodness of M, condition (ii) is satisfied.

Condition (iii). Let $p = \langle u_0, u_1, \ldots, u_k \rangle$ be any alternating path with respect to M in G_M. We need to show that p has at most one $(1,1)$ edge wrt M in G_M. Since it is only edges outside M that get labeled, we can assume without loss of generality that $(u_0, u_1) \notin M$. If $u_0 \in L$, then the same argument as in the earlier case (which showed that condition (ii) is satisfied) shows that $p \subseteq L \times R$ and so there is *no* $(1,1)$ edge in p.

So let us assume that $u_0 \in R$. Since there are $R \times R$ edges in G_M, there are two cases:

Case 1: Every odd indexed vertex (i.e., for every i, the vertex u_{2i+1}) is in L. Then the entire path uses only $L \times R$ edges, hence there is no $(1,1)$ edge in p.

Case 2: Not every odd indexed vertex is in L. Let u_{2j+1} be the first odd indexed vertex that is in R. That is, the edge $(u_{2j}, u_{2j+1}) \in R \times R$. Then u_{2j+2}, which is $M(u_{2j+1})$ has to be in L. Since there are no $L \times L$ edges in G_M, thereafter every odd indexed vertex u_{2k-1} of p is in R and $u_{2k} = M(u_{2k-1})$ has to be in L, so every even indexed vertex in p after u_{2j+1} is in L. Hence there can be only one $R \times R$ edge, which is (u_{2j}, u_{2j+1}), in p. Thus p has at most one $(1,1)$ edge and condition (iii) is satisfied.

Condition (iv). Suppose there exists an augmenting path $p = \langle u_0, u_1, \ldots, u_{2k+1} \rangle$ wrt M in G_M, that is, the vertices u_0 and u_{2k+1} are unmatched in M. Since M is R-perfect, $u_0 \in L$ and since there are no $L \times L$ edges, u_1 which is u_0's neighbor has to be in R. So the vertex $u_2 = M(u_1)$ is in L, and the vertex u_3 which is u_2's neighbor has to be in R, and $u_4 = M(u_3)$ has to be in L, and so on. That is, every *even* indexed vertex u_{2i} is in L and every *odd* indexed vertex u_{2i+1} is in R. Thus u_{2k+1} (the other endpoint of p) has to be in R, which contradicts that M is R-perfect, since u_{2k+1} is unmatched in M. Hence there exists no augmenting path wrt M in G_M.

This finishes the proof of Theorem 2. □

3 The Algorithm

Our job now is to find a partition (L, R) and a matching M that is good wrt this partition and which is R-perfect. The vertices in R can be viewed as the "sought-after" vertices since they are all matched in M and the vertices in L are the vertices that *seek* partners in R.

For convenience, we will refer to the elements of $\mathcal{A}$ and $\mathcal{B}$ as *men* and *women*, respectively. Let $A_0 \subset \mathcal{A}$ and $B_0 \subset \mathcal{B}$ be the sets of those men and women respectively, that are unmatched in any stable matching of $G = (\mathcal{A} \cup \mathcal{B}, E)$. Recall that every stable matching in G leaves the same vertices unmatched. Let $L_1 = A_0 \cup B_0$. Observe that L_1 is an independent set. Let $R_1 = V \setminus L_1$.

It is easy to construct a matching M_1 that is good with respect to the partition (L_1, R_1): let M_1 be the matching obtained when vertices of L_1 propose to the vertices of R_1 and vertices of R_1 dispose. That is, we run the Gale/Shapley algorithm on the "bipartite" graph obtained by placing L_1 on the left and R_1 on the right and the edge set restricted to $E \cap (L_1 \times R_1)$.

We now show that M_1 is good with respect to (L_1, R_1). Property (1) of goodness holds by the very nature of the proposal-disposal algorithm and property (2) of the goodness of any matching $M_1 \subseteq L_1 \times R_1$ is vacuously true, since L_1 is an independent set in G, and hence in G_M. If M_1 is R_1-perfect, then we are done. Otherwise we need to define a new L and show that we have made some progress.

Our algorithm is given below. We assume without loss of generality that $|\mathcal{B}| \leq |\mathcal{A}|$. Recall that we want our matching M to satisfy the following:

– M is good with respect to some partition (L, R) and
– M is R-perfect.

Algorithm 1 *Input: $G = (\mathcal{A} \cup \mathcal{B}, E)$ with strict preference lists*

1. Let S be the stable matching obtained by the proposal-disposal algorithm on $(\mathcal{A}, \mathcal{B})$. {*That is, men propose and women dispose.*}
2. Let L_1 = set of vertices left unmatched in S; let $R_1 = V \setminus L_1$.
3. $i = 1$.
4. **while** true **do**
5. compute a matching M_i by the proposal-disposal algorithm on (L_i, R_i).
6. **if** M_i is R_i-perfect **then** return M_i.
7. let $A_i \subset \mathcal{A}$ be the set of men in R_i who are unmatched in M_i.
8. set $L'_i = L_i \cup A_i$ and $R'_i = V \setminus L'_i$.
9. compute a matching M'_i by the proposal-disposal algorithm on (L'_i, R'_i).
10. **if** M'_i is R'_i-perfect **then** return M'_i.
11. let B_i be the set of vertices in R'_i left unmatched by M'_i. {*we will show that all these vertices have to be women*}
12. set $L_{i+1} = L_i \cup B_i$ and $R_{i+1} = V \setminus L_{i+1}$.
13. $i = i + 1$.
14. **end while**

We use the Gale/Shapley proposal-disposal algorithm several times in Algorithm 1. This proposal-disposal algorithm on (L, R) for any $L \subset \mathcal{A} \cup \mathcal{B}$ and $R = V \setminus L$ is given in Fig. 5. This subroutine describes the Gale/Shapley algorithm on the "bipartite" graph obtained by placing L

on the left and R on the right and the edge set restricted to $E \cap (L \times R)$; here vertices of L propose to the vertices of R and vertices of R dispose.

– $M = \emptyset$.
while there is some $u \in L$ unmatched in M who has not yet been rejected by all its neighbors in R **do**
  – u proposes to its most preferred neighbor $v \in R$ that has not rejected u.
  if v prefers u to $M(v)$ **then**
    – v assigns $M(v) = u$. $\{$*so the vertex that was v's previous partner in M is now rejected by v*$\}$
  else
    – v rejects u.
  end if
end while
– Return M.

Fig. 5. Computing a matching $M \subseteq L \times R$ with L proposing and $R = V \setminus L$ disposing

Note that in this subroutine even if there is an edge $(u,w) \in L \times L$ such that w is u's most preferred neighbor in G that has not yet rejected u, u *ignores* w and proposes to its most preferred neighbor in R that has not yet rejected u. This is because the above algorithm runs on the edge set $E \cap (L \times R)$, so edges of $L \times L$ play no part at all.

Since every unmatched $\ell \in L$ proposes in decreasing order of preference and every $r \in R$ improves in the choice of its partner whenever $M(r)$ gets reassigned, Claim 1 stated below is straightforward. This will be used in our analysis.

Claim 1 *If M is the matching returned by the Gale/Shapley proposal-disposal algorithm on (L,R), then there is no edge (ℓ,r) in $L \times R$ such that* $\mathsf{vote}_\ell(r, M(\ell)) = 1$ *and* $\mathsf{vote}_r(\ell, M(r)) = 1$.

Description of Algorithm 1. Our initial left side L_1 is the subset of vertices left unmatched in any stable matching of G. As discussed earlier, the matching M_1 obtained by running the Gale/Shapley proposal-disposal algorithm between L_1 and $R_1 = V \setminus L_1$ will be good wrt (L_1, R_1). If every vertex of R_1 receives a proposal, then we have our desired matching. Otherwise, we enter the *second stage* of the first iteration. In the second stage, we move all the unmatched *men* from R_1 to L_1 and run the proposal-disposal algorithm between the new L_1 (call this set L'_1) and the new R_1 (call this set R'_1) to compute M'_1. We will show that M'_1 is good with respect to the new left-right partition.

If M'_1 matches all the vertices on the right, then this is the desired matching. Otherwise let B_1 denote the set of unmatched vertices (women) [as is proved below] on the right who are not matched by M'_1. We set $L_2 = L_1 \cup B_1$ (our old L_1 along with B_1) and $R_2 = R_1 \setminus B_1$ (our old R_1 with B_1 deleted) and move to the next iteration of the algorithm. The unmatched men who moved from right to left in the second stage of the first iteration are back on the right now. Their purpose was to identify the set B_1.

At the start of the i-th iteration, we have a partition (L_i, R_i) of V.

- If the matching M_i that results from the proposal-disposal algorithm on (L_i, R_i) is R_i-perfect, then M_i is the desired matching.
- Else let A_i be the set of *men* in R_i who are unmatched in M_i. We run the proposal-disposal algorithm on $(L_i \cup A_i, R_i \setminus A_i)$. If the resulting matching M'_i matches all the vertices of $R_i \setminus A_i$, then M'_i is the desired matching.

– Else let B_i be the set of unmatched vertices (women) [as is proved below] on the right. We set $L_{i+1} = L_i \cup B_i$ and $R_{i+1} = R_i \setminus B_i$; the next iteration begins.

Lemma 2. *For every i, the set* $B_i \subseteq \mathcal{B}$.

Proof. The set B_i is the set of vertices of $R'_i = R_i \setminus A_i$ that are unmatched in M'_i. The matching M'_i is the result of vertices in $L'_i = L_i \cup A_i$ proposing and vertices in R'_i disposing. Note that every vertex of R'_i that was matched in M_i with vertices in L_i proposing, will remain matched in M'_i with $L'_i = L_i \cup A_i$ proposing to $R'_i = R_i \setminus A_i$.

Thus every *man* in R_i who was matched in M_i will remain matched in M'_i. Since we moved all the unmatched men of R_i (this is the set A_i) away from R_i to form $R'_i = R_i \setminus A_i$, every vertex of R'_i that is unmatched in M'_i has to be a *woman*. That is, the set B_i of vertices of R'_i that are unmatched in M'_i, is a subset of $\mathcal{B}$. □

Termination of the algorithm. It is easy to see that every iteration takes $O(m+n)$ time, which is $O(m)$. We now show that the while loop in Algorithm 1 runs for at most $|\mathcal{B}|$ iterations.

Lemma 3. *The number of while-loop iterations in Algorithm 1 is at most* $|\mathcal{B}|$.

Proof. To show that termination has to happen within the first $|\mathcal{B}|$ iterations is simple. This is because if termination does not happen in the *i*-th iteration, then $L_{i+1} \supset L_i$ because $B_i \neq \emptyset$ (otherwise termination would have happened in the *i*-th iteration). Once a woman moves to the left side of the graph, she never moves back to the right side again. Thus there is an iteration k, for some $1 \leq k \leq |\mathcal{B}|$, where either M_k is R_k-perfect or M'_k matches all the *women* in R'_k (in other words, M'_k will be R'_k-perfect). Thus the termination condition gets satisfied. Hence the algorithm terminates in the k-th iteration, for some $k \leq |\mathcal{B}|$. □

The bound of $\Theta(|\mathcal{B}|)$ is tight on the number of iterations. We show an instance $G = (\mathcal{A} \cup \mathcal{B}, E)$, where $\mathcal{A} = \{a_1, a_2, \ldots, a_{2n}\}$ and $\mathcal{B} = \{b_1, b_2, \ldots, b_{2n}\}$ on which Algorithm 1 runs for n iterations. For each $1 \leq k \leq n-1$, the preference lists are:

$$
\begin{array}{lllll}
a_{2k-1} : b_{2k-1} \;\; b_{2k} \;\; b_{2n} & \qquad & b_{2k-1} : a_{2k-1} \;\; a_{2k} \;\; a_{2n} \\
a_{2k} : \;\; b_{2k-1} \;\; b_{2k} & \qquad & b_{2k} : \;\; a_{2k-1} \;\; a_{2k}
\end{array}
$$

For the 4 vertices $a_{2n-1}, b_{2n-1}, a_{2n}$, and b_{2n}, the preference lists are:

$$
\begin{array}{lllll}
a_{2n-1} : b_{2n-1} \;\; b_{2n} & \qquad & b_{2n-1} : a_{2n-1} \;\; a_{2n} \\
a_{2n} : \;\; b_1 \;\; b_3 \;\; b_5 \cdots b_{2n-1} & \qquad & b_{2k} : \;\; a_1 \;\; a_3 \;\; a_5 \cdots a_{2n-1}
\end{array}
$$

In the above instance $S = \{(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{2n-1}, b_{2n-1})\}$ is the only stable matching. This leaves a_{2n} and b_{2n} unmatched. Algorithm 1 runs for $|\mathcal{B}|/2 = n$ iterations as shown below:

– The set $L_1 = \{a_{2n}, b_{2n}\}$. In the first round, the matching M_1 obtained is $\{(a_{2n}, b_1), (b_{2n}, a_1)\}$. Then the unmatched *men* $a_2, a_3, \ldots, a_{2n-1}$ on the right move to the left and make their proposals. The resulting matching M'_1 is $\{(a_2, b_1), (a_3, b_3), (a_4, b_4), \ldots, (a_{2n-1}, b_{2n-1}), (b_{2n}, a_1)\}$. The only vertex unmatched on the right is b_2. So we set $L_2 = L_1 \cup \{b_2\}$ and start the second round.

– In the second round when L_2 proposes to R_2, the matching $M_2 = \{(b_2,a_1),(a_{2n},b_1),(b_{2n},a_3)\}$ is obtained. Then the unmatched *men* $a_2,a_4,a_5,\ldots,a_{2n-1}$ move to the left from the right and make their proposals and the matching $M'_2 = \{(a_2,b_1),(a_1,b_2),(a_4,b_3),\ldots,(a_{2n-1},b_{2n-1}),(b_{2n},a_1)\}$ is obtained. The only vertex unmatched on the right is b_4. So we set $L_3 = L_2 \cup \{b_4\}$ and start the third round.
– At the end of the i-th round, where $1 \le i \le n-1$, the set L_{i+1} will be set to $L_i \cup \{b_{2i}\}$. In the n-th round the set $L_n = \{a_{2n},b_2,b_4,b_6,\ldots,b_{2n-2},b_{2n}\}$. When L_n proposes to $R_n = V \setminus L_n$, the matching is $M_n = \{(a_{2n},b_1),(b_2,a_1),(b_4,a_3),\ldots,(b_{2n-2},a_{2n-3}),(b_{2n},a_{2n-1})\}$. The unmatched men on the right are $a_2,a_4,\ldots,a_{2n-2}$. When these move to the left and propose, the resulting matching is $M'_n = \{(b_2,a_1),(a_2,b_1),(b_4,a_3),(a_4,b_3),\ldots,(b_{2n},a_{2n-1}),(a_{2n},b_{2n-1})\}$. All the vertices on the right are matched in M'_n, hence the algorithm returns this matching in the n-th iteration.

3.1 Correctness of Algorithm 1

We will show in this section that our algorithm maintains the following invariants:

- M_i is good with respect to (L_i,R_i).
- M'_i is good with respect to (L'_i,R'_i).

For all the matchings M_i and M'_i computed in our algorithm, property (1) of goodness is obvious since these matchings are obtained by the proposal-disposal algorithm between the left side and the right side (see Claim 1). What we need to show now is that property (2) of goodness is also obeyed by them.

We know that M_1 is good with respect to (L_1,R_1). This is because L_1 is an independent set and so property (2) of goodness is vacuously true. The next lemma shows that M'_1 also obeys property (2) of goodness.

Lemma 4. *If* $(a,b) \in L'_1 \times L'_1$, *then* $\mathsf{vote}_a(b,M'_1(a)) = -1$ *and* $\mathsf{vote}_b(a,M'_1(b)) = -1$.

Proof. Let $e = (a,b)$ be any edge in $L'_1 \times L'_1$. Since $L'_1 = A_0 \cup B_0 \cup A_1$ where $A_0 \cup B_0$ is an independent set, the vertex a has to be in A_1. Observe that every vertex of A_1 will be matched in M'_1 by virtue of the fact that the other vertices in L'_1 comprise the set of vertices unmatched in any stable matching of G. It is easy to see that $a \in A_1$ gets a partner in M'_1 that is at least as good as $S(a)$, where S is the stable matching that results from vertices in $\mathcal{A}$ proposing to vertices in $\mathcal{B}$. Recall that B_0 is the set of women unmatched in S, so a regards $S(a)$ better than any neighbor in B_0. Thus a prefers $M'_1(a)$ to all his neighbors in B_0, hence $\mathsf{vote}_a(b,M'_1(a)) = -1$.

Now we show that $\mathsf{vote}_b(a,M'_1(b)) = -1$. Recall that each man in A_1 was left *unmatched* in M_1: so $b \in B_0$ prefers $M_1(b)$ to all her neighbors in A_1. Observe that no vertex b of B_0 gets dislodged from $M_1(b)$ (a man) by the presence of A_1 in L'_1 since vertices of A_1 propose to *women*. Thus $M'_1(b) = M_1(b)$ and so $\mathsf{vote}_b(a,M'_1(b)) = -1$. This finishes the proof of the lemma. □

This proves that M'_1 is good with respect to (L'_1,R'_1). Now consider any $i \ge 2$. We assume by induction hypothesis on i that the matching $M'_{i-1} \subseteq L'_{i-1} \times R'_{i-1}$ is good with respect to (L'_{i-1},R'_{i-1}).

Lemma 5 shows that then $M_i \subseteq L_i \times R_i$ will be good with respect to (L_i,R_i).

Lemma 5. *If* $(a,b) \in L_i \times L_i$, *then* $\mathsf{vote}_a(b,M_i(a)) = -1$ *and* $\mathsf{vote}_b(a,M_i(b)) = -1$.

Proof. The set $L_i = A_0 \cup B_0 \cup B_1 \cup \cdots \cup B_{i-1}$. Let $e = (a,b) \in L_i \times L_i$. So a has to be in A_0 and $b \in B_0 \cup \cdots \cup B_{i-1}$. We need to show that every $a \in A_0$ prefers $M_i(a)$ to his neighbors in $B_0 \cup \cdots \cup B_{i-1}$ and every $b \in B_0 \cup \cdots \cup B_{i-1}$ prefers $M_i(b)$ to her neighbors in A_0.

By induction hypothesis, we know that M'_{i-1} is good with respect to (L'_{i-1}, R'_{i-1}), where the set $L'_{i-1} = A_0 \cup B_0 \cup \cdots \cup B_{i-2} \cup A_{i-1}$. So for every edge $(a,b) \in L'_{i-1} \times L'_{i-1}$, we know that a prefers $M'_{i-1}(a)$ to any neighbor b in $B_0 \cup \cdots \cup B_{i-2}$. It is easy to see that any $a \in A_0$ gets at least as good a partner in M_i as in M'_{i-1} because $L_i = (L'_{i-1} \setminus A_{i-1}) \cup B_{i-1}$.

- The presence of B_{i-1} in L_i does not hurt the men in A_0 when they propose to women in R_i because B_{i-1} is the set of women who were *unmatched* on the right when the men in A_0 were proposing in (L'_{i-1}, R'_{i-1}).
- Also, the absence of A_{i-1} on the left helps the men in A_0 as they are the only men proposing on the left now in (L_i, R_i) in comparison with (L'_{i-1}, R'_{i-1}).

Thus for any $a \in A_0$ and a's neighbor $b \in B_0 \cup \cdots \cup B_{i-2}$, $\mathsf{vote}_a(b, M_i(a)) = -1$. Also, for any $a \in A_0$ and neighbor $b \in B_{i-1}$, we know that $\mathsf{vote}_a(b, M'_{i-1}(a)) = -1$ since the vertices of B_{i-1} were unmatched in M'_{i-1}, hence $\mathsf{vote}_a(b, M_i(a)) = -1$.

Now we show that for every $(a,b) \in L_i \times L_i$, the vertex b also votes -1 for a vs $M_i(b)$. First, there are no edges between B_0 and A_0. So $b \in B_1 \cup \cdots \cup B_{i-1}$. Each woman in $B_1 \cup \cdots \cup B_{i-1}$ is matched in any stable matching of G and recall that A_0 is the set of men left unmatched in any stable matching of G. Hence when women in $B_1 \cup \cdots \cup B_{i-1}$ propose to men in $\mathcal{A} \setminus A_0$, each woman in $B_1 \cup \cdots \cup B_{i-1}$ gets matched to a man that she considers better than her neighbors in A_0. Thus for any $b \in B_1 \cup \cdots \cup B_{i-1}$ and b's neighbor $a \in A_0$, $\mathsf{vote}_b(a, M_i(b)) = -1$. This finishes the proof of this lemma. □

Suppose M_i does not match all the vertices in R_i, then we run the second stage of the i-th iteration, where all the men in R_i who were left unmatched by M_i (call this set A_i) are moved to the left. Thus $L'_i = L_i \cup A_i$. That is, $L'_i = A_0 \cup B_0 \cup \cdots \cup B_{i-1} \cup A_i$.

The proposal-disposal algorithm between L'_i and R'_i results in the matching M'_i. We will now show that property (2) of goodness also holds for M'_i. By induction hypothesis on i, we know that the matching M'_{i-1} is good with respect to (L'_{i-1}, R'_{i-1}). The following claim will be helpful to us.

Claim 2 *The set $A_i \subseteq A_{i-1}$, where A_{i-1} is the set of men in R_{i-1} left unmatched by M_{i-1}.*

Proof. The set A_{i-1} was the set of men in R_{i-1} left unmatched by M_{i-1}. Observe that every vertex in R_{i-1} that was matched with $L_{i-1} = A_0 \cup B_0 \cup \cdots \cup B_{i-2}$ proposing, will remain matched with $L_{i-1} \cup A_{i-1}$ proposing to $R_{i-1} \setminus A_{i-1}$. Thus every *man* in R_{i-1} who was matched in M_{i-1} will remain matched in M'_{i-1} with the women in $B_0 \cup \cdots \cup B_{i-2}$ proposing on the left. At the end of the $(i-1)$-th iteration, the set A_{i-1} goes back to the right and the set B_{i-1} moves to the left. With the women in $B_0 \cup \cdots \cup B_{i-1}$ proposing in the i-th iteration, all the men who were matched in the second stage of the previous iteration, continue to remain matched and some vertices of A_{i-1} also possibly get matched. So the set of men in R_i who are unmatched in M_i is a subset of A_{i-1}, that is, $A_i \subseteq A_{i-1}$. □

We will now show that for any $(a,b) \in L'_i \times L'_i$, $\mathsf{vote}_a(b, M'_i(a)) = -1$ and $\mathsf{vote}_b(a, M'_i(b)) = -1$ in Lemmas 6 and 7, respectively.

Lemma 6. *If $(a,b) \in L'_i \times L'_i$, then $\mathsf{vote}_a(b, M'_i(a)) = -1$.*

Proof. We know from Claim 2 that $A_i \subseteq A_{i-1}$. Now B_{i-1} is the set of women in R'_{i-1} left *unmatched* when vertices of L'_{i-1}, which contains $A_0 \cup A_{i-1}$, were proposing on the left in the second stage of the $(i-1)$-th iteration. Hence each man $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor $b \in B_{i-1}$. Each man in $A_0 \cup A_i$ gets at least as good a partner in M'_i as in M'_{i-1} because

– there are fewer men proposing now than in the second stage of the $(i-1)$-th iteration as $A_0 \cup A_i \subseteq A_0 \cup A_{i-1}$ and
– it is only the unmatched women who moved away from R'_{i-1}. Hence all women who belong to $\{M'_{i-1}(a) : a \in A_0 \cup A_{i-1}\}$ are still present in R'_i for $A_0 \cup A_i$ to propose to.

We know from the induction hypothesis that M'_{i-1} is good with respect to (L'_{i-1}, R'_{i-1}). Hence each $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor in $B_0 \cup \cdots \cup B_{i-2}$. Also, we just argued that $a \in A_0 \cup A_{i-1}$ prefers $M'_{i-1}(a)$ to any neighbor in B_{i-1}. Since $A_i \subseteq A_{i-1}$ and because $M'_i(a)$ is at least as good as $M'_{i-1}(a)$ for all $a \in A_0 \cup A_i$, it follows that $\mathsf{vote}_a(b, M'_i(a)) = -1$ for any edge (a,b) where $a \in A_0 \cup A_i$ and $b \in B_0 \cup \cdots \cup B_{i-2} \cup B_{i-1}$. □

Lemma 7. *If $(a,b) \in L'_i \times L'_i$, then $\mathsf{vote}_b(a, M'_i(b)) = -1$.*

Proof. Since $L'_i = A_0 \cup B_0 \cup \cdots \cup B_{i-1} \cup A_i$, for any $(a,b) \in L'_i \times L'_i$, the vertex $a \in A_0 \cup A_i$.
Case 1: Suppose $a \in A_i$. Since A_i is the set of men in R_i who are *unmatched* in M_i, it follows that each of $b \in B_0 \cup \cdots \cup B_{i-1}$ prefers $M_i(b)$ to any neighbor in A_i. Also for any $b \in B_0 \cup \cdots \cup B_{i-1}$, we have $M'_i(b) = M_i(b)$, since it is only *unmatched men* that moved from R_i to the left side to form L'_i. Thus $\mathsf{vote}_b(a, M'_i(b)) = -1$.
Case 2: Suppose $a \in A_0$. Consider any edge between a man in A_0 and a woman $b \in B_0 \cup \cdots \cup B_{i-1}$. In the first place, b has to be in $B_1 \cup \cdots \cup B_{i-1}$ since $A_0 \cup B_0$ is an independent set. Every $b \in B_1 \cup \cdots \cup B_{i-1}$ prefers her partner $M'_i(b)$ to any neighbor in A_0, since A_0 is the set of unmatched men in any stable matching of G. Thus $\mathsf{vote}_b(a, M'_i(b)) = -1$.

Hence for any $b \in B_0 \cup \cdots \cup B_{i-1}$, and any neighbor $a \in A_0 \cup A_i$, we have $\mathsf{vote}_b(a, M'_i(b)) = -1$. □

Thus property (2) of goodness is true for M'_i. We have thus shown that for every i, where $1 \le i \le$ number of iterations in our algorithm, M_i is good with respect to (L_i, R_i) and M'_i is good with respect to (L'_i, R'_i). Thus as soon as we find an M_i or an M'_i that matches all the vertices on the right, we have a good matching that matches all the vertices on the right. Lemma 3 tells us that within the first $|\mathcal{B}|$ iterations of the while loop, there is an iteration k such that either M_k or M'_k matches all the vertices on the right. This completes the proof of correctness of our algorithm.

Thus Algorithm 1 always returns a good matching M wrt a partition (L,R) such that M is R-perfect. Since the running time of Algorithm 1 is $O(m \cdot |\mathcal{B}|)$ (recall that we assumed $|\mathcal{B}| \le |\mathcal{A}|$), Theorem 3 stated in Section 1 follows. Combining all the results, we can conclude Theorem 5.

Theorem 5. *A maximum cardinality popular matching in a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists can be computed in $O(mn_0)$ time, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$.*

3.2 Testing for popularity

In this section we show a linear time algorithm to test if a matching in a stable marriage instance $G = (\mathcal{A} \cup \mathcal{B}, E)$ with strict preference lists is popular or not. This problem was previously considered in [2] where an $O(m\sqrt{n})$ algorithm was shown for this problem. Here we use Theorem 1 to design our linear time algorithm.

Our algorithm first labels each edge $e = (u,v)$ in $E \setminus M$ by $(\mathsf{vote}_u(v, M(u)), \mathsf{vote}_v(u, M(v)))$. If there exists no edge labeled $(1,1)$ in $E \setminus M$, then M is stable, hence popular. So let us assume that edges labeled $(1,1)$ exist. We first delete from G all edges labeled $(-1,-1)$ to form the graph G_M. Theorem 1 tells us if any of (a), (b), (c) given below is present in G_M, then M is unpopular; otherwise it is popular.

(a) an alternating cycle that contains an edge labeled $(1,1)$
(b) an alternating path starting from an unmatched vertex that contains an edge labeled $(1,1)$
(c) an alternating path that contains two or more edges labeled $(1,1)$.

So what we need to do in our algorithm is to check if (a), (b), or (c) exists in G_M. We build a tree similar to a *Hungarian tree* in G_M. (A Hungarian tree is typically used to find an augmenting path wrt a given matching.)

Our Algorithm. Our algorithm to check for the existence of (a), (b), or (c) in G_M is given below.

1. Mark the endpoints of all the edges labeled $(1,1)$ in G_M.
2. We build a tree or more appropriately, a layered graph, using edges of G_M as follows:
   - all the marked women are at level 0
   - the men matched to level 0 vertices are in level 1
   - the *new* neighbors of the level 1 vertices are level 2 vertices (i.e., the women in level 0 do not get repeated here)
   - the men matched to the women in level 2 are level 3 vertices
   - the neighbors of the level 3 vertices not seen so far are level 4 vertices, and so on.

   Note that for any vertex u in this tree, there is an alternating path that starts with a matched edge from a level 0 vertex to u.
3. If either a marked man or a unmatched woman is encountered in the above tree, then return "*unpopular*".
4. Build another such tree in G_M where all the marked *men* are in level 0, the vertices matched to them are in level 1, their new neighbors are in level 2, and so on. If an unmatched man is encountered in this tree, then return "*unpopular*".
5. Return "*popular*".

Claim 3 *If a marked man is encountered in Step 3, then this is evidence of (a) or (c).*

Proof. Let x be the marked man encountered in this tree. Observe that men are reached through matched edges in this tree. Thus there is an alternating path starting with a matched edge from a marked woman (call this vertex y) and ending with a matched edge in x. Both y and x are marked. Thus x has an edge labeled $(1,1)$ incident to it, so does y.

If it is the same $(1,1)$ edge that is incident to both x and y (i.e., there is a $(1,1)$ edge between x and y), then this is an alternating cycle with a $(1,1)$ edge. So suppose the $(1,1)$ edges incident on x and y are different. Since G is bipartite, x and y do not have a common neighbor, thus the $(1,1)$ edge incident on x, followed by the alternating path between x and y, followed by the $(1,1)$ edge incident on y is an alternating path with two $(1,1)$ edges. □

The following claims are straightforward.

Claim 4 *If an unmatched woman in encountered in Step 3, then this is evidence of (b).*

Claim 5 *If an unmatched man is encountered in Step 4, then this is evidence of (b).*

Thus it follows from Claims 3, 4, and 5 that whenever the algorithm returns "unpopular", the matching M is unpopular. Now we need to show that if the algorithm returns "popular", then M is indeed popular.

Lemma 8. *If an alternating cycle with a $(1,1)$ edge or an alternating path with two $(1,1)$ edges exists in G_M, then a marked man has to be encountered in Step 3 of our algorithm.*

Proof. Suppose G_M has either an alternating cycle with a $(1,1)$ edge or an alternating path with two $(1,1)$ edges in G_M. Let ρ be a shortest such cycle or path. If ρ is an alternating cycle, then let $\langle x_1, y_1, \ldots, x_k, y_k \rangle$ (the x's are men and the y's are women) denote $\rho \setminus \{e\}$, where e is the edge labeled $(1,1)$ in ρ. If ρ is an alternating path with two $(1,1)$ edges in G_M, then the first and last edges of ρ are $(1,1)$ edges and let $\langle x_1, y_1, \ldots, x_k, y_k \rangle$ denote ρ after removing these two $(1,1)$ edges.

In both cases (whether ρ is a cycle or a path), the vertices x_1 and y_k are marked. The vertex y_k, being a marked woman, is present in level 0 in our tree. The vertex x_k, being the vertex matched to y_k, is present in level 1, the vertex y_{k-1} being a neighbor of x_{k-1} has to be in level 2 (y_{k-1} cannot be in level 0 since that would contradict ρ being a shortest such cycle/path), and so on. Thus the vertex x_1, which is a marked man, will be encountered in level $2k-1$. □

Lemma 9. *If G_M has an alternating path starting from an unmatched woman that contains a $(1,1)$ edge, then an unmatched woman has to be encountered in Step 3 of our algorithm.*

Proof. Let ρ be a shortest alternating path in G_M starting from an unmatched woman that contains a $(1,1)$ edge. Since the first edge of ρ (incident on an unmatched vertex) and the last edge of ρ (a $(1,1)$ edge) are unmatched edges, ρ is of odd length. Let $\rho = \langle y_0, x_1, y_1, \ldots, x_k, y_k, x_{k+1} \rangle$, where y_0 is the unmatched woman and (y_k, x_{k+1}) is the $(1,1)$ edge. Hence y_k has to be marked and being a marked woman, it has to be present in level 0. The argument now is similar to the proof of Lemma 8. The vertex x_k is matched to y_k, so x_k is in level 1 and so on, thus the vertex y_0, an unmatched woman, will be encountered in level $2k$. □

We can symmetrically show that if G_M has an alternating path starting from an unmatched man that contains a $(1,1)$ edge, then a marked man has to be encountered in Step 4 of our algorithm. Thus it follows that if the algorithm reaches Step 5, then G_M has none of (a), (b), (c). The popularity of M now follows from Theorem 1. It is easy to see that our algorithm takes linear time. Thus we can conclude the following theorem.

Theorem 6. *Given a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists and a matching M in G, we can test if M is popular in G in linear time.*

Conclusions and Open problems. We gave a simple characterization of popular matchings in any instance G (not necessarily bipartite) with two-sided preference lists that are strictly ordered. We also showed a sufficient condition for a popular matching to be one of maximum cardinality. We introduced the notion of a "good" matching wrt a partition (L,R) of the vertex set and showed that such a matching that is also R-perfect has to be a maximum cardinality popular matching. For a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$, we gave an efficient algorithm to compute such a matching. We also showed a linear time algorithm to test if a given matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ is popular.

For non-bipartite G with strict preference lists (also called the roommates problem), the complexity of determining if G admits a popular matching or not is an open problem. For roommates instances that admit stable matchings (given a roommates instance, there is a linear time algorithm in [6] that computes a stable matching if it exists), there is no polynomial time algorithm known for computing a maximum cardinality popular matching. For testing a matching M in a roommates instance for popularity, an $O(m\sqrt{n\alpha(n,m)}\log^{3/2} n)$ algorithm was given in [2]. This algorithm uses a maximum weight matching algorithm, it is an open problem to extend our algorithm in Section 3.2 to the non-bipartite case.

Acknowledgments. We thank the reviewers of the conference version of this paper for their helpful comments.

References

1. D.J. Abraham, R.W. Irving, T. Kavitha and K. Mehlhorn. Popular matchings. SIAM Journal on Computing, Vol.37, No.4, pp. 1030–1045, 2007.
2. P. Biro, R. W. Irving, D. F. Manlove. Popular matchings in the Marriage and Roommates problems Proceedings of the seventh International Conference on Algorithms and Complexity (CIAC), 97-108, 2010.
3. D. Gale and L.S. Shapley. *College admissions and the stability of marriage*. American Mathematical Monthly, 69:9–15, 1962.
4. P. Gärdenfors. *Match making: assignments based on bilateral preferences*. Behavioural Sciences, 20:166–173, 1975.
5. D. Gusfield and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.
6. R.W. Irving. *An Efficient Algorithm for the "Stable Roommates" Problem.* Journal of Algorithms, 6: 577-595, 1985.
7. T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 574–584, 2009.
8. T. Kavitha and M. Nasre. Note: Optimal popular matchings. *Discrete Applied Mathematics*, 157(14):3181–3186, 2009.
9. D. E. Knuth. Mariages Stables. Les Presses de L'Université de Montreal, 1976.
10. M. Mahdian. Random popular matchings. In *Proceedings of the 7th ACM Conference on Electronic-Commerce*, pages 238–242, 2006.
11. 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.
12. M. McCutchen. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of LATIN 2008, the 8th Latin American Theoretical Informatics Symposium*.
13. E. McDermid and R. W. Irving. Popular matchings: Structure and algorithms. In *Proceedings of 15th Annual International Computing and Combinatorics Conference*, pages 506–515, 2009.
14. J. Mestre. *Weighted popular matchings.* In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, (LNCS 4051), pages 715–726, 2006.