Theoretical Computer Science 412 (2011) 1263–1274

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Popular matchings with variable item copies☆

Telikepalli Kavitha [a], Meghana Nasre [b,*]

[a] Tata Institute of Fundamental Research, Mumbai, India

[b] Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:
Received 11 March 2010
Received in revised form 3 October 2010
Accepted 27 December 2010
Communicated by P. Spirakis

Keywords:
Bipartite graphs
Matchings
One-sided preference lists
NP-hardness

ABSTRACT

We consider the problem of matching people to items, where each person ranks a subset of items in an order of preference, possibly involving ties. There are several notions of optimality about how to best match a person to an item; in particular, *popularity* is a natural and appealing notion of optimality. A matching M^* is popular if there is no matching M such that the number of people who prefer M to M^* exceeds the number who prefer M^* to M. However, popular matchings do not always provide an answer to the problem of determining an optimal matching since there are simple instances that do not admit popular matchings. This motivates the following extension of the popular matchings problem:

- Given a graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where $\mathcal{A}$ is the set of people and $\mathcal{B}$ is the set of items, and a list $\langle c_1, \ldots, c_{|\mathcal{B}|}\rangle$ denoting upper bounds on the number of copies of each item, does there exist $\langle x_1, \ldots, x_{|\mathcal{B}|}\rangle$ such that for each i, having x_i copies of the i-th item, where $1 \le x_i \le c_i$, enables the resulting graph to admit a popular matching?

In this paper we show that the above problem is NP-hard. We show that the problem is NP-hard even when each c_i is 1 or 2. We show a polynomial time algorithm for a variant of the above problem where the total increase in copies is bounded by an integer k.

1. Introduction

In this paper we consider the problem of matching people to items, where each person ranks a subset of items in an order of preference possibly involving ties, that is, preference lists are *one-sided*. Our input is a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where $\mathcal{A}$ is the set of people and $\mathcal{B}$ is the set of items, and $E = E_1 \dot\cup \cdots \dot\cup E_r$ is the set of edges, where E_t is the set of edges having rank t. For any $a \in \mathcal{A}$, we say a prefers item b to item b' if the rank of edge (a, b) is smaller than the rank of edge (a, b'). The goal is to come up with an *optimal* matching of people to items. Several notions of optimality like rank-maximality [8], maximum-utility, Pareto-optimality [2,1,16] have been studied in the literature for matchings with one-sided preferences. But most of these notions use the absolute ranks specified by people over items to distinguish between a pair of matchings. One criterion that does not use numerical ranks is the notion of popularity.

Let $M(a)$ denote the item to which a person a is matched in a matching M. We say that a person a prefers matching M to M' if (i) a is matched in M and unmatched in M', or (ii) a is matched in both M and M', and a prefers $M(a)$ to $M'(a)$.

☆ Work done as part of the DST-MPG partner group "Efficient Graph Algorithms" at IISc Bangalore. A preliminary version of this work appeared in the 20th International Symposium on Algorithms and Computation, ISAAC'09 [11].

* Corresponding address: Computer Science and Automation Department, Indian Institute of Science, Bangalore, 560012, India. Tel.: +91 80 22932368; fax: +91 80 23602911.

E-mail addresses: kavitha@tcs.tifr.res.in (T. Kavitha), meghana@csa.iisc.ernet.in, meghanamande@gmail.com (M. Nasre).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.12.067

Definition 1. M is *more popular than* M', denoted by $M \succ M'$, if the number of people who prefer M to M' is higher than those that prefer M' to M. A matching M^* is *popular* if there is no matching that is more popular than M^*.

Popular matchings were first introduced by Gärdenfors [5] in the context of stable matchings. The notion of popularity is an attractive notion of optimality since it is based on *relative* ranking rather than the absolute ranks used by any person; also popular matchings can be considered stable in the sense that no majority vote of people can force a migration to another matching. Unfortunately there exist simple instances that do not admit any popular matching. Abraham et al. [3] designed efficient algorithms for determining if a given instance admits a popular matching and computing one, if it exists.

A simple example that does not admit a popular matching is the following: let $\mathcal{A} = \{a_1, a_2, a_3\}$, $\mathcal{B} = \{b_1, b_2, b_3\}$, and each person prefers b_1 to b_2, and b_2 to b_3. Consider the three symmetrical matchings $M_1 = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$, $M_2 = \{(a_1, b_3), (a_2, b_1), (a_3, b_2)\}$ and $M_3 = \{(a_1, b_2), (a_2, b_3), (a_3, b_1)\}$. None of these matchings is popular, since $M_1 \prec M_2$, $M_2 \prec M_3$, and $M_3 \prec M_1$. In fact, it turns out that this instance admits no popular matching, the problem being that the *more popular than* relation is not transitive. Our focus in this paper will be on such instances that do not admit a popular matching. Intuitively, the absence of a popular matching in an instance is due to a small set of items being in too much demand by a large number of people. Suppose that we are allowed to add extra copies or *duplicates* of items and that these duplicates are indistinguishable from the original item. If there were no bounds on the number of copies that we could add for an item, a straightforward solution would be to have as many copies of an item as the number of people that demand the item as their top choice. For example, in the above instance that does not admit a popular matching, say we have 2 additional copies of b_1 (call them b_1' and b_1''). Thus, we have 3 copies of b_1 in our instance now and it is easy to see that the matching $\tilde{M} = \{(a_1, b_1), (a_2, b_1'), (a_3, b_1'')\}$ is a popular matching for the new instance. But our assumption of allowing arbitrarily many copies of an item need not be true in practice. Fortunately, popular matchings do not require each person to be matched to her top choice item. In the above instance it is possible to get a popular matching by making just one extra copy of item b_1 and not duplicating any other items. In this sense, we expect to improve the situation in terms of popularity by having additional copies of some items within the specified bounds.

Such a solution of making extra copies is appealing when the items represent books or DVDs. Say, along with a DVD b_i comes a license which allows us to make a specified number c_i many copies of that particular DVD. If we make up to c_i copies of that DVD, then we do not incur any additional cost. Our goal then is to make the appropriate number of copies of every DVD so that the resulting instance admits a popular matching. Another relevant scenario is when the set $\mathcal{B}$ represents training programs. Any training program b_i can be run for a single person, but some training programs may be able to accommodate up to c_i many people. We call this the capacity of a training program and we wish to fix the capacity of each training program so as to enable the resulting instance to admit a popular matching. It may seem from the above examples that having as many copies of a DVD as are allowed or stretching a training program to its maximum capacity should be the best thing to do. We show an example in the next section which illustrates that having as many copies as possible does not always help popularity.

The problem of fixing copies. Given a graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ where $\mathcal{B} = \{b_1, \dots, b_{|\mathcal{B}|}\}$ and a list $\langle c_1, \dots, c_{|\mathcal{B}|}\rangle$ of upper bounds on the number of copies, does there exist an $\langle x_1, \dots, x_{|\mathcal{B}|}\rangle$ such that for each $i \in \{1, \dots, |\mathcal{B}|\}$, having x_i copies of the i-th item, where $1 \le x_i \le c_i$, enables the resulting graph to admit a popular matching?

We assume that G does not admit a popular matching. Our problem is to determine if by fixing the copies of each item appropriately, whether the resulting graph admits a popular matching or not. We now define a special case of this problem which we call the 1-or-2 copies problem.

The 1-or-2 copies problem. In this case, each c_i is either 1 or 2. Note that when all the c_i's are 1, this is the standard popular matching problem. Thus the *1-or-2 copies problem* is a generalization of the popular matching problem. Here we have a subset K of items which can be duplicated, that is, the items in K can have up to 2 copies in the resulting instance, while the rest of the items will have a single copy. The problem is to determine if by duplicating some elements in K we get a graph that admits a popular matching.

The deleting-some-items problem. When the input instance does not admit a popular matching, another possibility is to delete some items so that the resulting graph admits a popular matching. Here we assume that only certain items: elements of some subset $\mathcal{B}'$ of $\mathcal{B}$ are allowed to be deleted; otherwise there is a trivial solution that says "delete all items, then the empty matching is popular". The problem is to determine if there exists a subset $T \subseteq \mathcal{B}'$ such that by deleting the items in T, the resulting graph admits a popular matching. If the 1-or-2 copies problem is NP-hard, then it is easy to show that the deleting-some-items problem is also NP-hard.

Given an instance $G = (\mathcal{A} \cup \mathcal{B}, E)$ and $K \subseteq \mathcal{B}$ of the 1-or-2 copies problem, we create an instance of the deleting-some-items problem as follows: our graph is $H = (\mathcal{A} \cup \tilde{\mathcal{B}}, E)$ where each item in K has 2 copies. Thus the set of items $\tilde{\mathcal{B}}$ here can be considered as $\{b_1, \dots, b_{|\mathcal{B}|}, b_1', \dots, b_k'\}$ where $K = \{b_1, \dots, b_k\}$ and $b_1', \dots, b_k'$ are *identical copies* of these items. Set the subset $\mathcal{B}'$ of items that may be deleted to $\{b_1', \dots, b_k'\}$. That is, the first copy of an item may not be deleted, while each of the *second* copies of items that we created can be deleted. Thus, any efficient algorithm for the deleting-some-items problem implies an efficient algorithm for the "1-or-2 copies" problem.

1.0.1. Related work

Subsequent to the work on popular matching algorithms in [3], Manlove and Sng [13] generalized the algorithms of [3] to the capacitated case in the context of house allocation, there items were called houses and houses had capacities.

This problem can be easily solved using the algorithm in [3] by including c_i copies of item b_i for each i. A faster algorithm for determining if such an instance admits a popular matching and computing one if it exists, was shown in [13] for this problem.

Mestre showed efficient algorithms for the *weighted* popular matching problem in [15], here each person is assigned a priority or weight, and the definition of popularity takes into account the priorities of the people. Mahdian [12] considered the problem of when random graphs (that is, preference lists are randomly constructed) admit popular matchings and showed that a popular matching exists with high probability in such graphs, when the number of items is a factor of $\alpha \approx 1.42$ larger than the number of people.

In order to deal with the problem of the input instance not admitting a popular matching, the following extensions of popular matchings have been considered so far.

- *Least unpopular matching*: A natural extension when the graph does not admit a popular matching is to ask for a least unpopular matching. The *unpopularity margin* of a matching M, call it $u(M)$, is $\max_{M'}$ |people who prefer M' to $M| - |$people who prefer M to $M'|$. The least unpopularity margin matching is that matching M with the least value of $u(M)$. McCutchen [14] showed that computing such a matching is NP-hard. In [7] Huang et al. gave efficient algorithms to compute matchings with bounded values of these unpopularity measures in certain graphs.
- *Mixed matchings*: Very recently, Kavitha et al. [10] considered the problem of computing a probability distribution over matchings, also called a *mixed matching*, that is popular. It was shown that every instance admits a popular mixed matching and a polynomial time algorithm was given to compute such a probability distribution. Although, popular mixed matchings are guaranteed to exist and are efficiently computable, they may not be an acceptable solution when the solution needs to be a pure matching.

1.0.2. Our contributions

In this paper we consider the following extension to the popular matching problem: does having extra copies of items within certain bounds, yield a graph that admits a popular matching? The main results in this paper are:

- The above problem is NP-hard. In fact, the 1-or-2 copies problem is NP-hard. As a consequence the problem of deleting-some-items is NP-hard.
- Our reduction constructs an instance of the 1-or-2 copies problem with preference lists of length just 2. In fact, each of these preference lists has a unique top choice item and there may be at most 2 items tied as second choice items. We also show that when preferences lists have length 2 with ties allowed only in the first position (i.e., the second choice item is unique), then the problem becomes solvable in polynomial time.
- We show that the 1-or-2 copies problem remains NP-hard even if preference lists are derived from a *master list*.
- We consider a variant of the above problem where we only have to maintain an upper bound k on the *total* number of extra copies of all the items, rather an upper bound on the number of copies of *each* item. That is, here we are given a graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ and an integer $k \geq 0$ and we have to decide whether there exist $\Delta_1, \ldots, \Delta_{|\mathcal{B}|}$ where each $\Delta_i \geq 1$ and $\sum_i \Delta_i \leq k$, such that having $\Delta_i + 1$ copies of the i-th item, for $1 \leq i \leq |\mathcal{B}|$, enables the resulting instance to admit a popular matching. Note that we add Δ_i extra copies of the i-th item and in this case the sum of the extra copies of all items is bounded by k. We show a polynomial time algorithm for this variant.

A preliminary version of these results can be found in [11].

Organization of the paper. Section 2 outlines the algorithm from [3] to compute a popular matching. Section 3 contains our NP-hardness proofs. Section 4 shows a polynomial time algorithm for the variant where only the sum of the extra copies is bounded.

2. Preliminaries

We first review the algorithmic characterization of popular matchings given in [3]. As was done in [3], it will be convenient to add a unique last item ℓ_a at the end of a's preference list for each person $a \in \mathcal{A}$. We will henceforth refer to this graph as $G = (\mathcal{A} \cup \mathcal{B}, E)$. In this way, it can be assumed that every matching in G is $\mathcal{A}$-complete, since any unmatched person a can be paired with ℓ_a.

A maximum matching M in a bipartite graph $G_1 = (\mathcal{A} \cup \mathcal{B}, E_1)$ has the following important properties: $M \cap E_1$ defines a partition of $\mathcal{A} \cup \mathcal{B}$ into three disjoint sets: *odd* ($\mathcal{O}$), *even* ($\mathcal{E}$), and *unreachable* ($\mathcal{U}$).

- A vertex u belongs to the set $\mathcal{E}$ (resp. $\mathcal{O}$) if there is an *even* (resp. *odd*) length alternating path in G_1 from an unmatched vertex to u.
- A vertex u belongs to the set $\mathcal{U}$, that is, it is *unreachable*, if there is no alternating path in G_1 from an unmatched vertex to u.

The following lemma, proved in [6], is well-known in matching theory. We include its proof for completeness.

Lemma 1. *Let $\mathcal{E}$, $\mathcal{O}$ and $\mathcal{U}$ be the sets of nodes defined by G_1 and $M \cap E_1$ above. Then*

(a) *$\mathcal{E}$, $\mathcal{O}$ and $\mathcal{U}$ are pairwise disjoint, and independent of the maximum matching $M \cap E_1$.*
(b) *In any maximum matching of G_1, every node in $\mathcal{O}$ is matched with a node in $\mathcal{E}$, and every node in $\mathcal{U}$ is matched with another node in $\mathcal{U}$. The size of a maximum matching is $|\mathcal{O}| + |\mathcal{U}|/2$.*
(c) *No maximum matching of G_1 contains an edge between a node in $\mathcal{O}$ and a node in $\mathcal{O} \cup \mathcal{U}$. Also, G_1 contains no edge between a node in $\mathcal{E}$ and a node in $\mathcal{E} \cup \mathcal{U}$.*

Proof. (a) The set $\mathcal{U}$ is disjoint from $\mathcal{E}$ and $\mathcal{O}$ by definition. To prove that $\mathcal{E}$ is disjoint from $\mathcal{O}$, assume that a node u is reachable by an *even* length path from a node a and an *odd* length path from a node b. Note that $a \neq b$ since G is bipartite. Composing the two paths, we get an augmenting path in G with respect to M, contradicting the maximality of M.

To prove that this partition in independent of M, let N be any other maximum cardinality matching in G. $M \oplus N$ consists of alternating paths and cycles and each of these paths and cycles are of *even* length. Since the graph is bipartite, it is immediate that the cycle have to be of even length. For paths, assume that a path has more edges from N, then such a path is an augmenting path with respect to M, a contradiction to the maximality of M. A similar argument holds if there are more edges from M. Using these paths and cycles to switch from M to N does not alter the *odd/even/unreachable* status of nodes, hence the partition is independent of the maximum cardinality matching.

(b) If a matched node u is reachable from a free node by an *odd* length path with respect to any maximum cardinality matching, then its partner is reachable by an *even* length path. Thus, all edges in any maximum matching of G are either $\mathcal{O}\mathcal{E}$ or $\mathcal{U}\mathcal{U}$ edges. Further, any node in $\mathcal{U}$ must be matched by a maximum matching, for, if not, the node is reachable with an *even* length (zero length) path from itself. Also a node in $\mathcal{O}$ must be matched by a maximum matching since an *odd* length alternating path starting and ending with a free node is an augmenting path. Thus, the size of any maximum matching is $|\mathcal{O}| + |\mathcal{U}|/2$.

(c) Nodes in $\mathcal{E}$ are reachable by an alternating path from an unmatched node. Such paths end in a matching edge. If there were an edge between two nodes in $\mathcal{E}$, we could use that to construct an augmenting path, contradicting maximality. Finally, if there were an edge between a node in $\mathcal{E}$ and a node in $\mathcal{U}$, such an edge would be a non-matching edge. We could use that to extend the alternating path and reach the node in $\mathcal{U}$, a contradiction to the definition of nodes in $\mathcal{U}$. □

As every maximum cardinality matching matches all vertices $u \in \mathcal{O} \cup \mathcal{U}$ we call these vertices *critical* as opposed to vertices $u \in \mathcal{E}$ which are called *non-critical*. Using the above partition, the following definitions can be made:

Definition 2. For each $a \in \mathcal{A}$, define $f(a)$ to be the set of top choice items for a. Define $s(a)$ to be the set of a's most-preferred *non-critical* items in G_1.

Theorem 1 (*[3]*)**.** *A matching M is popular in G iff (i) M is a maximum matching of $G_1 = (\mathcal{A} \cup \mathcal{B}, E_1)$, and (ii) for each person a, $M(a) \in f(a) \cup s(a)$.*

The algorithm for solving the popular matching problem is now straightforward: each $a \in \mathcal{A}$ determines the sets $f(a)$ and $s(a)$. An $\mathcal{A}$-complete matching that is a maximum matching in G_1 and that matches each a to an item in $f(a) \cup s(a)$ needs to be determined. If no such matching exists, then G does not admit a popular matching. The popular matching algorithm from [3] is presented in the Appendix.

We now present an example to illustrate these definitions as well as to show that having the maximum number of copies of an item need not always help in terms of popularity.

2.1. Illustrative example

Consider the following instance $G = (\mathcal{A} \cup \mathcal{B}, E)$, where $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$ and $\mathcal{B} = \{f_1, f_2, s_1, s_2, s_3, s_4\}$ and the preference lists of people are described in Fig. 1(a). In Fig. 1(b) we have the same set of applicants and preference lists except that every item has 2 copies. That is, $\mathcal{B}' = \{f_1, f_1', f_2, f_2', s_1, s_1', s_2, s_2', s_3, s_3', s_4, s_4'\}$, where we have explicitly included an identical copy b' of b, for every $b \in \mathcal{B}$. We refer to this instance with duplicates as the instance $H = (\mathcal{A} \cup \mathcal{B}', E')$. Here and in the rest of the paper, the first column in the figure denotes the set of people and the row adjoining it denotes the preference list of the particular person. For example, a_1 treats f_1 as its rank-1 item, f_2 as its rank-2 item and s_1 as its rank-3 item according to Fig. 1(a). When there are multiple items in the same cell as in case of Fig. 1(b), we say that the items are tied at that rank. That is, a_1 treats both f_1 and f_1' as its rank-1 item according to Fig. 1(b). We omit the numbering of the columns as given by the first row from the rest of the figures.

	1	2	3
a_1	f_1	f_2	s_1
a_2	f_1	f_2	s_2
a_3	f_1	f_2	s_3
a_4	f_1	f_2	s_4
a_5	f_2		

(a)

	1	2	3
a_1	f_1, f_1'	f_2, f_2'	s_1, s_1'
a_2	f_1, f_1'	f_2, f_2'	s_2, s_2'
a_3	f_1, f_1'	f_2, f_2'	s_3, s_3'
a_4	f_1, f_1'	f_2, f_2'	s_4, s_4'
a_5	f_2, f_2'		

(b)

Fig. 1. Example showing larger copies do not always help.

Consider the subgraph $G_1 = (\mathcal{A} \cup \mathcal{B}, E_1)$ of G where every person has edges only to her top-choice item. It is easy to see that every maximum matching of G_1 has to match the following vertices: the items $\{f_1, f_2\}$ and the person a_5. Thus, the items f_1 and f_2 are critical and $s_1, \ldots, s_4$ are non-critical in G_1. Hence s_i is the most preferred non-critical vertex in G_1 for person a_i, for $i = 1, \ldots, 4$. Applying Theorem 1 we see that the instance G admits a popular matching: for example, $\{(a_1, f_1), (a_5, f_2), (a_2, s_2), (a_3, s_3), (a_4, s_4)\}$ is a popular matching.

Let us now consider the subgraph H_1 of H where every person has edges to her rank-1 items. The critical vertices in $H_1 = (\mathcal{A} \cup \mathcal{B}', E_1')$ are f_1 and f_1' while f_2 and f_2' are now *non-critical* in H_1. Thus f_2 and f_2' become the most preferred non-critical vertices for $a_1, \ldots, a_4$ — hence any popular matching has to match each of $a_1, \ldots, a_4$ to one of $\{f_1, f_1', f_2, f_2'\}$. Also, a_5 has to be matched to f_2 or f_2' (since a popular matching is a maximum matching on rank-1 edges). Since there are 5 people and only 4 items that they can be matched to in any popular matching, there exists no popular matching now. Thus the instance shown in Fig. 1(b) where each item has 2 copies does not admit a popular matching while the instance in Fig. 1(a) where each item has a single copy does.

3. The 1-or-2 copies problem

Given a graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ and a subset $K \subseteq \mathcal{B}$ of items which can be duplicated, that is, we can have up to 2 copies of every item in the set K, the problem is to determine if there exists a setting of copies of items as $\langle x_1, \ldots, x_{|\mathcal{B}|}\rangle$ where $x_i = 1$ for each item $b_i \in \mathcal{B} \setminus K$ and x_i is either 1 or 2 for each item $b_i \in K$ such that with these copies the resulting graph admits a popular matching.

To prove that this problem is NP-hard, we reduce the problem of monotone 1-in-3 SAT to the 1-or-2 copies problem. Monotone 1-in-3 SAT is a variant of the 3-satisfiability problem (3SAT). Like 3SAT, the input instance is a collection of clauses, where each clause consists of exactly three variables and no variable appears in negated form. The monotone 1-in-3 SAT problem is to determine whether there exists a truth assignment to the variables so that each clause has exactly one true variable (and thus exactly two false variables). This problem is NP-hard [17]; in fact the variant of monotone 1-in-3 SAT where each variable occurs in at most 3 clauses is also NP-hard (refer to [4]). As with other reductions from the 3SAT problem, our reduction also involves designing small *gadgets* which build an instance of the 1-or-2 copies problem. We first give an overview of our reduction and then describe these gadgets in detail.

3.1. Overview of the reduction

Let I be an instance of the monotone 1-in-3 SAT problem with $\{X_1, X_2, \ldots, X_n\}$ being the set of variables and $\{C_1, C_2, \ldots, C_m\}$ being the set of clauses in I. We must construct from I an instance of the 1-or-2 copies problem $G = (\mathcal{A} \cup \mathcal{B}, E)$ and a subset $K \subseteq \mathcal{B}$ of items which can be duplicated.

We want the following properties of our input $\langle G, K\rangle$:

(i) G (with a single copy of each item) does not admit a popular matching.
(ii) By duplicating some items in K we get an instance that admits a popular matching *iff* there exists an assignment with exactly one variable in each clause of I set to true.

With these observations we design a gadget for every clause in I. Each gadget consists of a set of people and a set of items along with the preference lists of people. A subset of these items is *internal* to the gadget, that is, such items appear only on the preference lists of people within the gadget. In addition there will be items which are *public*, that is, such items appear on the preference lists of people across several gadgets. The set of all public items in our instance G is the set K of items which may be duplicated, that is, the items in K that get duplicated will have 2 copies in the resulting instance.

Note that the instance I requires us to decide the true/false status of variables $\{X_1, \ldots, X_n\}$. Similarly the instance G requires us to decide the duplication status for items in K. The non-triviality of the 1-or-2 copies problem lies in the following: Let b be an item that is a unique rank-1 item for exactly one person in G, then b is critical in G_1 (the graph where we have only rank-1 edges). Making an extra copy of b' of item b makes both b and b' non-critical in the resulting graph restricted to rank-1 edges. This in turn may change the s-items of people in the resulting graph, and hence the resulting graph may admit a popular matching. The preference lists of people in our gadgets therefore ensure that every item in K is a unique rank-1 item for exactly one person in G.

Public items. The set of public items in our instance G is the set K of items which may be duplicated. We now describe how we derive the set of public items from the instance I. For every occurrence of variable X_i in I, we have a public item in G. That is, if a variable X_i appears in clause C_t we have an item u_i^t in G. Note that since I is an instance of monotone 1-in-3 SAT, no variable appears in negated form in I.

We denote by $dup(b)$ the duplication status of item $b \in K$: $dup(b) = 0$ implies that item b is not duplicated and hence has only 1 copy in the resulting instance, while $dup(b) = 1$ implies that b gets duplicated and hence has 2 copies in the resulting instance. The value of $dup(u_i^t)$'s should capture the truth value of variable X_i appearing in clause C_t, that is, assign variable X_i in clause C_t as *false* if $dup(u_i^t) = 0$, whereas assign variable X_i in clause C_t as *true* if $dup(u_i^t) = 1$.

For us to make the above translation of duplication status, it has to be the case that for any i, all $dup(u_i^t)$'s have the same value. So if some $dup(u_i^t)$ is set to 1, we will need to set $dup(u_i^\ell) = 1$, for all ℓ where $u_i^\ell \in K$. Thus the set of items corresponding to all occurrences of variable X_i should simultaneously have the same duplication status. That is, we want the following property with respect to the duplication status of items corresponding to X_i.

$(*)$ Let $u_i^t \in K$, then $dup(u_i^t) = dup(u_i^{t'})$, for all t' where $u_i^{t'} \in K$.

It can be seen that if this property is satisfied, then the duplication status of item u_i^t can be translated to the truth assignment for X_i appearing in clause C_t. Further, this will be a consistent assignment, that is, all occurrences of variable X_i get the same value.

With this, we have completely described all the public items (elements of K) in our instance G. The set K, therefore, consists of $3m$ items as shown below.

$$K = \cup_{i,t}\{u_i^t : X_i \text{ appears in } C_t\}. \tag{1}$$

We now describe our gadgets - one corresponding to each clause and show how all the gadgets ensure that the above constraint is met.

3.2. Gadget corresponding to a clause

Let $C_t = (X_{i_1} \vee X_{i_2} \vee X_{i_3})$ be a clause in I. Corresponding to C_t we have a gadget which we denote by G_t. The gadget G_t consists of a set $A_t = \{a_1^t, \ldots, a_{14}^t\}$ of 14 people and a set $B_t = \{p_1^t, p_2^t, p_3^t, q_0^t, q_1^t, q_2^t\}$ of 6 internal items. Fig. 2(a)–(c) show the preference lists of the 14 people $a_1^t, \ldots, a_{14}^t$ associated with the clause C_t. Recall that we introduce a last resort item for each person to ensure that matchings are always $\mathcal{A}$-complete. The ℓ-items are these last resort items.

As seen in Fig. 2(a) and (c) the public items $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ appear on the preference lists of people shown in these two tables. Apart from these public items, people shown in Fig. 2(b) have public items $u_{i_1}^{t_1}, u_{i_2}^{t_2}, u_{i_3}^{t_3}$ appearing on their preference lists. Here, t_1, t_2, t_3 (each of them $\neq t$) is such that, $X_{i_1}, X_{i_2}, X_{i_3}$ appear in $C_{t_1}, C_{t_2}, C_{t_3}$ respectively. The role of the applicants $a_6^t, \ldots, a_{11}^t$ is to ensure that the property (*) mentioned above is satisfied. It is easy to see that if some variable X_{i_1} appears in exactly one clause (say C_t) in the instance I, then we have exactly one public item $u_{i_1}^t$ in our set K corresponding to variable X_{i_1} and the property (*) is trivially true with respect to X_{i_1}. Hence assume that the variable X_{i_1} appears in at least 2 clauses, then t_1 is such that C_{t_1} denotes the next higher numbered clause after C_t in which variable X_{i_1} appears. If C_t happens to be the highest numbered clause in which variable X_{i_1} appears, then let C_{t_1} denote the lowest numbered clause in which variable X_{i_1} appears. Similarly let t_2 (resp. t_3) denote the next higher numbered clause after C_t in which variable X_{i_2} (resp. X_{i_3}) appears. We note that t_1, t_2, t_3 need not be all distinct.

a_1^t	p_1^t	$u_{i_1}^t, q_0^t, q_1^t$	$\ell_{t,1}$
a_2^t	p_1^t	$u_{i_2}^t, q_0^t, q_1^t$	$\ell_{t,2}$
a_3^t	p_1^t	$u_{i_3}^t, q_0^t, q_1^t$	$\ell_{t,3}$
a_4^t	p_1^t	q_2^t	$\ell_{t,4}$
a_5^t	p_1^t	q_2^t	$\ell_{t,5}$

(a)

a_6^t	p_2^t	$u_{i_1}^t, u_{i_1}^{t_1}$	$\ell_{t,6}$
a_7^t	p_2^t	$u_{i_2}^t, u_{i_2}^{t_2}$	$\ell_{t,7}$
a_8^t	p_2^t	$u_{i_3}^t, u_{i_3}^{t_3}$	$\ell_{t,8}$
a_9^t	p_3^t	$u_{i_1}^t, u_{i_1}^{t_1}$	$\ell_{t,9}$
a_{10}^t	p_3^t	$u_{i_2}^t, u_{i_2}^{t_2}$	$\ell_{t,10}$
a_{11}^t	p_3^t	$u_{i_3}^t, u_{i_3}^{t_3}$	$\ell_{t,11}$

(b)

a_{12}^t	$u_{i_1}^t$	$\ell_{t,12}$
a_{13}^t	$u_{i_2}^t$	$\ell_{t,13}$
a_{14}^t	$u_{i_3}^t$	$\ell_{t,14}$

(c)

Fig. 2. Preference lists of people corresponding to a clause C_t.

The preference lists are designed such that when each public item has a single copy, then G_t does not admit any popular matching. Further, any instance that admits a popular matching and is obtained by duplicating public items in G_t obeys the following two properties:

At least one of $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ must get duplicated. As seen in Fig. 2(a), a_4^t and a_5^t have p_1^t as their top item and q_2^t as their second choice item — as q_2^t is nobody's top choice item, it follows that q_2^t is the most preferred non-critical item of a_4^t and a_5^t. It is easy to see that in any popular matching, one of a_4^t, a_5^t has to be matched to p_1^t and the other to q_2^t. The role of these 2 people is to ensure that a_1^t, a_2^t, a_3^t always get matched to items in $s(a_1^t), s(a_2^t), s(a_3^t)$, respectively.

Items $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ appear as unique top items for a_{12}^t, a_{13}^t and a_{14}^t respectively (Fig. 2(c)). Further, they do not appear as top choice items for any other person in G. Thus, with a single copy, all these items remain *critical* on rank-1 edges. Hence $s(a_1^t) = s(a_2^t) = s(a_3^t) = \{q_0^t, q_1^t\}$. Thus these 3 people cannot be matched to only these 2 items in any popular matching, so there exists no popular matching when each item has a single copy.

Therefore at least one of $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ should have 2 copies in the resulting instance for all of a_1^t, a_2^t, a_3^t to be matched to items in $s(a_1^t), s(a_2^t), s(a_3^t)$, respectively.

Exactly one of $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ can be duplicated. The 6 people $a^t_6, \ldots, a^t_{11}$ ensure that exactly one amongst $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ can have 2 copies in any instance that admits a popular matching. These 6 people as shown in Fig. 2(b) can be divided into two sets – $S_1 = \{a^t_6, a^t_7, a^t_8\}$ and $S_2 = \{a^t_9, a^t_{10}, a^t_{11}\}$. All these people have public items as their second choice items. Further, the preference lists of people in gadgets $G_{t_1}, G_{t_2}, G_{t_3}$ ensure that items $u^{t_1}_{i_1}, u^{t_2}_{i_2}, u^{t_3}_{i_3}$ also appear as unique top items for exactly one person. Hence, when each of the public items has a single copy, then one person from S_1 and one person from S_2 get matched to their top item (p^t_2 and p^t_3 resp.), whereas the rest of the people get matched to their respective last resort items (most preferred non-critical item) in any popular matching. However, we know that at least one of $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ has 2 copies due to $a^t_1, \ldots, a^t_5$.

We will assume here that copies of the items $u^{t_1}_{i_1}, u^{t_2}_{i_2}, u^{t_3}_{i_3}$ are unavailable for the people in the gadget G_t since these copies will be used up by people belonging to the other gadgets. It therefore suffices to focus on the duplication status of items $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ with respect to people in gadget G_t. If exactly one among $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ (say, $u^t_{i_1}$) gets duplicated, then both the copies of the item $u^t_{i_1}$ become non-critical in the resulting graph restricted to rank-1 edges. The person a^t_{12} gets matched to one copy of $u^t_{i_1}$ as its f-item whereas person a^t_1 gets matched to another copy of $u^t_{i_1}$ as its s-item. Further, a^t_6 and a^t_9 get matched to the respective top items (p^t_2 and p^t_3) while a^t_7, a^t_8 and a^t_{10}, a^t_{11} get matched to their respective last resort items. Assume that any two of $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ (say $u^t_{i_1}, u^t_{i_2}$) have 2 copies in the resulting instance. As in the previous case, one copy of each of these items is used up to match people a^t_{12} and a^t_{13} to their respective f-items. Further the extra copy of one of the 2 items (say $u^t_{i_1}$) is matched to people in a^t_1, a^t_2, a^t_3. However, in this case we also have four people a^t_6, a^t_9 and a^t_7, a^t_{10} treating $u^t_{i_1}$ and $u^t_{i_2}$ as their s-items respectively. Although one person from each of the above pairs can be matched to her top item, we cannot match one copy of the item ($u^t_{i_2}$) amongst the remaining two people. It is easy to see that a similar case occurs if all 3 of $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ are duplicated.

Thus the gadget G_t ensures that exactly one of the items $u^t_{i_1}, u^t_{i_2}, u^t_{i_3}$ has 2 copies in the resulting instance. Note that the gadget described above assumes that every variable appears in at least 2 clauses. For example, by this assumption for variable X_{i_1}, we have the 2 items $u^t_{i_1}$ and $u^{t_1}_{i_1}$ appearing in the preference lists of people a^t_6 and a^t_9. If some variable (say X_{i_1}) appears in exactly one clause (say C_t), then the preference lists of people a^t_6 and a^t_9 contain only the item $u^t_{i_1}$ as their second choice item. This change does not affect the above two properties that the gadget G_t ensures.

3.3. Putting it together

The graph G that we construct is the union of gadgets corresponding to each of the clauses. The sets $\mathcal{A}$ and $\mathcal{B}$ can be described as below:

$$\mathcal{A} = \cup_{t=1}^{m} A_t.$$
$$\mathcal{B} = \cup_{t=1}^{m} B_t \cup K.$$

The set K is the set of all the public items as described earlier in Eq. (1). The preference lists of the people are as shown in our gadget. We note that every item $b \in K$ is a unique rank-1 item for exactly one person in $\mathcal{A}$. We now show how all the gadgets co-operate to enforce property (∗) mentioned in Section 3.1.

Lemma 2. *Assume that by having 1 or 2 copies of items in K there exists an instance that admits a popular matching. In such an instance if for some i, t, $dup(u^t_i) = 1$ then $dup(u^{t'}_i) = 1$, for all t' where $u^{t'}_i \in K$.*

Proof. Recall that the item $u^t_i \in K$ because the variable X_i appears in the clause C_t. If C_t is the only clause in which X_i appears, then we are done, otherwise assume that X_i appears in another clause say C_{t_1}. Without loss of generality let X_i be the first variable in every clause that it appears in and t_1 be such that C_t is the next higher numbered clause after C_{t_1} in which X_i appears. We first show that $dup(u^t_i) = 1 \Rightarrow dup(u^{t_1}_i) = 1$. Suppose not.

The item u^t_i appears on the preference lists of following people:

- a^t_1 in gadget G_t.
- a^t_6 and a^t_9 in gadget G_t.
- $a^{t_1}_6$ and $a^{t_1}_9$ in gadget G_{t_1}.

With $dup(u^t_i) = 1$, all these people regard u^t_i as their s-item. The design of our gadgets ensures that no other item $u^t_{i'}$ can have 2 copies in the resulting instance. Hence, in any popular matching in the resulting instance, while a^t_1 gets matched to its s-item u^t_i, each of $a^t_6, a^t_9, a^{t_1}_6, a^{t_1}_9$ has to be matched to its f-items $p^t_2, p^t_3, p^{t_1}_2, p^{t_1}_3$ respectively.

Our assumption that the resulting instance admits a popular matching and that $dup(u^{t_1}_i) = 0$ implies that there exists another item $u^{t_1}_j \in K$ which has 2 copies, or equivalently $dup(u^{t_1}_j) = 1$. With $dup(u^{t_1}_j) = 1$, the following people belonging to the gadget G_{t_1} treat $u^{t_1}_j$ as their s-item.

- either the 3 people $a^{t_1}_2, a^{t_1}_7$, and $a^{t_1}_{10}$ or the 3 people $a^{t_1}_3, a^{t_1}_8$, and $a^{t_1}_{11}$.

For $k = 2$ or 3, while $a_k^{t_1}$ gets matched to her s-item (which is some $u_j^{t_1}$) in any popular matching, both $a_{k+5}^{t_1}$ and $a_{k+8}^{t_1}$ are left without an item amongst their f or s-items, a contradiction to the lemma hypothesis that the resulting instance admits a popular matching. Hence we have $dup(u_i^{t_1}) = 1$.

Recall that we started with an instance of monotone 1-in-3 SAT where every variable appears in at most 3 clauses. Thus corresponding to the variable X_i, we have at most one more item $u_i^{t_1'} \in K$ and we need to show that $dup(u_i^{t_1'}) = 1$. We note that a similar argument as above forbids any instance where $dup(u_i^t) = dup(u_i^{t_1}) = 1$ and $dup(u_i^{t_1'}) = 0$, to admit a popular matching. Hence we have $dup(u_i^{t_1'}) = 1$.

This completes our proof. □

Thus if by having 1 or 2 copies of each item in K there exists an instance that admits a popular matching, then the duplication status of items in K always translates to a consistent truth assignment of the variables in I. We now prove the following two lemmas which establish the correctness of our reduction.

Lemma 3. *If there exists an instance by having 1 or 2 copies of items in K such that this instance admits a popular matching, then there exists a 1-in-3 satisfying assignment for I.*

Proof. We obtain a truth assignment of variables $X_1, \ldots, X_n$ of I from the duplication status of items in K as follows: for X_i appearing in clause C_t, set X_i to *true* if $dup(u_i^t) = 1$, else set X_i to *false*. By Lemma 2, this assignment is a consistent assignment. To see that this is also a 1-in-3 satisfying assignment, consider a clause $C_t = (X_{i_1} \vee X_{i_2} \vee X_{i_3})$ in I. The fact that the resulting instance admits a popular matching enforces that in G_t, exactly one of $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ has its $dup(\cdot)$ value set to 1. As the above property is true for every clause C_t, it follows that we have a truth assignment to the variables of I such that every clause has exactly one variable set to true. This proves the lemma. □

Lemma 4. *If there exists a 1-in-3 satisfying assignment for I, then there exists an instance obtained by duplicating items in K which admits a popular matching.*

Proof. Let Truth-val denote a 1-in-3 satisfying truth assignment to variables $X_1, \ldots, X_n$ of I. Set the $dup(\cdot)$ value of each item in $K = \cup_{i,t}\{u_i^t : X_i \text{ appears in } C_t\}$ as follows:

$$dup(u_i^t) = 0 \quad \text{if Truth-val}(X_i) = \mathit{false}, \; \text{else } dup(u_i^t) = 1 \quad \forall i, t \text{ where } u_i^t \in K.$$

First, since Truth-val is a 1-in-3 satisfying assignment for I, every clause in I has exactly one variable set to true. This implies that for every clause $C_t = (X_{i_1} \vee X_{i_2} \vee X_{i_3})$, in the gadget G_t, exactly one item among $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$ has duplication status set to 1. Let us assume that Truth-val sets variable X_{i_1} in clause C_t to true. Thus the item $u_{i_1}^t$ has 2 copies in our instance.

To prove that the graph as defined by the above duplication status admits a popular matching, we show that there exists a matching that is a maximum cardinality matching on rank-1 edges and where each person $a \in \mathcal{A}$ gets matched to an item in $f(a) \cup s(a)$. We focus on the people belonging to the set A_t.

- Consider the 5 people $a_1^t, \ldots, a_5^t$ first: a_4^t and a_5^t get matched to p_1^t and q_2^t, respectively whereas a_2^t and a_3^t get matched to q_0^t and q_1^t, respectively (their s-items). Since we have 2 copies of item $u_{i_1}^t$, both these copies are non-critical in the graph on rank-1 edges. Thus, a_1^t can be matched to one copy of $u_{i_1}^t$ since it is one of her s-items.
- Consider the 6 people $a_6^t, \ldots, a_{11}^t$ next: we recall that the duplication scheme was derived from the 1-in-3 satisfying assignment for I and the property (*) mentioned in Section 3.1 is ensured by our gadgets. Thus, our assumption that item u_{i_1} has 2 copies in our instance implies that each of the items $u_{i_2}^t, u_{i_3}^t, u_{i_2}^{t_2}, u_{i_3}^{t_3}$ have a single copy in our instance. Thus all these items remain critical on the graph restricted to rank-1 edges and hence each of the people $a_7^t, a_8^t, a_{10}^t, a_{11}^t$ treat their respective last resort items as their s-items. We can therefore match a_6^t and a_9^t to their f-items p_2^t and p_3^t respectively while matching $a_7^t, a_8^t, a_{10}^t, a_{11}^t$ to their unique last-resorts.
- This leaves us with the 3 people $a_{12}^t, a_{13}^t, a_{14}^t$ who get matched to their respective f-items $u_{i_1}^t, u_{i_2}^t, u_{i_3}^t$.

This finishes our proof that if Truth-val is a 1-in-3 satisfying assignment for I, then we have a setting of duplication values of items in K such that the resulting graph admits a popular matching. □

We note that all the people in our gadgets have preference lists of length 2 where ties occur at second choice items (see Fig. 2). Recall that the last-resort items were introduced by us and hence are not counted. Further, all people except a_1^t, a_2^t, a_3^t have at most 2 items tied as their second choice items. It is easy to see that we can merge the items q_0^t, q_1^t into a single item q^t which now belongs to the set of items which can be duplicated. Further, in any instance that admits a popular matching, we will set $dup(q^t) = 1, \forall t = 1, \ldots, m$. We can now conclude the following theorem.

Theorem 2. *The 1-or-2 copies problem is NP-hard for preference lists of length 2 with ties of length 2 allowed in second choice items.*

Since the ties in the preference lists occur only for second choice items, this leaves us with two unresolved cases:

- The complexity of the 1-or-2 copies problem when preference lists are *strict* (that is, no ties are allowed). Our gadgets can be easily modified to show that this problem is also NP-hard.
- The complexity of the 1-or-2 copies problem when preference lists have length two and ties can occur only at the first position. We show that this case has a polynomial time algorithm.

3.4. Strict lists

We break ties in the above instance as follows:

- For each of a_k^t, $k \in \{1, 2, 3\}$, we first replace the items q_0^t and q_1^t by a single item q^t which becomes an element of K (set of those items that can be duplicated). Further we let item $u_{i_k}^t$ precede the item q^t in the preference list of a_k^t.
- For any of a_k^t, $k \in \{6, \dots, 11\}$, we have two public items tied as a_k^t's second choice, call them u_i^t and $u_i^{t_1}$. Recall that these two public items correspond to the same variable X_i appearing in two different clauses namely C_t and C_{t_1}. We break the tie in the rank-2 item for a_k^t by letting item u_i^t precede item $u_i^{t_1}$ such that $t < t_1$. That is, the item corresponding to the lower numbered clause precedes the one corresponding to the higher numbered clause.

It is easy to check that for an $\mathcal{A}$-complete matching to exist, the items q^t for $t = 1, \dots, m$ need to have 2 copies in the resulting instance. It is straightforward to verify that all our claims hold even with these strict preference lists and hence the 1-or-2-copies problem is NP-hard for strict preference lists.

Corollary 1. *The 1-or-2 copies problem for strict preference lists where the longest preference list has length 3 is NP-hard.*

3.5. Master preference list

In this section we consider the restriction of the problem when preference lists of the people are derived from a *master* preference list. A master list is a ranking of all items according to some global objective criterion. The master list may be allowed to contain ties or may be strict. Irving et al. [9] considered the stable marriage problem in the presence of master preference lists and proved that many interesting variants remain hard under this master list model. In the same vein, we consider the 1-or-2 copies problem in the presence of a master list.

In the master list model, the preference list of a person is the same as the master list, except that she can delete all items that she finds unacceptable. We show that under this severe restriction of master list also, the 1-or-2 copies problem is NP-hard for strict preferences.

For the sake of convenience, we partition the set $\mathcal{B}$ as $\mathcal{F}$ (f-items), $\mathcal{S}$ (s-items) and $\mathcal{D}$ (duplicable items).

$$\mathcal{B} = \mathcal{F} \cup \mathcal{D} \cup \mathcal{S}$$

where the sets $\mathcal{F}$, $\mathcal{D}$ and $\mathcal{S}$ are as defined below:

$$\mathcal{F} = \cup_{t=1}^{m} \{p_1^t, p_2^t, p_3^t\}$$
$$\mathcal{S} = \cup_{t=1}^{m} \{q_2^t\}$$
$$\mathcal{D} = K \cup_{t=1}^{m} \{q^t\}.$$

Here the set K is as defined by Eq. (1). The item q^t is the replacement for items q_0^t and q_1^t as done for strict lists. We note that the set $\mathcal{D}$ is the set of items which can be duplicated. To get a master list such that the preference lists of all people are derived from the master list, we order elements of $\mathcal{F}$, $\mathcal{S}$ and $\mathcal{D}$ as follows:

- Order the items in $\mathcal{F}$ in any arbitrary order in a strict manner. Let $\mathcal{F}_o$ denote the ordered list.
- Order the items in $\mathcal{S}$ in any arbitrary order in a strict manner. Let $\mathcal{S}_o$ denote the ordered list.
- Let $\mathcal{D}_o$ denote the ordered list of $\mathcal{D}$ as:
  $\mathcal{D}_o = u_{i_1}^{t_1}, u_{i_1}^{t_2}, u_{i_1}^{t_3}, \dots, u_{i_n}^{l_1}, u_{i_n}^{l_2}, u_{i_n}^{l_3}, q^1, q^2, \dots, q^m$
  Here $t_1 < t_2 < t_3$ and $l_1 < l_2 < l_3$.

It is clear that the list $\mathcal{F}_o, \mathcal{D}_o, \mathcal{S}_o$ forms a master list for the strict instance G constructed above. We therefore conclude the following corollary.

Corollary 2. *The 1-or-2 copies problem with strict preference lists derived from a master list is NP-hard.*

3.6. Lists of length 2 with ties in the top position only

Here we consider the 1-or-2 copies problem for preference lists of length 2 where the second choice item is unique. We are given $G = (\mathcal{A} \cup \mathcal{B}, E)$ and every $a \in \mathcal{A}$ can give any number of top choice items, say $f_0, f_1, \dots, f_d$, which are tied as a's most preferred items and at most a single item, say s, as a's next choice item. Thus the entire preference list of a is: $(f_0, f_1, \dots, f_d)$ followed by s followed by ℓ_a. We show there exists a polynomial time algorithm for such instances.

Let $\mathcal{E}$, $\mathcal{O}$ and $\mathcal{U}$ denote the set of *even*, *odd* and *unreachable* vertices respectively, in G_1 (the graph G restricted to rank-1 edges), refer to Section 2. The set $\mathcal{E}$ is the set of non-critical vertices, thus only elements of $\mathcal{E}$ are candidates for being elements in $s(a)$, for any $a \in \mathcal{A}$. Recall that certain items that are critical in G_1 can become non-critical on rank-1 edges once they get duplicated.

Let $G = (\mathcal{A} \cup \mathcal{B}, E)$ be the instance of the above special case and K be the set of items that can be duplicated. Assume that G does not admit a popular matching and there exists a way of duplicating items in K such that the augmented instance $\tilde{G} = (\mathcal{A} \cup \tilde{\mathcal{B}}, E)$ admits a popular matching. Then we have the following lemma.

Lemma 5. *There exists an augmented instance H such that every item that is critical in G_1 is also critical in H_1 and H admits a popular matching.*

Proof. If every item that is critical in G_1 is also critical in $\tilde{G}_1$ (recall that $\tilde{G}_1$ is the graph $\tilde{G}$ restricted to rank-1 edges), then we have $H = \tilde{G}$ and we are done. Suppose not, then there exists an item b_1 such that b_1 is critical in G_1 but with $dup(b_1) = 1$, both b_1 and its duplicate b'_1 are non-critical in $\tilde{G}_1$. Since b'_1 is non-critical in $\tilde{G}_1$, deleting b'_1 does not change the size of the maximum cardinality matching in $\tilde{G}_1$. As any popular matching has to be a maximum cardinality matching in $\tilde{G}_1$, the contribution of b'_1 is to add an edge (a, b'_1) of rank-2 in any popular matching of $\tilde{G}$. Note that the person a has to be non-critical in $\tilde{G}_1$, as all critical vertices of $\tilde{G}_1$ are matched along rank-1 edges in any popular matching of $\tilde{G}$.

The deletion of b'_1 makes b_1 critical in the resulting graph. Since a is non-critical, all of a's top choice items have to be critical as there is no edge between 2 non-critical vertices (Lemma 1, see Section 2) - thus a treats its unique last-resort item ℓ_a as its s-item once b'_1 is deleted. In the resulting graph too, a has a partner to match in the set $f(a) \cup s(a)$.

Deleting all such duplicates b' from $\tilde{G}$ we get the desired graph H in which every critical item in G_1 is also critical in H_1. Further, since every person a still has a partner in $f(a) \cup s(a)$, it follows that H admits a popular matching. □

The above lemma makes it simple for us to decide which items in K should be duplicated - we maintain the invariant that no critical item in G_1 becomes non-critical due to duplication. Let $\mathcal{O}_{\mathcal{B}}$, $\mathcal{E}_{\mathcal{B}}$ and $\mathcal{U}_{\mathcal{B}}$ denote the *odd*, *even* and *unreachable* items respectively, in the current graph. Initially, the current graph is G_1.

- No item $b \in \mathcal{U}_{\mathcal{B}} \cap K$ should be duplicated. This is because making an extra copy b' of such an item b makes both b and b' non-critical in the resulting graph restricted to rank-1 edges.
- We duplicate all items $b \in \mathcal{E}_{\mathcal{B}} \cap K$. This is because making an extra copy b' of such an item b does not change the criticality status of b as well as b' in the resulting graph restricted to rank-1 edges.
- For $b \in \mathcal{O}_{\mathcal{B}}$ we note the following: There exists an alternating path starting from an unmatched person a to an item $b \in \mathcal{O}_{\mathcal{B}}$. Duplicating such an item b creates an augmenting path from a to b; thus no critical item on rank-1 edges turns non-critical by this change. However an item $b \in \mathcal{O}_{\mathcal{B}}$ can now belong to $\mathcal{U}_{\mathcal{B}}$ due to the above change. Hence we need to update the status of all items $b \in \mathcal{O}_{\mathcal{B}}$ to check whether b belongs to $\mathcal{O}_{\mathcal{B}}$ or $\mathcal{U}_{\mathcal{B}}$. For instance, let (f_0, f_1) be the top choice for 3 people a_1, a_2, a_3. Both f_0 and f_1 are *odd*, however once f'_0 is introduced, we will have 3 people and 3 top items (f_0, f'_0, f_1), thus all these items are now unreachable on rank-1 edges. Hence we should not duplicate f_1 now.

Our algorithm is presented in Fig. 3. The correctness of the algorithm follows from Lemma 5.

1. Let G_1 denote the graph G on rank 1 edges; that is, $G_1 = (\mathcal{A} \cup \mathcal{B}, E_1)$.
2. Partition vertices in $\mathcal{B}$ w.r.t. a maximum cardinality matching M_1 in G_1 as *even*($\mathcal{E}_{\mathcal{B}}$), *unreachable*($\mathcal{U}_{\mathcal{B}}$), *odd*($\mathcal{O}_{\mathcal{B}}$).
3. $H_0 = G$.
4. For every $b \in \mathcal{E}_{\mathcal{B}} \cap K$ do
   - Add an extra copy of b to the graph H_0.
5. Order the items in $\mathcal{O}_{\mathcal{B}} \cap K$ as $o_1, \ldots, o_t$.
   For each $i = 1$ to t do:
   - If o_i is *odd* on rank 1 edges in H_{i-1}, then set $dup(o_i) = 1$.
     This defines the graph H_i, that is, $H_i = H_{i-1}+$ an extra copy of o_i if $dup(o_i) = 1$.
   - Else $H_i = H_{i-1}$.
6. If H_t admits an $\mathcal{A}$-complete matching, then return H_t as the graph $\tilde{G}$.
   Otherwise output "there is no $\tilde{G}$ corresponding to $\langle G, K\rangle$ that admits a popular matching".

Fig. 3. Algorithm for a special case of 1-or-2 copies problem.

Theorem 3. *There exists a polynomial time algorithm for the 1-or-2 copies problem with preferences lists of length 2 with ties occurring at the top position only.*

4. Bounded total copies problem

We now consider the following problem: suppose we are given an integer k and we have to decide whether there exist $\Delta_1, \ldots, \Delta_{|\mathcal{B}|}$ where each $\Delta_i \geq 1$ and $\sum_i \Delta_i \leq k$, such that having $\Delta_i + 1$ copies of the i-th item, for $1 \leq i \leq |\mathcal{B}|$, enables the resulting instance to admit a popular matching. We show that this problem can be solved in polynomial time.

Let G_1 denote the graph in which every person adds edges to her f-items. Every *even* person in G_1 adds an edge to her s-item. We call this graph G', that is where every person a has added edges to items in $f(a)$ and every *even* person a in G_1 has added edges to items in $s(a)$. Let M be any maximum cardinality matching in G'. Since G does not admit a popular matching, we know that $|M| < |\mathcal{A}|$. The following theorem is useful here.

Theorem 4. *Let $G = (\mathcal{A} \cup \mathcal{B}, E)$ be the given graph and let integer k denote an upper bound on the total extra copies of all items that we can have in the augmented instance. Then there exists a new instance $H = (\mathcal{A} \cup \tilde{\mathcal{B}}, E)$ with extra copies of items that admits a popular matching iff $k \geq |\mathcal{A}| - |M|$, where M denotes any maximum cardinality matching in G'.*

Proof. We first show that k has to be at least $|\mathcal{A}| - |M|$. We need to show that adding an extra copy of any item increases the size of the maximum cardinality matching in the resulting graph restricted to f and s-items by at most 1. This is easy to see since adding an extra copy of an s-item b does not change the f/s-status of the item b and its duplicate. Thus if b was *critical* in the graph restricted to f-items and s-items, then having a duplicate b' increases the size of the maximum cardinality matching in this graph by at most 1. Regarding f-items, note that this item is *critical* on rank-1 edges before its duplicate was introduced. Hence the duplicate introduced might make the copy of the item an s-item, however the contribution to the size of the maximum matching in the graph restricted to f-items and s-items is at most 1.

Since each extra copy of some item increases the size of a maximum matching by at most one, we need to have at least $|\mathcal{A}| - |M|$ extra copies of some items so that the resulting graph admits a popular matching.

To show the other side of the implication, let $k \geq |\mathcal{A}| - |M|$. We will show that it is possible to construct a new instance H such that H' (the graph H restricted to f-items and s-items) admits an $\mathcal{A}$-complete matching.

Since G does not admit a popular matching, there exists an $\alpha \in \mathcal{A}$ such that α is unmatched in G'; note that such an α is *even* or *non-critical* in G'. Further, the f-item b for such a person is *odd* in G'. Let H_1 be the same as the graph G, except that we add an extra copy b' of b and we match α to b'. Note that none of the sets $s(a)$ for $a \in \mathcal{A}$ has changed in H_1 as no critical vertex turns non-critical after adding the duplicate b'. This is because the item b was *odd* before the duplication; so either b and b' are still *odd* or they become *unreachable* now. Thus each person matched in M continues to remain matched in any maximum cardinality matching in H'_1. Further, we have an extra person matched to b'. It is easy to see that the same step can be repeated, until we have an $\mathcal{A}$-complete matching in some H'_i. This process is guaranteed to halt after $|A| - |M|$ iterations and the resulting graph $H_{|A|-|M|}$ is indeed our desired graph H that admits a popular matching. □

The algorithm to construct such a graph H from the given instance $G(\mathcal{A} \cup \mathcal{B}, E)$ follows from the proof of sufficiency of the above theorem. The algorithm is described in Fig. 4.

1. Construct the graph $G' = (\mathcal{A} \cup \mathcal{B}, E')$ where $E' = \{(a, b) : a \in \mathcal{A}, b \in f(a) \cup s(a)\}$.
2. $H_0 = G, H'_0 = G'$, Let M denote a maximum cardinality matching in H'_0.
3. $i = 0$.
4. While M is not $\mathcal{A}$-complete matching do:
   - Let a be any *unmatched* person in H'_i.
   - Let $b = f(a)$ [such a b is *odd* in H'_i since a is *even* in H'_i].
   - Add an extra copy of b and call the new instance H_{i+1}.
   - Construct the graph H'_{i+1} corresponding to H_{i+1} and update M to be a maximum cardinality matching in H'_{i+1}.
   - $i = i + 1$.
5. Output the graph H_i.

Fig. 4. Algorithm for bounded total copies problem.

The above while loop runs for $|\mathcal{A}| - |M|$ iterations, each time adding one more copy of some item. Note that the same item might get chosen in various iterations and thus individual copies are not necessarily bounded. For example, if the input had n people with identical preference lists: top item is b_1, followed by b_2, and so on, then our algorithm would add $n - 1$ copies of item b_1. It is also easy to see that M is a popular matching in the final graph.

5. Summary

Given a bipartite graph $G = (\mathcal{A} \cup \mathcal{B}, E)$ of people and items where people have preferences over items, we showed that the problem of deciding if there exists an $(x_1, \ldots, x_{|\mathcal{B}|}) \in \{1, 2\}^{|\mathcal{B}|}$ such that having x_i copies of the i-th items enables the resulting graph to have a popular matching is NP-hard. We reduced the monotone 1-in-3 SAT problem to an instance of the above problem.

Our reductions constructed instances where the maximum length of a preference list is 2 when preference lists can have ties and the maximum length of a preference list is 3 when preference lists are strict. We showed that the problem is solvable in polynomial time when preference lists have length at most 2 with a unique second choice item. Also, the variant of this problem where the total number of extra copies is bounded rather than a bound on the number of copies of individual items, is solvable in polynomial time.

Acknowledgements

We thank Sourav Chakraborty for discussions that motivated this work and Prajakta Nimbhorkar for helpful discussions. We also thank the anonymous reviewers for their helpful comments.

Appendix

The popular matchings algorithm

Below is the algorithm from [3] to compute a popular matching in $G = (\mathcal{A} \cup \mathcal{B}, E)$.

- Construct the graph $G' = (\mathcal{A} \cup \mathcal{B}, E')$, where $E' = \{(a, b) : a \in \mathcal{A}$ and $b \in f(a) \cup s(a)\}$.
- Construct a maximum matching M of $G_1 = (\mathcal{A} \cup \mathcal{B}, E_1)$.
- Remove any edge in G' between a vertex in $\mathcal{O}$ and a vertex in $\mathcal{O} \cup \mathcal{U}$.
  [No maximum matching of G_1 contains such an edge.]
- Augment M in G' until it is a maximum matching of G'.
- Return M if it is $\mathcal{A}$-complete, otherwise return *"no popular matching"*.

References

[1] A. Abdulkadiroğlu, T. Sönmez, Random serial dictatorship and the core from random endowments in house allocation problems, Econometrica 66 (3) (1998) 689–701.
[2] D.J. Abraham, K. Cechlárová, D.F. Manlove, K. Mehlhorn, Pareto-optimality in house allocation problems, in: Proceedings of 15th Annual International Symposium on Algorithms and Computation, 2004, pp. 3–15.
[3] D.J. Abraham, R.W. Irving, T. Kavitha, K. Mehlhorn, Popular matchings, SIAM Journal on Computing 37 (4) (2007) 1030–1045.
[4] R. Denman, S. Foster, Using clausal graphs to determine the computational complexity of k-bounded positive one-in-three sat, Discrete Applied Mathematics 157 (7) (2009) 1655–1659.
[5] P. Gärdenfors, Match making: assignments based on bilateral preferences, Behavioural Sciences 20 (1975) 166–173.
[6] R.L. Graham, M. Grotschel, L. Lovasz, Chapter 3, matchings and extensions, 1995, pp. 179–232.
[7] C.-C. Haung, T. Kavitha, D. Michail, M. Nasre, Bounded unpopularity matching, in: Proceedings of the 11th Scandinavian Workshop on Algorithm Theory, 2008, pp. 127–137.
[8] R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, K. Paluch, Rank-maximal matchings, ACM Transactions on Algorithms 2 (4) (2006) 602–610.
[9] R.W. Irving, D.F. Manlove, S. Scott, The stable marriage problem with master preference lists, Discrete Applied Mathematics 156 (15) (2008) 2959–2977.
[10] T. Kavitha, J. Mestre, M. Nasre, Popular mixed matchings, in: Proceedings of the 36th International Colloquium on Automata, Languages and Programming, 2009, pp. 574–584.
[11] T. Kavitha, M. Nasre, Popular matchings with variable job capacities, in: Proceedings of 20th Annual International Symposium on Algorithms and Computation, 2009, pp. 423–433.
[12] M. Mahdian, Random popular matchings, in: Proceedings of the 8th ACM Conference on Electronic Commerce, 2006, pp. 238–242.
[13] D. Manlove, C. Sng, Popular matchings in the capacitated house allocation problem, in: Proceedings of the 14th Annual European Symposium on Algorithms, 2006, pp. 492–503.
[14] R.M. McCutchen, The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences, in: Proceedings of the 15th Latin American Symposium on Theoretical Informatics, 2008, pp. 593–604.
[15] J. Mestre, Weighted popular matchings, in: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, 2006, pp. 715–726.
[16] A.E. Roth, A. Postlewaite, Weak versus strong domination in a market with indivisible goods, Journal of Mathematical Economics 4 (1977) 131–137.
[17] T. Schaefer, The complexity of satisfiability problems, in: Proceedings of the 10th Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226.