SIAM J. COMPUT.
Vol. 43, No. 1, pp. 52–71

A SIZE-POPULARITY TRADEOFF IN THE STABLE MARRIAGE PROBLEM*

TELIKEPALLI KAVITHA†

Abstract. Given a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex ranks its neighbors in a strict order of preference, the problem of computing a stable matching is classical and well studied. A stable matching has size at least $\frac{1}{2}|M_{\max}|$, where $M_{\max}$ is a maximum size matching in G, and there are simple examples where this bound is tight. It is known that a stable matching is a *minimum* size popular matching. A matching M is said to be popular if there is no matching where more vertices are better off than in M. In this paper we show the first linear time algorithm for computing a maximum size popular matching in G. A maximum size popular matching is guaranteed to have size at least $\frac{2}{3}|M_{\max}|$, and this bound is tight. We then consider the following problem: is there a maximum size matching M^* that is popular within the set of maximum size matchings in G, that is, $|M^*| = |M_{\max}|$ and there is no *maximum* size matching that is more popular than M^*? We show that such a matching M^* always exists and can be computed in $O(mn_0)$ time, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$. Though the above matching M^* is popular restricted to the set of maximum size matchings, in the entire set of matchings in G, its *unpopularity factor* could be as high as $n_0 - 1$. On the other hand, a maximum size popular matching could be of size only $\frac{2}{3}|M_{\max}|$. In between these two extremes, we show there is an entire spectrum of matchings: for any integer k, where $2 \le k \le n_0$, there is a matching M_k in G of size at least $\frac{k}{k+1}|M_{\max}|$ whose unpopularity factor is at most $k-1$. Also, such a matching M_k can be computed in $O(km)$ time by a simple generalization of our maximum size popular matching algorithm.

Key words. matchings, bipartite graphs, preference lists

AMS subject classification. 68W40

DOI. 10.1137/120902562

1. Introduction. An instance of the stable marriage problem is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex ranks its neighbors in a strict order of preference. Every vertex $u \in \mathcal{A} \cup \mathcal{B}$ seeks to be matched to one of its neighbors. Preference lists can be incomplete, which means that a vertex may be adjacent to only some of the vertices on the other side. Also, preference lists are symmetric, i.e., a belongs to b's list if and only if b belongs to a's list, for any pair of vertices a and b. It is customary to refer to the vertices in $\mathcal{A}$ and $\mathcal{B}$ as men and women, respectively. We will also refer to G as a bipartite graph with 2-*sided* strict preference lists. We assume that no vertex is isolated, so $m \ge n/2$, where $|E| = m$ and $|\mathcal{A} \cup \mathcal{B}| = n$.

A matching M is a set of edges, no two of which share an endpoint. For any vertex x that is matched in M, let $M(x)$ denote x's partner in M. An edge (u, v) is a *blocking edge* to M if both u and v prefer each other to their respective assignments in M, i.e., 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 M is *stable* if M has no blocking edges. The existence of stable matchings in every instance $G = (\mathcal{A} \cup \mathcal{B}, E)$ and the Gale–Shapley algorithm [4] for computing a stable matching are classical results in graph algorithms. Though the original Gale–Shapley algorithm assumed

*Received by the editors December 14, 2012; accepted for publication (in revised form) October 21, 2013; published electronically January 21, 2014. A preliminary version of this work appeared in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, 2012, pp. 123–134. Part of this work was done during a visit to the Max-Planck-Institut für Informatik, Saarbrücken under the IMPECS program.

http://www.siam.org/journals/sicomp/43-1/90256.html

†Tata Institute of Fundamental Research, Mumbai 400005, India (kavitha@tcs.tifr.res.in).

that all preference lists are complete, it is straightforward to generalize this algorithm to incomplete lists [7].

A stable matching has usually been considered the best way of matching vertices in $G = (\mathcal{A} \cup \mathcal{B}, E)$. However, stability is a very strong condition, and it has been shown that all stable matchings in G have the same size and they all leave the same vertices unmatched [5]. It is easy to see that every stable matching S has size at least $\frac{1}{2}|M_{\max}|$, where $M_{\max}$ is a maximum matching in G—otherwise, there would be an edge $(a, b) \in M_{\max} \setminus S$ such that S leaves a and b unmatched; in other words, (a, b) would be a blocking edge to S.

There are simple examples where this bound is tight. Consider the following instance with $\mathcal{A} = \{x_1, x_2\}$ and $\mathcal{B} = \{y_0, y_1\}$, and let the preference lists be as shown in Figure 1.

x_1	y_1	y_0
x_2	y_1	

y_0	x_1	
y_1	x_1	x_2

FIG. 1. *Here x_1's top choice is y_1 and second choice is y_0, while x_2 has a single neighbor y_1. Similarly, the vertex y_0 has a single neighbor x_1, while y_1's top choice is x_1 and second choice is x_2.*

The matching $S = \{(x_1, y_1)\}$ is the only stable matching here, while there exists a perfect matching $M_{\max} = \{(x_1, y_0), (x_2, y_1)\}$. Thus $|S| = \frac{1}{2}|M_{\max}|$ in this instance. This example can be easily generalized to $4t$ vertices, for any integer $t \geq 1$, where a stable matching has size t while the instance admits a perfect matching of size $2t$.

There are many applications where it is desirable to have matchings whose size is larger than that of a stable matching—for instance, in allocating projects to students, where the total absence of blocking edges is not necessary and a more relaxed notion of stability suffices. The notion of popularity captures a natural relaxation of the notion of stability: blocking edges are permitted in a popular matching M; nevertheless, M has overall stability. That is, in popular matchings, pairwise stability gets replaced by global stability.

1.1. Popular matchings. For any two matchings M_0 and M_1, we say that vertex u prefers M_0 to M_1 if u is *better off* in M_0 than in M_1, i.e., u is either matched in M_0 and unmatched in M_1 or matched in both and prefers $M_0(u)$ to $M_1(u)$. Let $\phi(M_0, M_1)$ equal the number of vertices that prefer M_0 to M_1. We say that M_0 is more popular than M_1 if $\phi(M_0, M_1) > \phi(M_1, M_0)$.

DEFINITION 1. *A matching M is popular if $\phi(M, M') \geq \phi(M', M)$ for all matchings M'.*

Thus a matching M is popular if there is no matching that is more popular than M. Popularity captures global stability since there is no matching where more vertices are better off than in M, where M is a popular matching. Gärdenfors [6] introduced the notion of popularity in the context of stable matchings. Every stable matching is popular: when comparing a stable matching S to any matching M', note that for any edge $e \in M'$, *both* 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 endpoint 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, and hence M' cannot be more popular than S.

Since stable matchings always exist in a stable marriage instance, popular matchings also always exist in a stable marriage instance. In the example described in

Figure 1, it is easy to see that $M_{\max} = \{(x_1, y_0), (x_2, y_1)\}$ is also popular. Thus there are instances where a maximum size popular matching can be twice as large as a stable matching. It has been shown that a stable matching is a *minimum* size popular matching [8].

So in problems where we are ready to substitute stability with popularity for the sake of obtaining a matching of larger size, the desired matching is a maximum size popular matching. The only polynomial time algorithm known for computing such a matching is an $O(mn_0)$ algorithm from [8], where $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$ and $m = |E|$. We show the following result here.

THEOREM 1. *A maximum size popular matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists can be computed in $O(m)$ time, where $m = |E|$.*

Thus we have a linear time algorithm for computing a maximum size popular matching in a stable marriage instance $G = (\mathcal{A} \cup \mathcal{B}, E)$, so the complexity of computing a maximum size popular matching is the same as that of computing a stable matching. The size of a maximum size popular matching need not be better than $\frac{2}{3}|M_{\max}|$, as shown by this example. Let $\mathcal{A} = \{a_1, a_2, a_3\}$ and $\mathcal{B} = \{b_0, b_1, b_2\}$, and the preference lists are given in Figure 2.

a_1	b_1	b_0
a_2	b_2	b_1
a_3	b_2	

b_0	a_1	
b_1	a_1	a_2
b_2	a_2	a_3

FIG. 2. *An example where a maximum size popular matching has size $\frac{2}{3}|M_{\max}|$.*

There is only one popular matching in the above instance, $\{(a_1, b_1), (a_2, b_2)\}$. However, the instance admits a perfect matching $\{(a_1, b_0), (a_2, b_1), (a_3, b_2)\}$ (see Figure 3). The above example can be easily generalized to $6t$ vertices, for any integer $t \geq 1$, by making t copies of the above graph with no edges between any of the copies, so that the maximum size popular matching has size $2t$ while the instance admits a perfect matching of size $3t$.

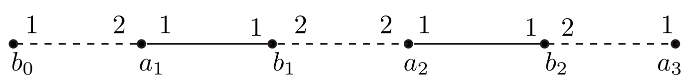

FIG. 3. *The bold edges form the maximum size popular matching and the dashed edges form a perfect matching. The preferences of the vertices are indicated on the edges: 1 is the top choice while 2 is the second choice.*

In some applications, for instance, assigning training positions to trainees, we cannot compromise on the size of the matching, so a maximum size popular matching may not always be the best matching in such applications. Here the matching has to be of maximum cardinality in G, and among such matchings, we want a "best" matching. So what we seek here is a maximum matching M^* such that for any maximum matching $M_{\max}$, we have $\phi(M^*, M_{\max}) \geq \phi(M_{\max}, M^*)$. In other words, there is no maximum matching where more vertices are better off than in M^*. It is not clear whether such a matching M^* always exists. Let $\mathcal{M}$ be the set of maximum matchings in G. We show the following result.

THEOREM 2. *In any bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ with 2-sided strict preference lists, there always exists a maximum matching M^* that is popular within the set $\mathcal{M}$ of maximum matchings, and M^* can be computed in $O(mn_0)$ time, where $m = |E|$ and $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$.*

Though M^* is popular within the set $\mathcal{M}$, note that M^* could be quite *unpopular* in the set of all matchings in G. In order to measure the unpopularity of a matching, we use the following definition from [15]. In any instance G, the function Δ measures how much one matching (say, M_1) can be more popular than another (say, M_0):

$$\Delta(M_0, M_1) = \frac{\phi(M_1, M_0)}{\phi(M_0, M_1)} \quad \text{if } \phi(M_0, M_1) \neq 0.$$

Otherwise (i.e., $\phi(M_0, M_1) = 0$), define $\Delta(M_0, M_1)$ to be ∞.

Let $\mathcal{X}$ denote the set of all matchings in G. The *unpopularity factor* of M, denoted by $u(M)$, is defined as

$$u(M) = \max_{M' \in \mathcal{X} \setminus \{M\}} \Delta(M, M').$$

A matching M is popular if and only if $u(M) \leq 1$. We show in section 3 that $u(M^*) \leq n_0 - 1$, where $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$, and the following simple example shows that this bound is tight.

This is a generalization of the instance on six vertices given in Figure 3. There are $2n_0$ vertices here, where $\mathcal{A} = \{a_1, \ldots, a_{n_0}\}$ and $\mathcal{B} = \{b_0, \ldots, b_{n_0-1}\}$ (see Figure 4). For each $1 \leq i \leq n_0 - 1$, the preference list of a_i is b_i (top choice) followed by b_{i-1} (second choice). The vertex a_{n_0} has only one neighbor, which is b_{n_0-1}. The vertex b_0 has only one neighbor, which is a_1. For each $1 \leq i \leq n_0 - 1$, the preference list of b_i is a_i (top choice) followed by a_{i+1} (second choice).

Fig. 4. *The example in Figure* 3 *extended to* $2n_0$ *vertices.*

There is only one maximum size matching here, which is the perfect matching $M^* = \cup_{i=0}^{n_0-1}\{(a_{i+1}, b_i)\}$. Consider the matching $M = \{(a_i, b_i),\ \text{where } 1 \leq i \leq n_0-1\}$. We have $\phi(M, M^*) = 2n_0 - 2$ since all the $2n_0 - 2$ vertices a_i, b_i for $i = 1, \ldots, n_0 - 1$ prefer M to M^*. The two vertices b_0 and a_{n_0} prefer M^* to M since they are unmatched in M but matched in M^*. So $\phi(M^*, M) = 2$. Thus $\Delta(M^*, M) = n_0 - 1$, so $u(M^*) \geq n_0 - 1$.

Summarizing, the solution given by Theorem 1 is a maximum size matching within the set of popular matchings, and the solution given by Theorem 2 is a matching of size $|M_{\max}|$ that is popular within the set of maximum matchings. The size of the former matching could be as low as $\frac{2}{3}|M_{\max}|$, while the unpopularity factor of the latter matching could be as high as $n_0 - 1$. It is natural to ask whether there are matchings sandwiched in size and popularity between these two extremes. We show that there is an entire spectrum of such matchings and that these can be computed efficiently.

Theorem 3. *For every integer $k \geq 2$, there exists a matching M_k in $G = (\mathcal{A} \cup \mathcal{B}, E)$ such that $|M_k| \geq \frac{k}{k+1}|M_{\max}|$ and $u(M_k) \leq k - 1$; moreover, no matching whose size is at least $|M_k|$ is more popular than M_k. Also, M_k can be computed in $O(km)$ time, where $m = |E|$.*

When $k = 2$, Theorem 3 promises a matching M_2 such that $u(M_2) \leq 1$, i.e., M_2 is popular. It will be shown in section 2 that this matching M_2 is a maximum size popular matching—thus this is the matching described in Theorem 1. When the parameter $k = n_0$, Theorem 3 promises a matching M_{n_0} of size $|M_{\max}|$ that is popular among maximum size matchings—thus this is the matching described in Theorem 2.

1.2. Background and related results. Several variants of the popular matchings problem have been studied in the model where only vertices of $\mathcal{A}$ have preferences while vertices of $\mathcal{B}$ have no preferences [1, 10, 11, 13, 14, 15, 16, 17]. This is the model of 1-sided preference lists. Here each edge $e = (a, b)$ in G has a rank associated with it (the rank that a assigns to b) and it is only vertices in $\mathcal{A}$ that cast their votes. There are simple examples in this model that admit no popular matching. Abraham et al. [1] gave efficient algorithms for determining whether a given instance admits a popular matching or not and, if so, for computing one of maximum size. McCutchen [15] introduced two measures of unpopularity, unpopularity factor and unpopularity margin, and he showed that the problem of computing a matching in the domain of 1-sided preference lists that minimized either of these measures is NP-hard.

Gärdenfors [6], who introduced the notion of popular matchings, considered this problem in the domain of 2-sided preference lists, i.e., in an instance of the stable marriage problem. When ties are allowed in preference lists, it was shown by Biró, Irving, and Manlove [2] that the problem of computing an arbitrary popular matching in a stable marriage instance is NP-hard. Biró, Manlove, and Mittal [3] showed that the problem of computing a maximum size matching with the minimum number of blocking edges in a stable marriage instance is NP-hard to approximate to within $n_0^{1-\epsilon}$, for any $\epsilon > 0$, where $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$.

As mentioned earlier, the first polynomial time algorithm for computing a maximum size popular matching in a stable marriage instance with strict preference lists was given in [8]. The running time of this algorithm is $O(mn_0)$. Here a set $L \subset \mathcal{A} \cup \mathcal{B}$ is computed in an iterative manner such that when the Gale–Shapley stable matching algorithm is run with vertices of L proposing to those in $R = (\mathcal{A} \cup \mathcal{B}) \setminus L$, every vertex in R gets matched and no neighbor in L is preferred to its partner by any $u \in L$. It was shown that such a matching has to be a maximum size popular matching. In order to construct an L that satisfies the above properties, this algorithm may take $\Theta(n_0)$ iterations, where $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|)$. Thus there are instances where this algorithm takes $\Theta(mn_0)$ time.

2. A linear time algorithm for a maximum size popular matching. Our input here is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where each vertex ranks its neighbors in a strict order of preference. We assume without loss of generality that $|\mathcal{A}| \leq |\mathcal{B}|$, so $n_0 = \min(|\mathcal{A}|, |\mathcal{B}|) = |\mathcal{A}|$.

Our algorithm partitions the vertex set $\mathcal{A} \cup \mathcal{B}$ into two layers: *bottom* and *top*. Initially the top layer is empty. At any point in time, the vertices of $\mathcal{A}$ (call them *men*) in the top layer are there because they could not find partners by being in the bottom layer. In this algorithm, the top layer men get preferential treatment—in each iteration, the top layer men first make their proposals and the vertices of $\mathcal{B}$ (call them *women*) that they seek are confined to the top layer. Only the women not sought after by them are available to the bottom layer men.

So in each iteration, the Gale–Shapley stable matching algorithm is first run with the top layer men proposing and all the women who received proposals disposing; let S_1 denote this matching. All the women who are matched in S_1 move to the top layer. The men in the bottom layer then run the stable matching algorithm with the women left in the bottom layer to yield a matching S_0. See Figure 5. If all the bottom layer men get matched in S_0, then $S_1 \cup S_0$ is returned. Else the unmatched men in the bottom layer are promoted to the top layer and the next iteration begins.

Suppose we run the above algorithm on the example given in Figure 1 where x_1 and y_1 are each other's top choices while x_2's only neighbor is y_1 and y_0's only

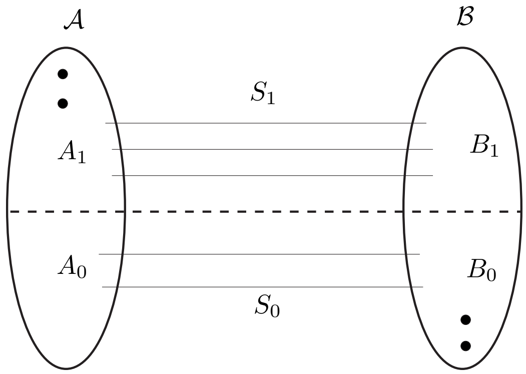

FIG. 5. *Let A_1 (similarly, A_0) denote the set of men in the top (resp., bottom) layer, and let B_1 (similarly, B_0) denote the set of women in the top (resp., bottom) layer. The returned matching is $S_1 \cup S_0$, where S_1 (similarly, S_0) is stable in the graph induced on $A_1 \cup \mathcal{B}$ (resp., $A_0 \cup B_0$).*

neighbor is x_1. Initially all the vertices are in the bottom layer. Though top layer men propose first in every iteration, however, since the top layer is empty in the first iteration, we have $S_1 = \emptyset$ in the first iteration and we compute a stable matching in the bottom layer with all the men proposing and women disposing, so $S_0 = \{(x_1, y_1)\}$. Then the vertex x_2, which is unmatched in S_0, gets promoted to the top layer. In the second iteration, the vertex x_2 gets to propose first and this yields $S_1 = \{(x_2, y_1)\}$. The bottom layer vertices are x_1 and y_0. Since there is an edge between x_1 and y_0, we get $S_0 = \{(x_1, y_0)\}$. The termination condition is now satisfied. Thus the matching $\{(x_1, y_0), (x_2, y_1)\}$ is returned (see Figure 6).

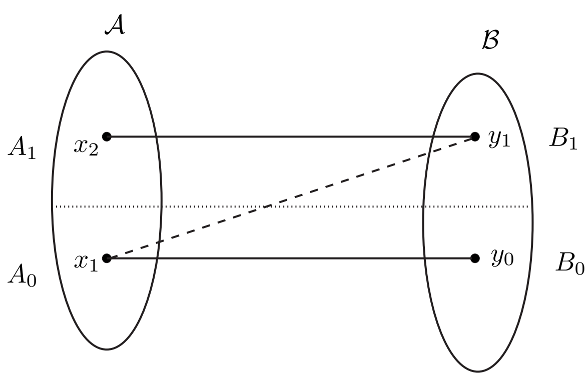

FIG. 6. *The matching $\{(x_1, y_0), (x_2, y_1)\}$ is computed by the 2-layer algorithm. It has a blocking edge (x_1, y_1).*

Note that the idea of "promoting" an unmatched man is reminiscent of a similar step in Király's approximation algorithm [12] for a maximum size weakly stable matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ where vertices have ties in their preference lists. However, since the goal in Király's algorithm is to compute a matching that admits *no* blocking edges, the promotion step there is only to break ties in the preference lists, whereas in our algorithm, the promotion of an unmatched man from the bottom layer to the top layer may create blocking edges. Nevertheless, the resulting matching will be popular.

Before we show the correctness of this algorithm, we will first show a simple linear time implementation of this algorithm. This involves calling a modified Gale–Shapley stable matching algorithm just once in an *augmented* graph $\tilde{G}_2 = (\tilde{\mathcal{A}}_2 \cup \mathcal{B}, \tilde{E}_2)$. The set of women in $\tilde{G}_2$ is the same as the set $\mathcal{B}$ of women in G. The fact that the vertex set of G gets partitioned into two layers is implemented by having *two* copies of every man $a_i \in \mathcal{A}$ in the graph $\tilde{G}_2$. So $\tilde{\mathcal{A}}_2 = \{a_1^0, \dots, a_{n_0}^0, a_1^1, \dots, a_{n_0}^1\}$, where $\{a_1, \dots, a_{n_0}\}$ is the set $\mathcal{A}$ of men in the given graph G. The preference list of each $a_i^\ell \in \tilde{\mathcal{A}}_2$, for $\ell = 0, 1$, is the same as that of $a_i \in \mathcal{A}$ in G.

The superscript ℓ in a_i^ℓ refers to the layer number: $\ell = 0$ denotes the bottom layer while $\ell = 1$ denotes the top layer. At the beginning, for all i, only a_i^0 participates in the algorithm since the top layer is empty at the start of the algorithm. For any i, if a_i^0 is rejected by all his neighbors, then a_i^0 exits and a_i^1 starts participating in the algorithm. The replacement of a_i^0 by a_i^1 captures a_i getting promoted from *bottom* to *top*. The fact that in every iteration the top layer men propose first to all women and the bottom layer men can propose only to those women who do not receive proposals from top layer men is captured by the women's preference lists.

- The preference list of each woman b in $\tilde{G}_2$ is as follows: if b's preference list in G is $\langle a_{i_1}, \dots, a_{i_t} \rangle$, then b's preference list in $\tilde{G}_2$ is $\langle a_{i_1}^1, \dots, a_{i_t}^1, a_{i_1}^0, \dots, a_{i_t}^0 \rangle$. Thus $\deg(b)$ in $\tilde{G}_2$ is $2\deg_G(b)$.
  - The top layer copies of all the neighbors of b in G (in the same order of preference as in G) are the most preferred $\deg_G(b)$ neighbors of b in $\tilde{G}_2$.
  - Then come the bottom layer copies of all the neighbors of b in the same order of preference.

So if a woman b receives a proposal from a top layer neighbor, she will henceforth reject proposals from all bottom layer neighbors. In fact, we can say that in the Gale–Shapley stable matching algorithm, when a woman receives an offer, she immediately deletes edges between her and worse ranked neighbors since such offers will henceforth never be accepted by her. So as soon as a woman receives a proposal from a top layer neighbor, she deletes all edges incident to bottom layer neighbors, and thus the bottom layer men can propose only to those women who have not yet received proposals from top layer men.

Our algorithm for constructing the desired matching in $\tilde{G}_2 = (\tilde{\mathcal{A}}_2 \cup \mathcal{B}, \tilde{E}_2)$ is given as Algorithm 1. This algorithm is essentially the same as running the Gale–Shapley algorithm in $\tilde{G}_2$, except for some modifications. In the Gale–Shapley algorithm, all the men in $\tilde{A}_2$ should propose. However, at the very beginning, we want only the bottom layer men to propose since the top layer is empty. So our initialization step initializes the queue Q of active men to the bottom layer men $\{a_1^0, \dots, a_{n_0}^0\}$.

In the Gale–Shapley algorithm, every man who has not yet found a partner will propose in decreasing order of preference until he is accepted by some neighbor or he gets rejected by all his neighbors. Any offer that a woman receives is always from a better neighbor than her current partner since she deletes edges to worse ranked neighbors upon receiving a proposal. So when a woman receives a proposal, if she is already matched, she rejects her current partner and he is inserted into Q, since he has to find a new partner now. If a man gets rejected by all his neighbors, then he will be unmatched in the final matching output by the Gale–Shapley algorithm.

In our algorithm the modification is that once a bottom layer man a_i^0 has been rejected by all his neighbors, then a_i^0 exits and a_i^1 enters the picture. (This is the new step in our algorithm when compared to the Gale–Shapley algorithm.) Hence a_i^1 is inserted into Q. When it is a_i^1's turn, he starts proposing from the top of his preference list. If a_i^1 also gets rejected by all his neighbors, then it means that a_i will remain unmatched in our final matching. Algorithm 1 returns $\tilde{S}$ in $\tilde{G}_2$, and this translates in a straightforward manner to a matching M_2 in G: $(a,b) \in M_2$ if and only if $\tilde{S}(b)$ is a^0 or a^1.

ALGORITHM 1. *Input:* $\tilde{G}_2 = (\tilde{\mathcal{A}}_2 \cup \mathcal{B}, \tilde{E}_2)$; *Output:* $\tilde{S}$

---

1. Initialize the queue Q to $\{a_1^0, \ldots, a_{n_0}^0\}$ and $\tilde{S}$ to the empty matching.
2. **while** Q is not empty **do**
3.   delete the first element a^ℓ from Q.
4.   **if** a^ℓ's list of neighbors in the current graph is nonempty **then**
5.   – let b be the most preferred neighbor of a^ℓ in this list.
6.   – **if** $\tilde{S}(b)$ exists **then** add this man $\tilde{S}(b)$ to Q.
        {*Since the current graph has no edges between b and neighbors ranked worse than* $\tilde{S}(b)$, *the existence of* (a^ℓ, b) *in this graph implies b prefers* a^ℓ *to* $\tilde{S}(b)$*.*}
7.   – set $\tilde{S}(b) = a^\ell$. {*So* a^ℓ *becomes b's current partner.*}
8.   – delete from the current graph edges between b and neighbors worse than a^ℓ.
9.   **else if** $\ell = 0$ **then**
10.   – add a^1 to Q.
        {*At this point* a^0 *has been rejected by all his neighbors; hence* a^0 *exits and* a^1 *enters.*}
11.   **end if**
12. **end while**
13. Return $\tilde{S}$.

Building the graph $\tilde{G}_2$ takes $O(n+m)$ time. The running time of Algorithm 1 is the same as the running time of the Gale–Shapley algorithm on $\tilde{G}_2$, which is linear in the size of $\tilde{G}_2$. The number of edges in $\tilde{G}_2$ is $2\sum_{i=1}^{n_0} \deg_G(a_i)$. Thus the running time of Algorithm 1 is $O(n+m)$. So the time taken to compute M_2 is $O(n+m)$, which is $O(m)$.

We now show that the matching M_2 is a maximum size popular matching in G. The following definition partitions $\mathcal{A}$ into the set A_0 of bottom layer men and the set A_1 of top layer men and, similarly, $\mathcal{B}$ into the set B_0 of bottom layer women and the set B_1 of top layer women.

DEFINITION 2. *Let* A_0 *consist of those men* $a_i \in \mathcal{A}$ *such that there exists some* $b \in \mathcal{B}$ *that satisfies* $\tilde{S}(b) = a_i^0$*, and let* $A_1 = \mathcal{A} \setminus A_0$*. Let* $B_1 \subseteq \mathcal{B}$ *be the set of women matched in* M_2 *to the men in* A_1*, and let* $B_0 = \mathcal{B} \setminus B_1$*.*

Thus we have $M_2 \subseteq (A_0 \times B_0) \cup (A_1 \times B_1)$. Claim 1 follows from the definitions of the sets A_0 and B_1.

CLAIM 1. *All the men unmatched in* M_2 *belong to* A_1 *and all the women unmatched in* M_2 *belong to* B_0*.*

The following definition will be useful in showing the properties satisfied by M_2.

DEFINITION 3. *For any* $u \in \mathcal{A} \cup \mathcal{B}$ *and neighbors* x *and* y *of* u*, define* u*'s vote between* x *and* y*, denoted by* $\mathsf{vote}_u(x,y)$*, as follows: it is* 1 *if* u *prefers* x *to* y*, and it is* -1 *if* u *prefers* y *to* x*; else, it is* 0 *(i.e.,* $x = y$*).*

Label each $e = (u,v)$ in $E \setminus M_2$ by (α_e, β_e), where $\alpha_e = \mathsf{vote}_u(v, M_2(u))$ and $\beta_e = \mathsf{vote}_v(u, M_2(v))$; in case x is unmatched in M_2, then $\mathsf{vote}_x(y, M_2(x)) = 1$ for any neighbor y of x since every vertex prefers being matched with any of its neighbors to being unmatched. Note that an edge is a blocking edge with respect to M_2 if and only if it is labeled $(1,1)$. Lemmas 1 and 2 show crucial properties of our vertex partition.

LEMMA 1. *Every edge* $(a,b) \in A_1 \times B_0$ *is labeled* $(-1,-1)$.

Proof. Let (a,b) be an edge in $A_1 \times B_0$. We first claim that a must be matched in M_2. Otherwise, a^1 would have proposed to b. However, $b \in B_0$, which means that b never received a proposal from a top layer neighbor during the entire course of the algorithm; otherwise, b would have accepted such a proposal. So a^1 has to be matched in $\tilde{S}$ to a woman that a ranks better than b. So $\mathsf{vote}_a(b, M_2(a)) = -1$.

The man a^0 was rejected by all his neighbors in Algorithm 1; that is why he got promoted to the top layer. So at some point a^0 must have been rejected by b. When b rejected a^0, b was matched to a man ranked better than a^0 in b's preference list in $\tilde{G}_2$. Also, b never received a proposal from a top layer neighbor (since $b \in B_0$). Thus the final partner of b in $\tilde{S}$ is a bottom layer man z^0 whom b prefers to a^0; in other words, b ranks her partner $M_2(b) = z$ better than a. So we have $\mathsf{vote}_b(a, M_2(b)) = -1$. This proves the lemma. ∎

LEMMA 2. *Every edge labeled* $(1,1)$ *has to be in* $A_0 \times B_1$.

Proof. During the entire course of Algorithm 1, no woman in B_0 ever receives a proposal from a top layer neighbor; otherwise, she would be matched to some a_i^1 in $\tilde{S}$. Thus the matching M_2 restricted to the vertex set $A_0 \cup B_0$ is stable since these women receive proposals only from the bottom layer men and they dispose according to their preference lists in G. Hence there are no blocking edges in $A_0 \times B_0$.

The men in A_1 propose according to their preference lists in G, and the women who receive their proposals prefer one top layer man to another according to their preference lists in G. Thus M_2 has no blocking edges in $A_1 \times B_1$. We know from Lemma 1 that every edge in $(A_1 \times B_0)$ is labeled $(-1,-1)$. Since M_2 has no blocking edges in $A_1 \times (B_0 \cup B_1)$ or in $A_0 \times B_0$, it follows that every edge labeled $(1,1)$ has to be in $A_0 \times B_1$. ∎

Let G_{M_2} denote the subgraph of G obtained by deleting from G all edges that are labeled $(-1,-1)$. We now show the following lemma in the graph G_{M_2}. A path (similarly, cycle) where alternate edges belong to M_2 is called an *alternating* path (resp., cycle) with respect to M_2. If the endpoints of the alternating path are unmatched in M_2, then such a path is also called an *augmenting* path with respect to M_2.

LEMMA 3. *Let* $\rho = \langle y_0, x_1, y_1, x_2, y_2, \ldots \rangle$ *be an alternating path in* G_{M_2}*, where* $(x_i, y_i) \in M_2$ *for* $i \geq 1$.

(i) *If* $y_0 \in A_1 \cup B_0$*, then there is no edge labeled* $(1,1)$ *in* ρ.

(ii) *If* $y_0 \in A_0 \cup B_1$*, then there can be at most one edge labeled* $(1,1)$ *in* ρ.

Proof. We first show (i). Suppose $y_0 \in A_1$. There are no edges in G_{M_2} between A_1 and B_0 (by Lemma 1). So y_0's neighbor in ρ, i.e., the vertex x_1, has to be in B_1. Since the matched partners of all vertices in B_1 have to be in A_1, $M_2(x_1) = y_1 \in A_1$. Thus it follows that $x_i \in B_1$ and $y_i \in A_1$ for all $i \geq 1$. So every edge of the path ρ is in $A_1 \times B_1$. As all the edges labeled $(1,1)$ are in $A_0 \times B_1$ (by Lemma 2), there is no edge labeled $(1,1)$ in ρ.

Suppose $y_0 \in B_0$. Since there are no edges between B_0 and A_1 in G_{M_2}, and because the matched partners of all vertices in A_0 are in B_0, it follows that $x_i \in A_0$ and $y_i \in B_0$ for all $i \geq 1$. So every edge of the path ρ is in $A_0 \times B_0$. Hence there is no edge labeled $(1,1)$ in ρ.

We now show (ii). Suppose $y_0 \in A_0$. There are edges (some of them possibly labeled $(1,1)$) between A_0 and B_1. However, once an edge of $A_0 \times B_1$ is traversed in ρ, the path ρ gets stuck in $A_1 \cup B_1$. This is so by the same argument as in the earlier case. Once ρ reaches a vertex $x_i \in B_1$, its matched partner $y_i \in A_1$ and thereafter all the vertices have to be in $A_1 \cup B_1$ as there are no edges between A_1 and B_0 in G_{M_2} and because the matched partners of all vertices in B_1 are in A_1.

Supposing $y_0 \in B_1$, a similar argument holds: though there are edges (possibly labeled $(1,1)$) between B_1 and A_0, once an edge of $B_1 \times A_0$ is traversed in ρ, the path ρ gets stuck in $A_0 \cup B_0$ because every vertex in A_0 is matched to a vertex in B_0 and there are no edges between B_0 and A_1 in G_{M_2}. So once ρ reaches a vertex $x_i \in A_0$, thereafter all the vertices have to be in $A_0 \cup B_0$. Thus we have shown that in both cases of (ii), there can be at most one edge labeled $(1,1)$ in ρ. ◻

We will refer to an alternating path $\langle y_0, x_1, y_1, \ldots \rangle$ in G_{M_2} where $y_0 \in A_1 \cup B_0$ as a *type* (i) alternating path and one where $y_0 \in A_0 \cup B_1$ as a *type* (ii) alternating path.

Let M' be any matching in G. In order to compare M_2 and M' with respect to popularity, we can assume that M' belongs to the subgraph G_{M_2}. This is because if (u,v) is an edge of M' that is labeled $(-1,-1)$, then we can assume as well that M' leaves u and v unmatched; i.e., we can delete the edge (u,v) from M' since this makes no difference to $\mathsf{vote}_u(M'(u), M_2(u))$ or $\mathsf{vote}_v(M'(v), M_2(v))$: both these values were -1 when (u,v) was in M' and they both remain -1 after assuming that u and v are unmatched in M'.

So for the purpose of evaluating $\phi(M_2, M')$ and $\phi(M', M_2)$, we can assume that M' is in G_{M_2}. Hence $M_2 \oplus M'$ is in G_{M_2}. The set $M_2 \oplus M'$ is a collection of alternating paths and alternating cycles with respect to M_2. Theorem 4 will imply that $\phi(M', M_2) \le \phi(M_2, M')$.

THEOREM 4. *For any matching M' in G_{M_2}, the following three statements hold:*

1. *If ρ is an alternating cycle in $M_2 \oplus M'$, then $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$.*
2. *If ρ is an alternating path in $M_2 \oplus M'$ such that at least one endpoint of ρ is unmatched in M_2, then $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$.*
3. *If ρ is an alternating path in $M_2 \oplus M'$ such that both endpoints of ρ are matched in M_2, then $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$.*

Proof. Let ρ be any alternating path or cycle in $M_2 \oplus M'$. So ρ is in G_{M_2}. We will now show that $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$. The value $\phi(M_2 \oplus \rho, M_2) - \phi(M_2, M_2 \oplus \rho)$ is $\sum_{u \in \rho} \mathsf{vote}_u(M'(u), M_2(u))$, where the sum is over all the vertices u in ρ. This can be written as

$$\phi(M_2 \oplus \rho, M_2) - \phi(M_2, M_2 \oplus \rho) \ = \sum_{\substack{u \in \rho \\ \text{unmatched in } M'}} -1 \ + \sum_{e \in \rho \cap M'} (\alpha_e + \beta_e), \tag{1}$$

where $\alpha_e = \mathsf{vote}_u(v, M_2(u))$ and $\beta_e = \mathsf{vote}_v(u, M_2(v))$ for edge $e = (u,v)$ in $\rho \cap M'$. We will bound the right-hand side of (1) now.

Let ρ be an alternating cycle in $M_2 \oplus M'$. Since every edge of M_2 is either in $A_0 \times B_0$ or in $A_1 \times B_1$, there has to exist a vertex $x \in A_1 \cup B_0$ in ρ. Thus $\rho \setminus \{(x, M_2(x))\}$ is a type (i) alternating path. Lemma 3 tells us that there can be no $(1,1)$ edge in such an alternating path. Hence $\alpha_e + \beta_e \le 0$ for each $e \in \rho \cap M'$. Thus the right-hand side of (1) is at most 0 here, and hence $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$ in this case.

Let ρ be an alternating path in $M_2 \oplus M'$. In part 2, there is an endpoint of ρ that is unmatched in M_2 and this vertex has to be in $A_1 \cup B_0$ (by Claim 1). So ρ is a type (i) alternating path. There can be no $(1,1)$ edge in ρ by Lemma 3. Hence $\alpha_e + \beta_e \le 0$ for each $e \in \rho \cap M'$. Thus the right-hand side of (1) is again at most 0, and hence $\phi(M_2 \oplus \rho, M_2) \le \phi(M_2, M_2 \oplus \rho)$ in this case also.

In part 3, ρ is an alternating path with respect to M_2 in G_{M_2} such that both endpoints of ρ are matched in M_2. So neither endpoint is matched in M' and both these vertices prefer M_2 to M'. So these two vertices contribute -1 each to the first term on the right-hand side of (1). We know by Lemma 3 that there can be at most one edge labeled $(1,1)$ in ρ. Hence $\sum_{e\in\rho\cap M'}(\alpha_e+\beta_e)\le 2$. Thus the sum on the right-hand side of (1) is at most $-2+2=0$. So we have $\phi(M_2\oplus\rho,M_2)\le\phi(M_2,M_2\oplus\rho)$ here also. ∎

For any matching M' in G, we have (let $M'\cap G_{M_2}$ denote $M'\cap$ the edge set of G_{M_2})

$$\begin{aligned}\phi(M',M_2) &= \sum_{\rho\in M_2\oplus(M'\cap G_{M_2})}\phi(M_2\oplus\rho,M_2)\\ &\le \sum_{\rho\in M_2\oplus(M'\cap G_{M_2})}\phi(M_2,M_2\oplus\rho) \qquad \{\text{by Theorem 4}\}\\ &= \phi(M_2,M').\end{aligned}$$

Thus M_2 is popular. We now show (via Lemmas 4 and 5) that M_2 is a maximum size popular matching. Recall that an augmenting path with respect to M_2 is an alternating path p where both endpoints of p are unmatched in M_2.

LEMMA 4. *There is no augmenting path with respect to M_2 in G_{M_2}.*

Proof. Let $p=\langle b_0,a_1,b_1,\dots,b_t,a_{t+1}\rangle$ be an augmenting path with respect to M_2 in G_{M_2}, where $b_0\in B_0$ and $a_{t+1}\in A_1$ (by Claim 1). Since M_2 uses only edges of $(A_0\times B_0)\cup(A_1\times B_1)$, p has to contain an edge between a vertex $b_{j-1}\in B_0$ and a vertex $a_j\in A_1$. However, we know there is no such edge in G_{M_2} (by Lemma 1). Thus there exists no augmenting path with respect to M_2 in G_{M_2}. ∎

LEMMA 5. *If M' is a matching in G such that $|M'|>|M_2|$, then $\phi(M_2,M')>\phi(M',M_2)$.*

Proof. Let M' be a matching in G such that $|M'|>|M_2|$. Then there is an augmenting path $p\in M_2\oplus M'$ with respect to M_2 in G. In order to evaluate $\phi(M_2,M')$ and $\phi(M',M_2)$, recall that we can restrict M' to G_{M_2}. Since there is no augmenting path with respect to M_2 in G_{M_2} (by Lemma 4), the augmenting path p in G breaks into subpaths $p_1,p_2,\dots,p_s$ in G_{M_2}, where p_1 and p_s have one endpoint each unmatched in M_2. Such an endpoint has to be in $A_1\cup B_0$ (by Claim 1), and thus there is no $(1,1)$ edge in either p_1 or p_s by Lemma 3.

So all edges of M' in p_1 (say, there are t of them) are only $(1,-1)$ and $(-1,1)$ edges. Also, p_1 has another endpoint u that is unmatched in M' (restricted to G_{M_2}) but is matched in M_2, so u prefers M_2 to M'. So p_1 has $2t+1$ vertices, where $t+1$ of these prefer M_2 to M' and the remaining t prefer M' to M_2, i.e., $\phi(M_2,M_2\oplus p_1)=\phi(M_2\oplus p_1,M_2)+1$.

Let ρ be any other alternating path or cycle in $M_2\oplus M'$, including one of $p_2,\dots,p_s$ (the other subpaths that p gets split into in G_{M_2}). We have $\phi(M_2,M_2\oplus\rho)\ge\phi(M_2\oplus\rho,M_2)$ by Theorem 4. Hence it follows that $\phi(M_2,M')>\phi(M',M_2)$. ∎

Thus no matching of size larger than $|M_2|$ can be popular since M_2 is more popular than such a matching. So M_2 is a maximum size popular matching in G. This completes the proof of Theorem 1 stated in section 1.

3. The generalized algorithm. We know there are instances (Figure 3) where a maximum size popular matching has size $\frac{2}{3}|M_{\max}|$, where $M_{\max}$ is a maximum matching in G. In order to obtain matchings of larger size, we now generalize our algorithm on 2 layers to an algorithm on k layers, for any $k\ge 2$.

As in the case for $k=2$, initially all the men are in layer 0. In each iteration, run the proposal/disposal algorithm between the men in the topmost layer (layer $k-1$) and all women—call this matching S_{k-1}. Then run the proposal/disposal algorithm between the men in layer $k-2$ and the women left unmatched in S_{k-1}—call this matching S_{k-2}. In decreasing order, for every $i \geq 0$, run the proposal/disposal algorithm between the men in layer i and the women left unmatched in $\cup_{j>i} S_j$. If *all* the men, except possibly those in layer $k-1$, are matched in $S = \cup_{i=0}^{k-1} S_i$, then S is returned; otherwise, the unmatched men of layer i are promoted to layer $i+1$, for each $0 \leq i \leq k-2$, and the next iteration begins.

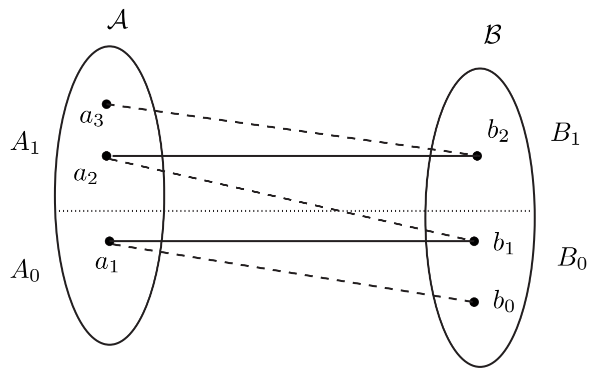

FIG. 7. *The bold edges form the matching* $\{(a_1, b_1), (a_2, b_2)\}$ *computed by the* 2*-layer algorithm.*

Figure 7 has the matching and partition computed by the maximum size popular matching on the instance in Figure 3. Here $k = 2$ and we have $A_0 = \{a_1\}$, $A_1 = \{a_2, a_3\}$, $B_0 = \{b_0, b_1\}$, and $B_1 = \{b_2\}$. When $k = 3$, the vertex a_3 (unmatched in level 1 by the 2-layer algorithm) gets promoted one layer higher, i.e., to level 2. When b_2 receives a proposal from a_3 , she accepts this proposal and $S_2 = \{(a_3, b_2)\}$. Now a_2, who is in level 1, proposes to b_1, who accepts him and $S_1 = \{(a_2, b_1)\}$. Then a_1 (in level 0) proposes to b_0, who accepts him and $S_0 = \{(a_1, b_0)\}$. The termination condition is satisfied, and hence we get the matching $S = \{(a_1, b_0), (a_2, b_1), (a_3, b_2)\}$. Thus using 3 layers gives us the partition shown in Figure 8 and the resulting perfect matching.

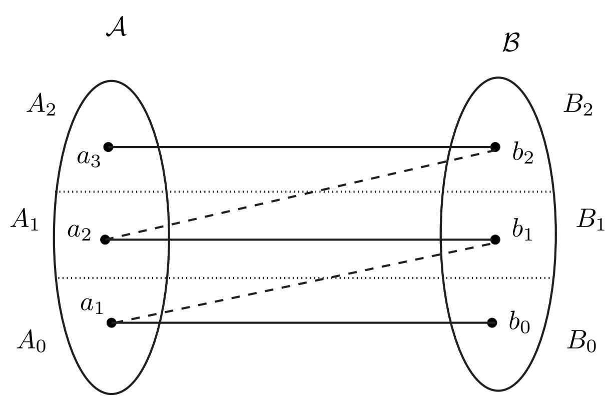

FIG. 8. *The bold edges form the matching* $\{(a_1, b_0), (a_2, b_1), (a_3, b_2)\}$ *computed by the* 3*-layer algorithm.*

Just as Algorithm 1 was an efficient implementation of the idea of partitioning the vertex set into two layers, *bottom* and *top*, we now show an efficient implementation of the generalized algorithm that partitions the vertex set into k layers, for any integer

$k \geq 2$. Let the k layers be layer 0, layer $1, \ldots,$ layer $k-1$, where layer 0 is the bottommost layer and layer $k-1$ is the topmost layer. We want the men in layer $k-1$ to get the most preferential treatment, then the men in layer $k-2$, and so on.

To implement this idea efficiently, we will work with the augmented graph $\tilde{G}_k = (\tilde{\mathcal{A}}_k \cup \mathcal{B}, \tilde{E}_k)$, where the set $\tilde{\mathcal{A}}_k$ of men is $\cup_{\ell=0}^{k-1}\{a_1^\ell, \ldots, a_{n_0}^\ell\}$ (recall that $\{a_1, \ldots, a_{n_0}\}$ is the set $\mathcal{A}$ of men in G). The set of women in $\tilde{G}_k$ is the same as the set $\mathcal{B}$ of women in G.

The preference list of each $a_i^\ell \in \tilde{\mathcal{A}}_k$, for $\ell = 0, \ldots, k-1$, is the same as that of $a_i \in \mathcal{A}$ in G. The preference list of each woman b in $\tilde{G}_k$ is as follows: if b's preference list in G is $\langle a_{i_1}, \ldots, a_{i_t}\rangle$, then $\deg(b)$ in $\tilde{G}_k$ is $k \cdot \deg_G(b)$ and b's neighbors are $\cup_{\ell=0}^{k-1}\{a_{i_1}^\ell, \ldots, a_{i_t}^\ell\}$.

- In b's preference list in $\tilde{G}_k$, we have $a_i^{\ell_1}$ preferred to $a_j^{\ell_2}$ if and only if either $\ell_1 > \ell_2$, or $\ell_1 = \ell_2$ and b ranks a_i better than a_j in her preference list in G.
- Thus for any $b \in \mathcal{B}$, layer $k-1$ neighbors are the most preferred, then come the layer $k-2$ neighbors, and so on, and at the bottom come the layer 0 neighbors in b's preference list in $\tilde{G}_k$.

We now present Algorithm 2, whose code is the same as that of Algorithm 1, except for lines 9–10, where "*if* $\ell = 0$" becomes "*if* $\ell < k-1$" here (since there are k layers now). For any a_i^ℓ where $\ell < k-1$, if a_i^ℓ gets rejected by all his neighbors, then a_i^ℓ has to get promoted to the next higher layer: this is achieved in our algorithm by the exit of a_i^ℓ and the arrival of $a_i^{\ell+1}$. The vertex $a_i^{\ell+1}$ is inserted into Q and starts proposing from the top of his preference list when he gets deleted from Q.

ALGORITHM 2. *Input:* $\tilde{G}_k = (\tilde{\mathcal{A}}_k \cup \mathcal{B}, \tilde{E}_k)$*; Output:* $\tilde{S}$

1. Initialize the queue Q to $\{a_1^0, \ldots, a_{n_0}^0\}$ and $\tilde{S}$ to the empty matching.
2. **while** Q is not empty **do**
3.   delete the first element a^ℓ from Q.
4.   **if** a^ℓ's list of neighbors in the current graph is nonempty **then**
5.   – let b be the most preferred neighbor of a^ℓ in this list.
6.   – **if** $\tilde{S}(b)$ exists **then** add this man to Q. {*This is because b prefers a^ℓ to $\tilde{S}(b)$.*}
7.   – set $\tilde{S}(b) = a^\ell$. {*So a^ℓ becomes b's current partner.*}
8.   – delete from the current graph edges between b and neighbors worse than a^ℓ.
9.   **else if** $\ell < k-1$ **then**
10.   – add $a^{\ell+1}$ to Q. {*At this point, a^ℓ exits and $a^{\ell+1}$ enters.*}
11.   **end if**
12. **end while**
13. Return $\tilde{S}$.

Algorithm 2 returns a matching $\tilde{S}$ in the graph $\tilde{G}_k$, and this translates in a straightforward manner to a matching M_k in G: $(a, b) \in M_k$ if and only if $\tilde{S}(b) = a^\ell$ for some $\ell \in \{0, \ldots, k-1\}$.

It is straightforward to see that the time taken to construct M_k is $O(|\tilde{G}_k|)$, which is $O(km)$. We will first bound the size of M_k from below and then bound its unpopularity factor from above. Definition 4 partitions $\mathcal{A}$ and $\mathcal{B}$ into layers.

DEFINITION 4. *For* $0 \leq \ell \leq k-2$*, let* $A_\ell \subseteq \mathcal{A}$ *consist of those men* a_i *such that* $\tilde{S}(b) = a_i^\ell$ *for some* $b \in \mathcal{B}$*, and let* $A_{k-1} = \mathcal{A} \setminus (A_0 \cup \cdots \cup A_{k-2})$*. For* $1 \leq \ell \leq k-1$*,* *let $B_\ell \subseteq \mathcal{B}$ be the set of women matched in M_k to the men in A_ℓ, and let $B_0 = \mathcal{B} \setminus (B_1 \cup \cdots \cup B_{k-1})$.*

Thus we have $M_k \subseteq \cup_{\ell=0}^{k-1}(A_\ell \times B_\ell)$. Claim 2 is straightforward from Definition 4.

CLAIM 2. *All the men unmatched in M_k are in A_{k-1} and all the women unmatched in M_k are in B_0.*

Lemma 6 shows an important property of the partitioning of $\mathcal{A} \cup \mathcal{B}$ into the layers as given by Definition 4.

LEMMA 6. *For every $2 \le \ell \le k-1$, there is no edge between any man in A_ℓ and any woman in $\cup_{t=0}^{\ell-2} B_t$.*

Proof. Consider any $a \in A_\ell$ for $\ell \ge 2$. The fact that $a \in A_\ell$ implies that $a^{\ell-1}$ was rejected by all his neighbors in $\mathcal{B}$. Consider any $b \in \cup_{j \le \ell-2} B_j$. If there had been an edge (a, b) in G, then $a^{\ell-1}$ would have proposed to b. However, we know that b could not have received any proposal from a man z^t with $t \ge \ell-1$; otherwise, b would have accepted such a proposal since a neighbor in layer $\ell-1$ or higher is ranked better than any neighbor in layer $\ell-2$ or lower, and so b would not be in $\cup_{j \le \ell-2} B_j$. Thus if (a, b) had been an edge in G, then $a^{\ell-1}$ would have proposed to b and b would have accepted $a^{\ell-1}$, contradicting that $a^{\ell-1}$ was rejected by all his neighbors. Hence there is no edge (a, b) in G, where $a \in A_\ell$ and $b \in \cup_{t=0}^{\ell-2} B_t$. ∎

Label every edge $e = (u, v) \in E \setminus M_k$ by (α_e, β_e), where $\alpha_e = \mathsf{vote}_u(v, M_k(u))$ and $\beta_e = \mathsf{vote}_v(u, M_k(v))$. If u is unmatched in M_k, then $\mathsf{vote}_u(v, M_k(u)) = 1$ for any neighbor v.

LEMMA 7. *For each $1 \le \ell \le k-1$, every edge $(a, b) \in A_\ell \times B_{\ell-1}$ is labeled $(-1, -1)$.*

The proof of Lemma 7 is analogous to that of Lemma 1. We now show an important property of M_k in Lemma 8, and this property will allow us to bound $|M_k|$ from below.

LEMMA 8. *Any augmenting path with respect to M_k in G has length at least $2k+1$.*

Proof. Let $p = \langle b_0, a_1, b_1 \ldots, b_t, a_{t+1} \rangle$ be an augmenting path with respect to M_k in G. We know from Claim 2 that $b_0 \in B_0$ and $a_{t+1} \in A_{k-1}$, and we also know that M_k uses only edges of $\cup_{\ell=0}^{k-1}(A_\ell \times B_\ell)$. In the first place, there is no edge in G between an *unmatched* $b_0 \in B_0$ and any $a_1 \in A_1$, since such an edge would not be a $(-1, -1)$ edge (because b_0 prefers being matched to a_1 to being unmatched in M_k), contradicting Lemma 7. Also, there is no edge between B_i and $\cup_{j \ge i+2} A_j$ for any $i \ge 0$ (by Lemma 6).

At the other end, there is no edge between an *unmatched* vertex in A_{k-1} and any vertex b_t in B_{k-2}, as b_t would accept such a proposal and then not be in B_{k-2}. So the first edge of p has to be from $B_0 \times A_0$ and the last edge has to be from $B_{k-1} \times A_{k-1}$ (see Figure 9). Thus the shortest augmenting path that is possible is the following:

$$b_0 - (x_0, y_0) - (x_1, y_1) \cdots (x_{k-2}, y_{k-2}) - (x_{k-1}, y_{k-1}) - a_{t+1},$$

where for $0 \le i \le k-1$, the vertex x_i is in A_i, the edge (x_i, y_i) is in M_k, and thus the vertex y_i is in B_i. So there have to be at least k edges of M_k in p. Hence $|p| \ge 2k+1$. ∎

COROLLARY 1. $|M_k| \ge \frac{k}{k+1}|M_{\max}|$, *where $M_{\max}$ is a maximum size matching in G.*

Proof. Every path in $M_k \oplus M_{\max}$ that is augmenting with respect to M_k has length at least $2k+1$ (by Lemma 8). So every such path has t edges of M_k and $t+1$ edges of $M_{\max}$, for some $t \ge k$. Hence $|M_k| \ge \frac{k}{k+1}|M_{\max}|$. ∎

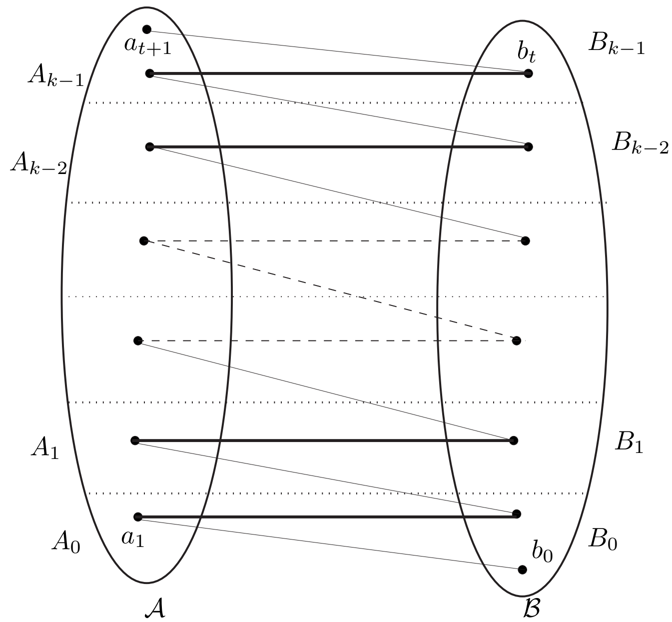

FIG. 9. *In any augmenting path $\langle b_0, a_1, b_1, \ldots, b_t, a_{t+1}\rangle$ with respect to M_k, the vertices b_0 and a_1 have to be in B_0 and A_0, respectively; similarly, the vertices a_{t+1} and b_t have to be in A_{k-1} and B_{k-1}, respectively.*

This proves the lower bound on the size of M_k. We now bound its unpopularity factor from above via Theorem 5. First, we show the following simple lemma.

LEMMA 9. *Every edge labeled $(1,1)$ has to be in $\cup_{\ell=0}^{k-2}(A_\ell \times \cup_{j>\ell} B_j)$.*

Proof. There is no blocking edge in $A_\ell \times \cup_{j\le\ell} B_j$ for any ℓ, since M_k restricted to edges in $A_\ell \times \cup_{j\le\ell} B_j$ is obtained by running the Gale–Shapley algorithm on these vertices, with the men in A_ℓ proposing and the women in $\cup_{j\le\ell} B_j$ disposing. Thus every blocking edge to M_k has to be in $\cup_{\ell=0}^{k-2}(A_\ell \times \cup_{j>\ell} B_j)$. ∎

THEOREM 5. *Let $\rho = \langle y_0, x_1, y_1, \ldots, x_{t-1}, y_{t-1}, x_t\rangle$ be an alternating path with respect to M_k in G, where $(x_i, y_i) \in M_k$ for $i \ge 1$. Then the number of edges labeled $(1,1)$ in ρ is at most $h-\ell$ plus the number of edges labeled $(-1,-1)$ in ρ, where $y_0 \in A_\ell$ and $x_t \in B_h$.*

Proof. Let $\rho = \langle y_0, x_1, \ldots, y_{t-1}, x_t\rangle$ be an alternating path where each $(x_i, y_i) \in M_k$. We are given that $y_0 \in A_\ell$ and $x_t \in B_h$. The claim is that the number of $(1,1)$ edges in ρ is at most $h-\ell$ plus the number of $(-1,-1)$ edges in ρ. We prove this claim by induction on the number of $(-1,-1)$ edges in ρ.

Suppose there are no $(-1,-1)$ edges in ρ. Then we will show that the number of $(1,1)$ edges in ρ is at most $h-\ell$. We know from Lemma 6 that there are no edges between A_ℓ and $\cup_{j<\ell-1} B_j$ and from Lemma 7 that there are only $(-1,-1)$ edges between A_ℓ and $B_{\ell-1}$. Thus the entire path ρ is stuck in layers greater than or equal to ℓ. There are no $(1,1)$ edges in $A_\ell \times B_\ell$. So while the x_i's are in B_ℓ (which forces the y_i's to be in A_ℓ), we do not encounter any $(1,1)$ edge in ρ. Hence it is necessary to traverse an edge in ρ between some $y_j \in A_\ell$ and $x_{j+1} \in B_{\ell'}$, for some $\ell' > \ell$, so that a $(1,1)$ edge is encountered (by Lemma 9). For any ℓ and $\ell' > \ell$, once we traverse an edge between A_ℓ and $B_{\ell'}$, the rest of the path ρ gets stuck in layers greater than or equal to ℓ'. Once the path jumps to a higher layer, since there is no way it can come back to a lower layer (due to the absence of $(-1,-1)$ edges in ρ), it follows that we are allowed at most $h-\ell$ jumps in layer numbers from $y_0 \in A_\ell$ to $x_t \in B_h$. Thus we can traverse at most $h-\ell$ edges labeled $(1,1)$ in ρ. This settles the base case.

We assume by induction hypothesis that the claim is true when the number of $(-1,-1)$ edges in any alternating path is at most $i-1$. Let ρ have $i \geq 1$ edges labeled $(-1,-1)$, and let (y_{j-1}, x_j) be one of these $(-1,-1)$ edges in ρ. Let $y_{j-1} \in A_r$ and $x_j \in B_s$ (see Figure 10). The subpath $\langle x_{j-1}, y_{j-1}, x_j, y_j\rangle$ consists of one $(-1,-1)$ edge and two edges of M_k. Deleting this subpath from ρ, we get two alternating subpaths ρ_1 and ρ_2, where $\rho_1 = \langle y_0, x_1, \ldots, x_{j-1}\rangle$ and $\rho_2 = \langle y_j, x_{j+1}, \ldots, x_t\rangle$. Since the number of $(-1,-1)$ edges in ρ_1 and in ρ_2 is at most $i-1$, by applying the induction hypothesis on ρ_1 and on ρ_2, it follows that the number of $(1,1)$ edges in ρ is at most the number of $(-1,-1)$ edges in ρ_1 plus the number of $(-1,-1)$ edges in $\rho_2 + (r-\ell) + (h-s)$, where the $(r-\ell)$ term comes from ρ_1 and the $(h-s)$ term comes from ρ_2.

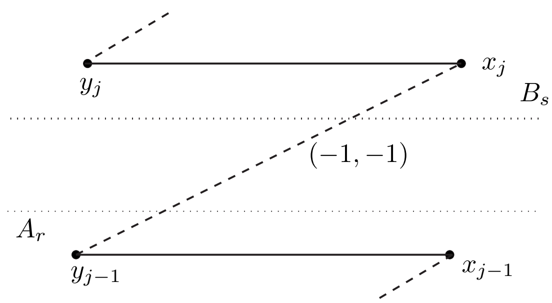

FIG. 10. *A subpath of ρ consisting of two matched edges and a $(-1,-1)$ edge.*

The number of $(-1,-1)$ edges in ρ_1 plus the number of $(-1,-1)$ edges in ρ_2 is one less than the number of $(-1,-1)$ edges in ρ. So the number of $(1,1)$ edges in ρ is at most the number of $(-1,-1)$ edges in $\rho + (r-\ell) + (h-s) - 1$. Since there is an edge between $y_{j-1} \in A_r$ and $x_j \in B_s$, it follows from Lemma 6 that $s \geq r-1$. Hence $h - \ell + r - s - 1 \leq h - \ell$. Thus the claim holds when the number of $(-1,-1)$ edges in ρ is i. This completes the proof of Theorem 5. $\square$

Observe that Theorem 5 does not have to impose any conditions on h and ℓ to ensure that "$h-\ell$ plus the number of $(-1,-1)$ edges in ρ" is nonnegative. In fact, by Lemmas 6 and 7, there can be no alternating path $\rho = \langle y_0, x_1, \ldots, y_{t-1}, x_t\rangle$ with respect to M_k such that $h-\ell$ plus the number of $(-1,-1)$ edges in ρ is negative, where $y_0 \in A_\ell$ and $x_t \in B_h$.

Theorem 6, stated below, uses Theorem 5 to generalize Theorem 4. Note that parts 1 and 2 in this theorem are the same as parts 1 and 2 in Theorem 4, while part 3 involves multiplication by $(k-1)$ on its right-hand side.

THEOREM 6. *For any matching M' in G, the following three statements hold:*

1. *If ρ is an alternating cycle in $M_k \oplus M'$, then $\phi(M_k \oplus \rho, M_k) \leq \phi(M_k, M_k \oplus \rho)$.*
2. *If ρ is an alternating path in $M_k \oplus M'$ such that at least one endpoint of ρ is unmatched in M_k, then $\phi(M_k \oplus \rho, M_k) \leq \phi(M_k, M_k \oplus \rho)$.*
3. *If ρ is an alternating path in $M_k \oplus M'$ such that both endpoints of ρ are matched in M_k, then $\phi(M_k \oplus \rho, M_k) \leq (k-1) \cdot \phi(M_k, M_k \oplus \rho)$.*

Proof. Let ρ be an alternating cycle in $M_k \oplus M'$. Every edge of M_k is in $\cup_{\ell=0}^{k-1}(A_\ell \times B_\ell)$. Let (a,b) be an edge in $M_k \cap \rho$. So $\rho \setminus \{(a,b)\}$ is an alternating path $\langle a, \ldots, b\rangle$ where $a \in A_t$ and $b \in B_t$, for some t. Hence it follows from Theorem 5 that in ρ, the number of edges labeled $(1,1)$ is at most the number of $(-1,-1)$ edges. As the other edge labels are $(-1,1)$ or $(1,-1)$, it follows that among the vertices of ρ, the number of 1 votes (votes in favor of M') is at most the number of -1 votes (votes in favor of M_k). Thus $\phi(M_k \oplus \rho, M_k) \le \phi(M_k, M_k \oplus \rho)$ in part 1.

Let ρ be an alternating path in $M_k \oplus M'$ that begins with a vertex unmatched in M_k. Claim 2 states that every unmatched vertex has to be in $A_{k-1} \cup B_0$. Since either $h = 0$ or $\ell = k-1$ here, it follows from Theorem 5 that the number of edges labeled $(1,1)$ in ρ is at most the number of edges labeled $(-1,-1)$ in ρ. As the other edge labels are $(-1,1)$ or $(1,-1)$, it again follows that among the vertices of ρ, the number of 1 votes is at most the number of -1 votes. Hence $\phi(M_k \oplus \rho, M_k) \le \phi(M_k, M_k \oplus \rho)$ in part 2.

Let ρ be an alternating path in $M_k \oplus M'$ where both endpoints of ρ are matched in M_k. That means that neither endpoint is matched in M'. Note that these two vertices prefer M_k to $M_k \oplus \rho$. Let the number of $(-1,-1)$ edges in ρ be s. Theorem 5 tells us that the number of $(1,1)$ edges in ρ is at most $s+k-1$. Each of the other edges (say, there are t of these other edges) is labeled either $(-1,1)$ or $(1,-1)$. Then among all the vertices of ρ, we have at most $2k-2+2s+t$ that prefer $M_k \oplus \rho$ to M_k and at least $2+2s+t$ that prefer M_k to $M_k \oplus \rho$. Hence $\phi(M_k \oplus \rho, M_k) \le (k-1) \cdot \phi(M_k, M_k \oplus \rho)$, as $k \ge 2$ and $s, t \ge 0$. This finishes the proof of Theorem 6. ∎

We are now ready to prove Theorems 2 and 3 stated in section 1.

Proof of Theorem 2. Consider the matching M_{n_0} obtained by running Algorithm 2 with $k = n_0$. Corollary 1 gives us the following bound on the size of M_{n_0}:

$$|M_{n_0}| \ge \frac{n_0}{n_0+1}|M_{\max}| = \left(1 - \frac{1}{n_0+1}\right)|M_{\max}|. \tag{2}$$

Since $|M_{\max}| \le \min(|\mathcal{A}|, |\mathcal{B}|) = n_0$, (2) implies that $|M_{n_0}| = |M_{\max}|$. Thus M_{n_0} is a maximum size matching in G.

The matchings $M_{\max}$ and M_{n_0} are both maximum matchings in G. So $M_{n_0} \oplus M_{\max}$ is a collection of alternating cycles and even length alternating paths. So each alternating path has one endpoint unmatched in M_{n_0}. Hence part 3 of Theorem 6 does not apply here. Let ρ be any alternating path or cycle in $M_{n_0} \oplus M_{\max}$. We have $\phi(M_{n_0} \oplus \rho, M_{n_0}) \le \phi(M_{n_0}, M_{n_0} \oplus \rho)$ by parts 1 and 2 of Theorem 6. Thus

$$\begin{aligned}\phi(M_{\max}, M_{n_0}) &= \sum_{\rho \in M_{n_0} \oplus M_{\max}} \phi(M_{n_0} \oplus \rho, M_{n_0}) \\ &\le \sum_{\rho \in M_{n_0} \oplus M_{\max}} \phi(M_{n_0}, M_{n_0} \oplus \rho) \\ &= \phi(M_{n_0}, M_{\max}).\end{aligned}$$

Since $\phi(M_{\max}, M_{n_0}) \le \phi(M_{n_0}, M_{\max})$ for any maximum matching $M_{\max}$, it follows that M_{n_0} is popular within the set $\mathcal{M}$ of maximum matchings in G. Thus M_{n_0} satisfies all the properties claimed in Theorem 2. We know that the time taken to compute M_{n_0} is $O(mn_0)$. This completes the proof of Theorem 2 stated in section 1. ∎

Proof of Theorem 3. For any matching M', consider $M_k \oplus M'$. For any alternating cycle/path $\rho \in M_k \oplus M'$, we know from Theorem 6 that $\phi(M_k \oplus \rho, M_k) \le (k-1) \cdot \phi(M_k, M_k \oplus \rho)$. So we have

$$\begin{aligned}\phi(M', M_k) &= \sum_{\rho \in M_k \oplus M'} \phi(M_k \oplus \rho, M_k)\\ &\le \sum_{\rho \in M_k \oplus M'} (k-1)\cdot \phi(M_k, M_k \oplus \rho)\\ &= (k-1)\cdot \phi(M_k, M').\end{aligned}$$

Hence $\Delta(M_k, M') \le k-1$ for all matchings $M' \neq M_k$, and thus $u(M_k) \le k-1$. We will now show that if $|M'| \ge |M_k|$ for any matching M' in G, then $\phi(M_k, M') \ge \phi(M', M_k)$.

Let ρ be an alternating cycle or path in $M_k \oplus M'$. If ρ is an alternating cycle or an even length alternating path, then we know from parts 1 and 2 of Theorem 6 that $\phi(M_k \oplus \rho, M_k) \le \phi(M_k, M_k \oplus \rho)$. So what is left is the case when ρ is an odd length alternating path.

We cannot claim that $\phi(M_k \oplus \rho, M_k) \le \phi(M_k, M_k \oplus \rho)$ for an odd length alternating path ρ. However, we will be able to show that $\sum_{\rho \in O} \phi(M_k \oplus \rho, M_k) \le \sum_{\rho \in O} \phi(M_k, M_k \oplus \rho)$, where O is the set of all odd length alternating paths in $M_k \oplus M'$. For any $\rho \in M_k \oplus M'$,

$$\phi(M_k \oplus \rho, M_k) - \phi(M_k, M_k \oplus \rho) \;=\; \sum_{\substack{u \in \rho\\ \text{unmatched in } M'}} -1 \;+\; \sum_{e \in \rho \cap M'} (\alpha_e + \beta_e), \tag{3}$$

where $\alpha_e = \mathsf{vote}_u(v, M_k(u))$ and $\beta_e = \mathsf{vote}_v(u, M_k(v))$ for edge $e = (u, v)$ in $\rho \cap M'$.

Let $\rho = \langle y_0, \ldots, x_t\rangle$. There are two subcases when ρ is an odd length alternating path in $M_k \oplus M'$: (i) y_0 and x_t are unmatched in M_k, or (ii) y_0 and x_t are unmatched in M'.

Consider subcase (i). We know from Claim 2 that $y_0 \in A_{k-1}$ and $x_t \in B_0$. Theorem 5 tells us that the number of edges labeled $(-1,-1)$ in ρ is at least $(k-1)$ plus the number of edges labeled $(1,1)$ in ρ. So if there are r edges labeled $(1,1)$ in ρ, then the number of edges labeled $(-1,-1)$ in ρ is at least $r+k-1$. Every other edge in ρ is labeled either $(1,-1)$ or $(-1,1)$. Hence the right-hand side of (3) is at most $2r - 2(r+k-1) = -2(k-1)$.

Consider subcase (ii). Since the vertices y_0 and x_t are unmatched in M', the first term on the right-hand side of (3) equals -2. Theorem 5 tells us that the number of edges labeled $(1,1)$ in ρ is at most $(k-1)$ plus the number of edges labeled $(-1,-1)$ in ρ. Thus if there are s edges labeled $(-1,-1)$ in ρ, then the number of edges labeled $(1,1)$ in ρ is at most $s+k-1$. Every other edge in ρ is labeled either $(1,-1)$ or $(-1,1)$. Hence the right-hand side of (3) is at most $-2 + 2(s+k-1) - 2s = 2(k-2)$.

Recall that O is the set of odd length alternating paths in $M_k \oplus M'$. Among the paths in O, let there be t_1 paths whose endpoints are unmatched in M_k, and let there be t_2 paths whose endpoints are unmatched in M'. Since $|M'| \ge |M_k|$, we have $t_1 \ge t_2$:

$$\begin{aligned}\sum_{\rho \in O} \phi(M_k \oplus \rho, M_k) &\le \sum_{\rho \in O} \phi(M_k, M_k \oplus \rho) - 2(k-1)t_1 + 2(k-2)t_2\\ &\le \sum_{\rho \in O} \phi(M_k, M_k \oplus \rho) \qquad \{\text{since } t_1 \ge t_2\}.\end{aligned}$$

Thus we have $\phi(M', M_k) \leq \phi(M_k, M')$ for any matching M' whose size is at least $|M_k|$. We have $|M_k| \geq \frac{k}{k+1}|M_{\max}|$ by Corollary 1. We also know that the time taken to compute M_k is $O(km)$. We can now conclude Theorem 3 stated in section 1. ∎

Remark. Note that the matching M_k need not be popular among matchings of size at least $\frac{k}{k+1}|M_{\max}|$. Consider the following instance on 10 vertices by taking a copy of the instance on four vertices described in Figure 1 along with a copy of the instance on six vertices described in Figure 2; no new edges are added. $M_2 = \{(x_1, y_0), (x_2, y_1), (a_1, b_1), (a_2, b_2)\}$ is a maximum size popular matching here and $M_3 = \{(x_1, y_0), (x_2, y_1), (a_1, b_0), (a_2, b_1), (a_3, b_2)\}$. So $|M_2| > \frac{3}{4}|M_3|$. Since M_2 is more popular than M_3, the matching M_3 is not popular among matchings of size at least $\frac{3}{4}|M_3|$.

4. Conclusions and open problems. We considered the problem of computing matchings with large size and low unpopularity factor in a stable marriage instance $G = (\mathcal{A} \cup \mathcal{B}, E)$ with incomplete lists, where each vertex ranks its neighbors in a strict order of preference. For any integer $k \geq 2$, we extended the Gale–Shapley stable matching algorithm to k layers, to show that a matching M_k whose size is at least $\frac{k}{k+1}|M_{\max}|$ and whose unpopularity factor is at most $k-1$ always exists. Moreover, any matching whose size is at least the size of M_k cannot be more popular than M_k. Such a matching M_k can be computed in $O(km)$ time, where $|E| = m$. When $k = 2$, we showed that the resulting matching M_2 will be a maximum size popular matching in G.

An open problem is to efficiently find a maximum size matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$ whose unpopularity factor is the least among all maximum size matchings in G. Another open problem is to settle the complexity of determining whether a general graph $G = (V, E)$ with strict preference lists, also called a *roommates* instance, admits a popular matching or not.

Acknowledgment. Many thanks to the reviewers of the conference version of this paper and an earlier draft of this paper for their helpful comments and suggestions.

REFERENCES

[1] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn, *Popular matchings*, SIAM J. Comput., 37 (2007), pp. 1030–1045 .

[2] P. Biro, R. W. Irving, and D. F. Manlove, *Popular matchings in the marriage and roommates problems*, in Proceedings of the Seventh International Conference on Algorithms and Complexity (CIAC), 2010, pp. 97–108.

[3] P. Biro, D. F. Manlove, and S. Mittal, *Size versus stability in the marriage problem*, Theoret. Comput. Sci., 411 (2010), pp. 1828–1841.

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

[5] D. Gale and M. Sotomayor, *Some remarks on the stable matching problem*, Discrete Appl. Math., 11 (1985), pp. 223–232.

[6] P. Gärdenfors, *Match making: Assignments based on bilateral preferences*, Behav. Sci., 20 (1975), pp. 166–173.

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

[8] C.-C. Huang and T. Kavitha, *Popular matchings in the stable marriage problem*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 6755, Springer-Verlag, Berlin, 2011, pp. 666–677.

[9] T. Kavitha, *Popularity vs. maximum cardinality in the stable marriage setting*, in Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, 2012, pp. 123–134.

[10] T. Kavitha, J. Mestre, and M. Nasre, *Popular mixed matchings*, in Proceedings of the 36th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 5555, Springer-Verlag, Berlin, 2009, pp. 574–584.

[11] T. Kavitha and M. Nasre, *Note: Optimal popular matchings*, Discrete Appl. Math., 157 (2009), pp. 3181–3186.

[12] Z. Király, *Better and simpler approximation algorithms for the stable marriage problem*, Algorithmica, 60 (2011), pp. 3–20.

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

[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, Lecture Notes in Comput. Sci. 4168, Springer-Verlag, Berlin, 2006, pp. 492–503.

[15] M. McCutchen, *The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences*, in Proceedings of the 8th Latin American Theoretical Informatics Symposium (LATIN 2008), Lecture Notes in Comput. Sci. 4957, Springer-Verlag, Berlin, 2008, pp. 593–604.

[16] E. McDermid and R. W. Irving, *Popular matchings: Structure and algorithms*, in Proceedings of the 15th Annual International Computing and Combinatorics Conference, 2009, pp. 506–515.

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