Popular Matchings with Multiple Partners*

Florian Brandl[1] and Telikepalli Kavitha[2]

1 **Technische Universität München, Germany**
brandlfl@in.tum.de
2 **Tata Institute of Fundamental Research, India**
kavitha@tcs.tifr.res.in

Abstract

Our input is a bipartite graph $G = (A \cup B, E)$ where each vertex in $A \cup B$ has a preference list strictly ranking its neighbors. The vertices in A and in B are called *students* and *courses*, respectively. Each student a seeks to be matched to $\mathsf{cap}(a) \geq 1$ many courses while each course b seeks $\mathsf{cap}(b) \geq 1$ many students to be matched to it. The Gale-Shapley algorithm computes a pairwise-stable matching (one with no blocking edge) in G in linear time. We consider the problem of computing a *popular* matching in G – a matching M is popular if M cannot lose an election to any matching where vertices cast votes for one matching versus another. Our main contribution is to show that a max-size popular matching in G can be computed by the *2-level Gale-Shapley* algorithm in linear time. This is an extension of the classical Gale-Shapley algorithm and we prove its correctness via linear programming.

1998 ACM Subject Classification G.2.2 Graph algorithms, G.1.6 Linear programming

Keywords and phrases Bipartite graphs, Linear programming duality, Gale-Shapley algorithm

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2017.19

1 Introduction

We study the many-to-many matching problem in bipartite graphs: formally, this is given by a set A of vertices (these vertices are called *students*) and a set B of vertices (these are called *courses*), where every vertex u has a capacity $\mathsf{cap}(u) \geq 1$. Every student a seeks to be matched to $\mathsf{cap}(a)$ many courses and every course b seeks $\mathsf{cap}(b)$ many students to be matched to it. Moreover, every student $a \in A$ has a strict ranking $\succ_a$ over courses that are acceptable to a and every course b has a strict ranking $\succ_b$ over students that are acceptable to b. The set of mutually acceptable pairs is given by $E \subseteq A \times B$. Thus our input is a bipartite graph $G = (A \cup B, E)$ and the preferences of a vertex are expressed as an ordered list of its neighbors, e.g., $u\colon v, v'$ denotes the preference $v \succ_u v'$, i.e., u prefers v to v'.

▶ **Definition 1.** A matching M in $G = (A \cup B, E)$ is a subset of E such that $|M(u)| \leq \mathsf{cap}(u)$ for each $u \in A \cup B$, where $M(u) = \{v : (u, v) \in M\}$.

The goal is to compute an *optimal* matching in G. The usual definition of optimality in this setting has been *pairwise-stability* [26]. A matching M in G is said to be pairwise-stable if there is no student-course pair (a, b) that "blocks" M. We say a pair $(a, b) \in E \setminus M$ blocks M if (1) either a has less than $\mathsf{cap}(a)$ partners in M or a prefers b to its least preferred neighbor in $M(a)$ and (2) either b has less than $\mathsf{cap}(b)$ partners in M or b prefers a to its least preferred

* The full version of this paper is available at http://arxiv.org/abs/1609.07531.

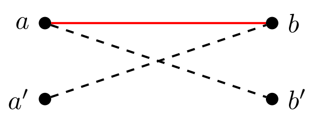
37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017).
Editors: Satya Lokam and R. Ramanujam; Article No. 19; pp. 19:1–19:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

neighbor in $M(b)$. It is known that pairwise-stable matchings always exist [26] and the Gale-Shapley algorithm [8] can be easily generalized to find such a matching in $G = (A \cup B, E)$. The many-to-one variant of this problem, also called the *hospitals/residents* problem, where $\mathsf{cap}(a) = 1$ for every $a \in A$, was studied by Gale and Shapley [8] who showed that their algorithm for the *marriage* problem (where $\mathsf{cap}(u) = 1$ for all $u \in A \cup B$) can be easily extended to find a stable matching in the hospitals/residents problem as well.

Since a (pairwise) stable matching is a maximal matching in G, its size is at least $|M_{\max}|/2$, where $M_{\max}$ is a max-size matching in G. This bound can be tight as shown by the following simple example: let $A = \{a, a'\}$ and $B = \{b, b'\}$ where each vertex has capacity 1 and the edge set is $E = \{(a,b), (a,b'), (a',b)\}$. The preferences are shown in the table below. Here the only stable matching (red line) is $S = \{(a,b)\}$, which is of size 1. However, the max-size matching (dashed lines) $M_{\max} = \{(a',b), (a,b')\}$ is of size 2.

$a\colon b, b'$ $\qquad$ $b\colon a, a'$

$a'\colon b$ $\qquad$ $b'\colon a$



It can be shown that all pairwise-stable matchings have to match the same set of vertices and every vertex gets matched to the same capacity in every pairwise-stable matching. In the hospitals/residents setting, this is popularly called the "Rural Hospitals Theorem" [9, 27]. More precisely, Roth [27] showed that not only is every hospital matched to the same number of residents in every stable matching, but moreover, every hospital that is not matched up to its capacity in some stable matching is actually matched to the same *set of residents* in any stable matching. Thus the notion of stability is very restrictive.

From a social point of view, it seems desirable to have a high number of students registered for courses to make effective use of available resources. Similarly, in the hospitals/residents setting, it seems desirable to have a higher number of residents matched to hospitals in order to keep few residents unemployed and guarantee sufficient staffing for hospitals. The latter point particularly applies to rural hospitals that oftentimes face the problem of being understaffed with residents by the National Resident Matching Program in the USA (cf. [25, 27]). This motivates relaxing the notion of "absence of blocking edges" to a weaker notion of stability so as to obtain matchings that are guaranteed to be significantly larger than $|M_{\max}|/2$. Note that we do not wish to ignore the preferences of vertices and impose a max-size matching on them as such a way of assignment may be highly undesirable from a social viewpoint. Instead our approach is to replace the *local* stability notion of "no blocking edges" with a weaker notion of *global* stability that achieves more "global good" in the sense that its size is always at least $\gamma \cdot |M_{\max}|$ for some $\gamma > 1/2$.

Popularity

To this end, we consider the notion of popularity, which was introduced by Gärdenfors [10] for the stable marriage problem: the input here consists of a set of men and a set of women, where each person seeks to get matched to at most one person from the opposite sex. Hence the marriage or the one-to-one matching setting, where every vertex has capacity 1, is a special case of the many-to-many matching setting considered here. Popularity is based on voting by vertices on the set of feasible matchings. In the one-to-one setting, the preferences of a vertex over its neighbors are extended to preferences over matchings by postulating that every vertex orders matchings in the order of its partners in these matchings. This postulates that vertices do not care which other pairs are formed.

A matching is popular if it never loses a head-to-head election against any matching where each vertex casts a vote. Thus popular matchings are (weak) *Condorcet winners* [5] in the corresponding voting instance. The Condorcet paradox shows that collective preferences can be cyclic and so there need not be a Condorcet winner; this is the source of many impossibility results in social choice theory such as Arrow's impossibility theorem.

However, in the context of matchings in the one-to-one setting, Gärdenfors [10] showed that every stable matching is popular. Hence the fact that stable matchings always exist here [8] implies that popular matchings always exist. This is quite remarkable given that popular matchings correspond to (weak) Condorcet winners. In the one-to-one setting, there is a vast literature on popular matchings [3, 15, 19, 13, 7, 20, 16].

Here we generalize the notion of popularity to the many-to-many matching setting. This requires us to specify how vertices vote over different subsets of their neighbors. In particular, one may want to allow a single vertex to cast multiple votes if its capacity is greater than 1. Our definition of voting by a vertex between two subsets of its neighbors is the following: first remove all vertices that are contained in both sets; then find a bijection from the first set to the second set and compare every vertex with its image under this bijection (if the sets are not of equal size, we add dummy vertices that are less preferred to all non-dummy vertices); the number of wins minus the number of losses is cast as the vote of the vertex. The vote may depend on the bijection that is chosen, however.

Our definition is based on the bijection that *minimizes* the vote, which results in a rather restrictive notion of popularity. We show however that even for this notion of popularity, every stable matching is popular. In particular, popular matchings always exist. As a consequence, popular matchings always exist for every notion of popularity that is less restrictive than our notion of popularity. Our goal is to find a *max-size* popular matching and crucially, it turns out that the size of a max-size popular matching is independent of the bijection that is chosen for the definition of popularity. We formalize these notions below.

In the one-to-one setting, given any two matchings M_0, M_1 and a vertex u, we say u prefers M_0 to M_1 if u prefers $M_0(u)$ to $M_1(u)$, where $M_i(u)$ is u's partner in M_i, for $i = 0, 1$, and we say "$M_i(u) = \text{null}$" if u is left unmatched in matching M_i – note that the null option is the least preferred state for any vertex. Define the function $\mathsf{vote}_u(v, v')$ for any vertex u and neighbors v, v' of u as follows: $\mathsf{vote}_u(v, v')$ is 1 if u prefers v to v', it is -1 if u prefers v' to v, and it is 0 otherwise (i.e., if $v = v'$). In the one-to-one setting, $\Delta_u(M_0, M_1)$, which is u's vote for M_0 versus M_1, is defined to be $\mathsf{vote}_u(M_0(u), M_1(u))$.

In the many-to-many setting, while comparing one matching with another, we allow a vertex to cast more than one vote. For instance, when we compare the preference of vertex u with $\mathsf{cap}(u) = 3$ for $S_0 = \{v_1, v_2, v_3\}$ versus $S_1 = \{v_4, v_5, v_6\}$ (where $v_1 \succ_u v_2 \succ_u \cdots \succ_u v_6$), we would like u's vote to capture the fact that u is better-off by *3 partners* in S_0 when compared to S_1. So we define u's vote for S_0 versus S_1 as follows. Let S_0, S_1 be any two subsets of the set of u's neighbors where we add some occurrences of "null" to the smaller of S_0, S_1 to make the two sets of the same size. We will view the sets $S'_0 = S_0 \setminus S_1$ and $S'_1 = S_1 \setminus S_0$ as arrays $\langle S'_i[1], \ldots, S'_i[k]\rangle$ (for $i = 0, 1$) where $k = |S_0| - |S_0 \cap S_1| = |S_1| - |S_0 \cap S_1|$. The preference of vertex u for S_0 versus S_1, denoted by $\delta_u(S_0, S_1)$, is defined as follows:

$$\delta_u(S_0, S_1) = \min_{\sigma \in \Pi[k]} \sum_{i=1}^{k} \mathsf{vote}_u(S'_0[i], S'_1[\sigma(i)]), \tag{1}$$

where $\Pi[k]$ is the set of permutations on $\{1, \ldots, k\}$. Let $\Delta_u(M_0, M_1) = \delta_u(S_0, S_1)$, where $S_0 = M_0(u)$ and $S_1 = M_1(u)$. So $\Delta_u(M_0, M_1)$ counts the number of votes by u for $M_0(u)$ versus $M_1(u)$ when the sets $S'_0 = M_0(u) \setminus M_1(u)$ and $S'_1 = M_1(u) \setminus M_0(u)$ are being compared in the order that is most adversarial or *negative* for M_0. That is, this order $\sigma \in \Pi[k]$ of comparison between elements of S'_0 and S'_1 gives the least value for $n^+ - n^-$, where n^+ is the number of indices i such that $S'_0[i] \succ_u S'_1[\sigma(i)]$ and n^- is the number of indices i such that $S'_0[i] \prec_u S'_1[\sigma(i)]$. Note that $\Delta_u(M_0, M_1) + \Delta_u(M_1, M_0) \le 0$ and it can be *strictly negative.*

For instance, when a vertex u with $\mathsf{cap}(u) = 3$ compares two subsets $S_0 = \{v_1, v_3, v_5\}$ and $S_1 = \{v_2, v_4, v_6\}$ (where $v_1 \succ_u v_2 \succ_u \cdots \succ_u v_6$), we have $\delta_u(S_0, S_1) = -1$ since comparing the following pairs results in the least value of $\delta_u(S_0, S_1)$: this pairing is (v_1 with v_6), (v_3 with v_2), (v_5 with v_4). This makes $\delta_u(S_0, S_1) = 1 - 1 - 1 = -1$. While computing $\delta_u(S_1, S_0)$, the pairing would be (v_2 with v_1), (v_4 with v_3), (v_6 with v_5): then $\delta_u(S_1, S_0) = -1-1-1 = -3$.

For any two matchings M_0 and M_1 in G, we compare them using the function $\Delta(M_0, M_1)$ defined as follows:

$$\Delta(M_0, M_1) = \sum_{u \in A \cup B} \Delta_u(M_0, M_1). \tag{2}$$

We say M_0 is at least as popular as M_1 if $\Delta(M_0, M_1) \ge 0$ and M_0 is more popular than M_1 if $\Delta(M_0, M_1) > 0$. If $\Delta(M_0, M_1) \ge 0$ then for every vertex u in $A \cup B$: *no matter in which order* the elements of $S'_0 = M_0(u) \setminus M_1(u)$ and $S'_1 = M_1(u) \setminus M_0(u)$ are compared against each other by u in the evaluation of $\Delta_u(M_0, M_1)$, when we sum up the total number of votes cast by all vertices, the votes for M_1 can *never* outnumber the votes for M_0.

▶ **Definition 2.** M_0 is a popular matching in $G = (A \cup B, E)$ if $\Delta(M_0, M_1) \ge 0$ for all matchings M_1 in G.

Thus for a matching M_0 to be popular, it means that M_0 is at least as popular as every matching in G, i.e., there is no matching M_1 with $\Delta(M_0, M_1) < 0$. If there exists a matching M_1 such that $\Delta(M_0, M_1) < 0$ then this is taken as a certificate of *unpopularity* of M_0. Note that it is possible that both $\Delta(M_0, M_1)$ and $\Delta(M_1, M_0)$ are negative, i.e., for each vertex u there is some order of comparison between the elements of $S'_0 = M_0(u) \setminus M_1(u)$ with those in $S'_1 = M_1(u) \setminus M_0(u)$ so that when we sum up the total number of votes cast by all the vertices, the number for M_1 is more than the number for M_0; similarly for each u there is another order of comparison between the elements of S'_0 with those in S'_1 so that when we sum up the total number of votes cast by all the vertices, the number for M_0 is more than the number for M_1. In this case neither M_0 nor M_1 is popular. It is not obvious whether popular matchings always exist in G.

Our definition of popularity may seem too strict and restrictive since for each vertex u, we choose the most negative or adversarial ordering for $M_0(u) \setminus M_1(u)$ versus $M_1(u) \setminus M_0(u)$ while calculating $\Delta_u(M_0, M_1)$. A more relaxed definition may be to order the sets $S'_0 = M_0(u) \setminus M_1(u)$ and $S'_1 = M_1(u) \setminus M_0(u)$ in increasing order of preference of u and take $\sum_i \mathsf{vote}_u(S'_0[i], S'_1[i])$ as u's vote. An even more relaxed definition may be to choose the most favorable or *positive* ordering for S'_0 versus S'_1 while calculating $\Delta_u(M_0, M_1)$. Note that as per (1) we have:

$$-\Delta_u(M_0, M_1) = -\min_{\sigma \in \Pi[k]} \sum_{i=1}^{k} \mathsf{vote}_u(S'_0[i], S'_1[\sigma(i)]) = \max_{\pi \in \Pi[k]} \sum_{i=1}^{k} \mathsf{vote}_u(S'_1[i], S'_0[\pi(i)]). \tag{3}$$

▶ **Definition 3.** Call a matching M_1 *weakly popular* if $\Delta(M_0, M_1) \le 0$, i.e., $-\Delta(M_0, M_1) \ge 0$, for all matchings M_0 in G.

Thus it follows from (3) that M_1 is a weakly popular matching if the sum of votes for M_1 is at least the sum of votes for any matching M_0 when each vertex u compares $M_1(u) \setminus M_0(u)$ versus $M_0(u) \setminus M_1(u)$ in the ordering (as given by π) that is most favorable for M_1. In the one-to-one setting, we have $\Delta(M_0, M_1) + \Delta(M_1, M_0) = 0$ for any pair of matchings M_0, M_1 since $\Delta_u(M_0, M_1) = \mathsf{vote}_u(M_0(u), M_1(u)) = -\mathsf{vote}_u(M_1(u), M_0(u)) = -\Delta_u(M_1, M_0)$ for each u; thus the notions of "popularity" and "weak popularity" coincide here. In the many-to-many setting, weak popularity is a more relaxed notion than popularity.

We choose a strong definition of popularity so that a matching that is popular according to our notion will also be popular according to any notion "in between" between popularity and weak popularity. However this breadth may come at a price as it could be the case that a max-size weakly popular matching is larger than a max-size popular matching.

Our results and techniques

We will show that every pairwise-stable matching in $G = (A \cup B, E)$ is popular, thus our (seemingly strong) definition of popularity is a relaxation of pairwise-stability. We will present a simple linear time algorithm for computing a max-size popular matching M_0 in G and show that $|M_0| \geq \frac{2}{3} \cdot |M_{\max}|$.

We also show that M_0 is more popular than every *larger* matching, i.e., $\Delta(M_0, M_1) > 0$ (refer to (2)) for any matching M_1 that is larger than M_0. Thus M_0 is also a *max-size weakly popular matching* in G as no matching M_1 larger than M_0 can be weakly popular due to the fact that $\Delta(M_0, M_1) > 0$. Thus surprisingly, we lose nothing in terms of the size of our matching by sticking to a strong notion of popularity.

Akin to the rural hospitals theorem, we show that all max-size popular matchings have to match the same set of vertices and every vertex gets matched to the same capacity in every max-size popular matching. However, even in the hospitals/residents setting, hospitals that are not matched up to their capacity in some max-size popular matching do *not* need to be matched to the same sets of residents in any max-size popular matching, which is in contrast to stable matchings [27].

Our algorithm is an extension of the 2-level Gale-Shapley algorithm from [19] to find a max-size popular matching in a stable marriage instance. While the analysis of the 2-level Gale-Shapley algorithm in [19] is based on a structural characterization of popular matchings (from [15]) on forbidden alternating paths and alternating cycles, we use linear programming here to show a simple and self-contained proof of correctness of our algorithm. We would like to remark that the structural characterization from [15] and the proof of correctness from [19] can be extended (in a rather laborious manner) to show the correctness of the generalized algorithm in our more general setting as well, however our proof of correctness is much simpler and this yields a much easier proof of correctness of the algorithm in [19]. Our linear programming techniques are based on a linear program used in [21] to find a popular fractional matching in a bipartite graph with *1-sided preference lists*.

Background and related work

The first algorithmic question studied for popular matchings was in the domain of 1-sided preference lists [1] where it is only vertices on the left, who are *agents*, that have preferences; the vertices on the right are *objects* and they have no preferences. Popular matchings need not always exist here, however fractional matchings that are popular always exist and can be computed in polynomial time via linear programming [21]. Popular matchings always exist in any instance of the stable marriage problem with strict preference lists since every stable matching is popular [10].

Efficient algorithms to find a max-size popular matching in a stable marriage instance are known [15, 19] and a subclass of max-size popular matchings called dominant matchings was studied in [7] and it was shown that these matchings coincide with stable matchings in a larger graph. A polynomial time algorithm was shown in [20] to find a min-cost popular half-integral matching when there is a cost function on the edge set and it was shown in [16] that the popular fractional matching polytope here is half-integral. When preference lists admit ties, the problem of determining if a stable marriage instance $(A \cup B, E)$ admits a popular matching or not is NP-hard [3] and the NP-hardness of this problem holds even when ties are allowed on only one side (say, in the preference lists of vertices in A) [6].

Very recently and independent of our work, the problem of computing a max-size popular matching in an extension of the hospitals/residents problem, i.e., in the *many-to-one* setting, was considered by Nasre and Rawat [23]. The notion of "more popular than" in [23] is weaker than ours: in order to compare matchings M_0 and M_1, in [23] every hospital h orders $S'_0 = M_0(h) \setminus M_1(h)$ and $S'_1 = M_1(h) \setminus M_0(h)$ in increasing order of preference of h and $\sum_i \mathsf{vote}_h(S'_0[i], S'_1[i])$ is h's vote for M_0 versus M_1. An efficient algorithm was shown for their problem by reducing it to a stable matching problem in a larger graph – this closely follows the method and techniques in [15, 19, 7] for the max-size popular matching problem in the one-to-one setting. Note that popularity as per their definition is "in between" our notions of popularity and weak popularity.

The stable matching problem in a marriage instance has been extensively studied – we refer to the books [11, 22] on this topic. The problem of computing stable matchings or its variants in the hospitals/residents setting is also well-studied [14, 2, 12, 17, 18]. The stable matching algorithm in the hospitals/residents problem has several real-world applications: it is used to match residents to hospitals in Canada [28] and in the USA [24]. The many-to-many stable matching problem has also received considerable attention [26, 4, 29].

2 Our algorithm

A first attempt to solve the max-size popular matching problem in a many-to-many instance $G = (A \cup B, E)$ may be to define an equivalent one-to-one instance $G' = (A' \cup B', E')$ by making $\mathsf{cap}(u)$ copies of each $u \in A \cup B$ and $\mathsf{cap}(a) \cdot \mathsf{cap}(b)$ many copies of each edge (a, b); the max-size popular matching problem in G' can be solved using the algorithm in [19] and the obtained matching $\tilde{M}$ in G' can be mapped to a matching M in G. In the first place, one should ensure that there are no multi-edges in M. The matching M will be popular, however it is not obvious that M is a *max-size* popular matching in G as every popular matching in G need not be realized as some popular matching in G': we show such an example in the Appendix. Thus one needs to show that there is at least one max-size popular matching in G that can be realized as a popular matching in G'; we do not pursue this approach here as the running time of the max-size popular matching algorithm in G' is linear in the size of G', which is $O(mn)$, where $|E| = m$ and $|A| + |B| = n$.

In this section we describe a simple $O(m + n)$ algorithm called the generalized *2-level Gale-Shapley algorithm* to compute a max-size popular matching in $G = (A \cup B, E)$. This algorithm works on the graph $H = (A'' \cup B, E'')$ defined as follows: A'' consists of two copies a^0 and a^1 of every student a in A, i.e., $A'' = \{a^0, a^1 : a \in A\}$. The set B of courses in H is the same as in G and the edge set here is $E'' = \{(a^0, b), (a^1, b) : (a, b) \in E\}$.

The preference list of a^i (for $i = 0, 1$) is exactly the same as the preference list of a. The elements in the set $\{a^i : a \in A\}$ will be called *level i* students, for $i = 0, 1$. Every $b \in B$ prefers any level 1 neighbor to any level 0 neighbor: within the set of level i neighbors (for $i = 0, 1$), b's preference order is the same as its original preference order. For instance, if a course b has only 2 neighbors a and v in G where $a \succ_b v$, the preference order of b in

Algorithm 1 *Input:* $H = (A'' \cup B, E'')$; *Output: A matching* M *in* H

1. Initialize $Q = \{a^0 : a \in A\}$ and $M = \emptyset$. Set $\mathsf{residual}(a) = \mathsf{cap}(a)$ for all $a \in A$.
2. **while** $Q \neq \emptyset$ **do**
3. delete the first vertex from Q: call this vertex a^i.
4. **while** a^i has one or more neighbors in H to propose to <u>and</u> $\mathsf{residual}(a) > 0$ **do**
5. – let b be the most preferred neighbor of a^i in H that a^i has not yet proposed to. $\{$*So every neighbor of* a^i *in the current graph* H *that is ranked better than* b *is already matched to* a^i *in* M.$\}$
6. – add the edge (a^i, b) to M.
7. **if** $i = 1$ and b is already matched to a^0 **then**
8. – delete the edge (a^0, b) from M. $\{$*So* (a^0, b) *in* M *gets replaced by* (a^1, b).$\}$
9. **else**
10. – set $\mathsf{residual}(a) = \mathsf{residual}(a) - 1$. $\{$*since* $|M(a)|$ *has increased by 1*$\}$
11. **if** b is matched to more than $\mathsf{cap}(b)$ partners in M **then**
12. – let v^j be b's worst partner in M. Delete the edge (v^j, b) from M. $\{$*Note that "worst" is as per preferences in* H.$\}$
13. – set $\mathsf{residual}(v) = \mathsf{residual}(v) + 1$ and if $v^j \notin Q$ then add v^j to Q.
14. **end if**
15. **end if**
16. **if** b is matched to $\mathsf{cap}(b)$ many partners in M **then**
17. – delete all edges (u, b) from H where u is a neighbor of b in H that is ranked worse than b's worst partner in M. $\{$*"Worse" is as per preferences in* H.$\}$
18. **end if**
19. **end while**
20. **if** $\mathsf{residual}(a) > 0$ and $i = 0$ **then**
21. – add a^1 to Q. $\{$*Though* $\mathsf{residual}(a) > 0$, *the condition in the above while-loop does not hold, i.e.,* a^0 *has no neighbors in* H *to propose to; hence* a^1 *gets activated.*$\}$
22. **end if**
23. **end while**
24. Return the matching M.

G' is: a^1, v^1, a^0, v^0. The *sum* of capacities of a^0 and a^1 will be $\mathsf{cap}(a)$ and we will use $\mathsf{residual}(a)$ to denote the $\mathsf{cap}(a) - |M(a)|$, where M is the current matching. At any point in time, only one of a^0 and a^1 will be *active* in our algorithm.

A description of our algorithm is given as Algorithm 1. To begin with, all level 0 students are active in our algorithm and all level 1 students are inactive. We keep a queue Q of all the active students and they propose as follows:

- every active student a^i, where a is not fully matched, proposes to its most preferred neighbor in H that it has not yet proposed to (lines 4-5 of Algorithm 1)
- if a^0 has already proposed to all its neighbors in H and a is not fully matched, then a^0 becomes inactive and a^1 becomes active and it joins the queue Q (lines 20-21).

When a course b receives a proposal from a^i, the vertex b accepts this offer (in line 6). In case b is already matched to a^0 and it now received a proposal from a^1, the edge (a^0, b) in M is replaced by the edge (a^1, b) (otherwise b would end up being matched to a with multiplicity 2 which is not allowed) – this is done in lines 7-8 of Algorithm 1.

If b is now matched to more than $\mathsf{cap}(b)$ partners then b rejects its worst partner v^j in the current matching and so $\mathsf{residual}(v)$ increases by 1 and v^j joins Q if it is not already in Q

(in lines 11-13). If b is now matched to $\mathsf{cap}(b)$ partners then we delete all edges (u, b) from H where u is a neighbor of b in H that is ranked worse than b's worst partner in the current matching – so no such resident u can propose to b later on in the algorithm (lines 16-17). Once Q becomes empty, the algorithm terminates.

Let M be the matching returned by this algorithm and let M_0 be the matching in G that is obtained by projecting M to the edge set of G, i.e., $(a, b) \in M_0$ if and only if $(a^i, b) \in M$ for some $i \in \{0, 1\}$. We will prove that M_0 is a max-size popular matching in Section 3.

3 The correctness of our algorithm

In this section we show a sufficient condition for a matching N in G to be popular. This is shown via a graph called G'_N: this is a bipartite graph constructed using N such that N gets mapped to a *simple matching* N' in G'_N, i.e., $|N'(v)| \le 1$ for all vertices v in G'_N.

The vertex set of G'_N includes $A' \cup B'$ where $A' = \{a_i : a \in A \text{ and } 1 \le i \le \mathsf{cap}(a)\}$ and $B' = \{b_j : b \in B \text{ and } 1 \le j \le \mathsf{cap}(b)\}$. That is, for each vertex $u \in A \cup B$, there are $\mathsf{cap}(u)$ many copies of u in G'_N. For each edge (a, b) in G such that $(a, b) \in N$, we will arbitrarily choose a distinct $i \in \{1, \ldots, \mathsf{cap}(a)\}$ and a distinct $j \in \{1, \ldots, \mathsf{cap}(b)\}$ and include (a_i, b_j) in N'. If $u \in A \cup B$ was not fully matched in N, i.e., it has less than $\mathsf{cap}(u)$ many partners in N, then some u_k's will be left unmatched in N'.

1. For each edge (a, b) in G such that $(a, b) \notin N$, we will have edges (a_i, b_j) in G'_N, for all $1 \le i \le \mathsf{cap}(a)$ and $1 \le j \le \mathsf{cap}(b)$.
2. For each edge $(a, b) \in N$, we will have the edge (a_i, b_j) in G'_N where $(a_i, b_j) \in N'$.

Thus for any edge $e = (a, b) \notin N$, there are $\mathsf{cap}(a) \cdot \mathsf{cap}(b)$ many copies of e in G': these are (a_i, b_j) for all $(i, j) \in \{1, \ldots, \mathsf{cap}(a)\} \times \{1, \ldots, \mathsf{cap}(b)\}$. However for any edge $(a, b) \in N$, there is only *one* edge (a_i, b_j) in G'_N where $(a_i, b_j) \in N'$, in other words, the student a_i is not adjacent in G'_N to course $b_{j'}$ for $j' \neq j$ and similarly, the course b_j is not adjacent in G'_N to student $a_{i'}$ for $i' \neq i$. The Appendix has an example of G'_N corresponding to a matching N in a many-to-one instance G (see Fig. 2).

There are also some new vertices called "last resort neighbors" in G'_N: for any $u_k \in A' \cup B'$, we introduce a new vertex $\ell(u_k)$; the vertex u_k ranks $\ell(u_k)$ at the bottom of its preference list.

3. The edge set of G'_N also contains the edges $(u_k, \ell(u_k))$ for each $u_k \in A' \cup B'$.

The purpose of the vertex $\ell(u_k)$ is to capture the state of $u_k \in A' \cup B'$ being left unmatched in any matching so that every matching in G gets mapped to an $(A' \cup B')$-*complete matching* in G'_N, i.e., one that matches all vertices in $A' \cup B'$. We will use these last resort neighbors to obtain an $(A' \cup B')$-complete matching N^* from N'.

$$N^* = N' \cup \{(u_k, \ell(u_k)) : u_k \in A' \cup B' \text{ and } u_k \text{ is unmatched in } N'\}.$$

Thus if a vertex $u \in A \cup B$ was not fully matched in N, then some u_i's will be matched to their last resort neighbors in N^*. We now define edge weights in G'_N.

- For any edge $e = (a_i, b_j) \in A' \times B'$: the weight of edge e is $\mathsf{wt}_N(e) = \mathsf{vote}_a(b, N^*(a_i)) + \mathsf{vote}_b(a, N^*(b_j))$, where $N^*(u_k)$ is u_k's partner in the $(A' \cup B')$-complete matching N^*. Thus $\mathsf{wt}_N(a_i, b_j)$ is the sum of votes of a and b for each other versus $N^*(a_i)$ and $N^*(b_j)$, respectively. We have $\mathsf{wt}_N(e) \in \{\pm 2, 0\}$ and $\mathsf{wt}_N(e) = 2$ if and only if e blocks N.

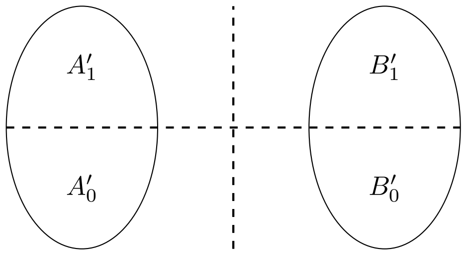


Figure 1 $A' = A'_0 \cup A'_1$ and $B' = B'_0 \cup B'_1$: all courses b_j left unmatched in M'_0 are in B'_0 and all students a_i left unmatched in M'_0 are in A'_1. Note that $M'_0 \subseteq (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$.

- For any edge $e = (u_k, \ell(u_k))$ where $u_k \in A' \cup B'$: the weight of edge e is $\mathsf{wt}_N(e) = \mathsf{vote}_u(\ell(u_k), N^*(u_k))$. Thus $\mathsf{wt}_N(u_k, \ell(u_k)) = -1$ if the vertex u_k was matched in N' and $\mathsf{wt}_N(u_k, \ell(u_k)) = 0$ otherwise (in which case $N^*(u_k) = \ell(u_k)$).

Observe that every edge $e \in N^*$ satisfies $\mathsf{wt}_N(e) = 0$. Thus the weight of the matching N^* in G'_N is 0. Theorem 4 below states that if *every* $(A' \cup B')$-complete matching in the graph G'_N has weight at most 0, then N is a popular matching in G.

The proof of Theorem 4 (given in the full version of the paper) shows that for any matching T in G, we can construct a realization T^* of T in G'_N such that T^* is an $(A' \cup B')$-complete matching and $\mathsf{wt}_N(T^*) = -\Delta(N, T)$. Thus if every $(A' \cup B')$-complete matching in G'_N has weight at most 0, then $\mathsf{wt}_N(T^*) \leq 0$, in other words, $\Delta(N, T) \geq 0$. Since $\Delta(N, T) \geq 0$ for all matchings T in G, the matching N is popular.

▶ **Theorem 4.** *Let N be a matching in G such that every $(A' \cup B')$-complete matching in G'_N has weight at most 0. Then N is popular.*

▶ **Corollary 5.** *Every pairwise-stable matching in G is popular.*

Proof. Let S be any pairwise-stable matching in G. Consider the graph G'_S: since S has no blocking edge in G, every edge e in G'_S satisfies $\mathsf{wt}_S(e) \leq 0$. Thus every matching in G'_S has weight at most 0 and so by Theorem 4, we can conclude that S is popular. ◀

3.1 The popularity of M_0

We will now use Theorem 4 to prove the popularity of the matching M_0 computed in Section 2. We will construct the matchings M'_0, M^*_0 and the graph G'_{M_0} corresponding to the matching M_0 as described at the beginning of Section 3. Our goal is to show that every $(A' \cup B')$-complete matching in G'_{M_0} has weight at most 0. Note that the matching M^*_0 has weight 0 in G'_{M_0}.

We partition the set A' into $A'_0 \cup A'_1$ and the set B' into $B'_0 \cup B'_1$ as follows. Initialize $A'_0 = A'_1 = B'_0 = B'_1 = \emptyset$. For each edge $(a_i, b_j) \in M'_0$ do:
- if $(a^0, b) \in M$ then add a_i to A'_0 and b_j to B'_0;
- else (i.e., $(a^1, b) \in M$) add a_i to A'_1 and b_j to B'_1.

Recall that $M \subseteq A'' \times B$ is the matching in the graph H obtained at the end of the 2-level Gale-Shapley algorithm (see Algorithm 1) and the projection of M on to $A \times B$ is M_0.

The definition of the sets A'_0, A'_1, B'_0, B'_1 implies that $M'_0 \subseteq (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$. Also add students unmatched in M'_0 to A'_1 and courses unmatched in M'_0 to B'_0. Thus we have $A' = A'_0 \cup A'_1$ and $B' = B'_0 \cup B'_1$ (see Fig. 1).

Theorem 6 will show that the matching M_0 satisfies the condition of Theorem 4, this will prove that M_0 is a popular matching in G. This proof is inspired by the proof in [20] that shows the membership of certain half-integral matchings in the popular fractional matching polytope of a stable marriage instance.

In order to show that every $(A' \cup B')$-complete matching in G'_{M_0} has weight at most 0, we consider the max-weight $(A' \cup B')$-complete matching problem in G'_{M_0} as our primal LP. We show a dual feasible solution $\vec{\alpha}$ that makes the dual objective function 0. This means the primal optimal value is at most 0 and this is what we set out to prove.

▶ **Theorem 6.** *Every $(A' \cup B')$-complete matching in G'_{M_0} has weight at most 0.*

Proof. Let our primal LP be the max-weight $(A' \cup B')$-complete matching problem in G'_{M_0}. We want to show that the primal optimal value is at most 0. The primal LP is the following:

$$\max \sum_{e \in G'_{M_0}} \mathsf{wt}_{M_0}(e) \cdot x_e$$

$$\text{subject to} \quad \sum_{e \in E'(u_k)} x_e = 1 \qquad \text{for all } u_k \in A' \cup B',$$

$$x_e \geq 0 \qquad \text{for all edges } e \in G'_{M_0},$$

where $E'(u_k)$ is the set of edges incident on u_k in G'_{M_0}.

The dual LP is the following: we associate a variable α_{u_k} to each vertex $u_k \in A' \cup B'$.

$$\min \sum_{u_k \in A' \cup B'} \alpha_{u_k}$$

$$\text{subject to} \quad \alpha_{a_i} + \alpha_{b_j} \geq \mathsf{wt}_{M_0}(a_i, b_j) \qquad \text{for all edges } (a_i, b_j) \in G'_{M_0}, \tag{4}$$

$$\alpha_{u_k} \geq \mathsf{wt}_{M_0}(u_k, \ell(u_k)) \qquad \text{for all } u_k \in A' \cup B'. \tag{5}$$

Consider the following assignment of α-values for all $u_k \in A' \cup B'$: set $\alpha_{u_k} = 0$ for all u_k unmatched in M'_0 (each such vertex is in $A'_1 \cup B'_0$) and for the matched vertices u_k in M'_0, we set α-values as follows: $\alpha_{u_k} = 1$ if $u_k \in A'_0 \cup B'_1$ and $\alpha_{u_k} = -1$ if $u_k \in A'_1 \cup B'_0$.

Observe that Inequality (5) holds for all vertices $u_k \in A' \cup B'$. This is because $\alpha_{u_k} = 0 = \mathsf{wt}_{M_0}(u_k, \ell(u_k))$ for all u_k unmatched in M'_0; similarly, for all u_k matched in M'_0 we have $\alpha_{u_k} \geq -1 = \mathsf{wt}_{M_0}(u_k, \ell(u_k))$. In order to show Inequality (4), we will use Claim 7 stated below – its proof follows from our algorithm and is included in the full version.

▶ **Claim 7.** *Let $e = (a_i, b_j)$ be any edge in G'_{M_0}.*
(i) *If $e \in A'_1 \times B'_0$, then $\mathsf{wt}_{M_0}(e) = -2$.*
(ii) *If $e \in (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$, then $\mathsf{wt}_{M_0}(e) \leq 0$.*

- Claim 7 (i) says that for every edge $(a_i, b_j) \in A'_1 \times B'_0$ in G'_{M_0}, we have $\mathsf{wt}_{M_0}(a_i, b_j) = -2$. Since $\alpha_{u_k} \geq -1$ for all $u_k \in A'_1 \cup B'_0$, Inequality (4) holds for all edges of G'_{M_0} in $A'_1 \times B'_0$.
- Claim 7 (ii) says that for every edge (a_i, b_j) in $(A'_0 \times B'_0) \cup (A'_1 \times B'_1)$, we have $\mathsf{wt}_{M_0}(a_i, b_j) \leq 0$. Since $\alpha_{a_i} + \alpha_{b_j} \geq 0$ for all $(a_i, b_j) \in A'_t \times B'_t$ (for $t = 0, 1$), Inequality (4) holds for all edges of G'_{M_0} in $(A'_0 \times B'_0) \cup (A'_1 \times B'_1)$.

Since $\mathsf{wt}_{M_0}(e) \leq 2$ for all edges e in G'_{M_0} and we set $\alpha_{u_k} = 1$ for all vertices $u_k \in A'_0 \cup B'_1$, Inequality (4) is satisfied for all edges of G'_{M_0} in $A'_0 \times B'_1$. Thus Inequality (4) holds for all edges (a_i, b_j) in G'_{M_0} and so these α-values are dual feasible.

For every edge $(a_i, b_j) \in M'_0$, we have $\alpha_{a_i} + \alpha_{b_j} = 0$ and $\alpha_{u_k} = 0$ for vertices u_k unmatched in M'_0. Hence it follows that $\sum_{u_k \in A' \cup B'} \alpha_{u_k} = 0$. So by weak duality, the optimal value of the primal LP is at most 0. In other words, every matching in G'_{M_0} that matches all vertices in $A' \cup B'$ has weight at most 0. ◀

3.2 Maximality of the popular matching M_0

We need to show that M_0 is a max-size popular matching in G and we now show that this follows quite easily from the proof of Theorem 6. Let T be any matching in G. We can obtain a realization T^* of the matching T in G'_{M_0} that is absolutely analogous to how it was done to prove Theorem 4. Thus T^* is an $(A' \cup B')$-complete matching in G'_{M_0} and $\mathsf{wt}_{M_0}(T^*) = -\Delta(M_0, T)$.

We know from Theorem 6 that $\mathsf{wt}_{M_0}(T^*) \le 0$. Suppose T is a popular matching in G. Then $\mathsf{wt}_{M_0}(T^*)$ has to be 0, otherwise the popularity of T is contradicted since $\mathsf{wt}_{M_0}(T^*) < 0$ implies that $\Delta(M_0, T) > 0$ (because $\mathsf{wt}_{M_0}(T^*) = -\Delta(M_0, T)$).

So if T is a popular matching in G, then T^* is an optimal solution to the maximum weight $(A' \cup B')$-complete matching problem in G'_{M_0}. Recall that this is the primal LP in the proof of Theorem 6. We will use the dual feasible solution $\vec{\alpha}$ that we constructed in the proof of Theorem 6 and apply complementary slackness to show that if $(u_k, \ell(u_k)) \in M_0^*$, i.e., if u_k is left unmatched in M'_0, then T^* also has to contain the edge $(u_k, \ell(u_k))$. This will imply that $|T| \le |M_0|$, i.e., every popular matching in G has size at most $|M_0|$.

▶ **Lemma 8.** *Let T be a popular matching in G and let T^* be the realization of T in G'_{M_0}. Then for any vertex $u_k \in A' \cup B'$ we have: $(u_k, \ell(u_k)) \in M_0^*$ implies $(u_k, \ell(u_k)) \in T^*$.*

Proof. Consider the α-values assigned to vertices in $A' \cup B'$ in the proof of Theorem 6. This is an *optimal dual* solution since its value is 0 which is the value of the optimal primal solution. Thus complementary slackness conditions have to hold for each edge in the optimal solution $(T^*_e)_{e \in G'_{M_0}}$ to the primal LP. That is, for each edge $(u_k, v_t) \in G'_{M_0}$, we have:

$$\text{either} \quad \alpha_{u_k} + \alpha_{v_t} = \mathsf{wt}_{M_0}(u_k, v_t) \quad \text{or} \quad T^*_{(u_k, v_t)} = 0. \tag{6}$$

Let $u_k \in A' \cup B'$ be a vertex such that $(u_k, \ell(u_k)) \in M_0^*$, so $\alpha_{u_k} = 0$. If $u \in A$, then $u_k \in A'_1$. Observe that all of u_k's neighbors in G'_{M_0} are in B'_1 – this is because for any neighbor $v_t \neq \ell(u_k)$ of u_k, we have $\mathsf{vote}_u(v, \ell(u_k)) = 1$ and so $\mathsf{wt}_{M_0}(u_k, v_t) \ge 0$. Claim 7 (i) says that $\mathsf{wt}_{M_0}(u_k, v_t) = -2$ for all edges $(u_k, v_t) \in A'_1 \times B'_0$. Thus u_k has no neighbor in B'_0. Similarly, if $u \in B$, then $u_k \in B'_0$ and all its neighbors in G'_{M_0} are in A'_0; otherwise u_k has a neighbor v_t in A'_1 and Claim 7 (i) would get contradicted since $\mathsf{wt}_{M_0}(u_k, v_t) \ge 0$.

In both cases, every edge $(u_k, v_t) \in A' \times B'$ that is incident on u_k in G'_{M_0} is *slack* because $(u_k, v_t) \in (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$: thus $\alpha_{u_k} = 0$ and $\alpha_{v_t} = 1$ while $\mathsf{wt}_{M_0}(u_k, v_t) = \mathsf{vote}_u(v, \ell(u_k)) + \mathsf{vote}_v(u, M'_0(v_t)) = 1 - 1 = 0$. Thus it follows from Equation (6) that $T^*_{(u_k, v_t)} = 0$ for $v_t \neq \ell(u_k)$. Since T^* is $(A' \cup B')$-complete, we have $(u_k, \ell(u_k)) \in T^*$. ◀

Now it immediately follows that M_0 is a max-size popular matching in G. Let T be any popular matching in G. Consider the matching $T' = T^* \setminus \{(u_k, \ell(u_k)) : u_k \in A' \cup B'\}$. Lemma 8 implies that $|T'| \le |M'_0|$ because every vertex u_k left unmatched in M'_0 has to be left unmatched in T' also. Since $|T| = |T'|$ and $|M'_0| = |M_0|$, we have $|T| \le |M_0|$. As this holds for any popular matching T in G, we can conclude that M_0 is a max-size popular matching in G.

Our algorithm can be easily implemented to run in linear time (the full version has these details). Hence we can conclude the following theorem.

▶ **Theorem 9.** *A max-size popular matching in a many-to-many instance $G = (A \cup B, E)$ can be computed in linear time.*

Lemma 10 (proved in the full version of the paper) states that no matching larger than M_0 can be weakly popular (see Definition 3) as $\Delta(M_0, T) > 0$ for any such matching T. This implies that M_0 is also a max-size weakly popular matching in G.

▶ **Lemma 10.** *Let T be a matching such that $|T| > |M_0|$. Then $\Delta(M_0, T) > 0$, i.e., M_0 is more popular than T.*

Interestingly, Lemma 10 implies that for any definition of popularity that is "in between" popularity and weak popularity, the size of a max-size popular matching is the same. To formalize the meaning of "in between", consider the two relations on matchings $\succsim_p$ and $\succsim_{wp}$, where $M_0 \succsim_p M_1$ if $\Delta(M_0, M_1) \geq 0$ and $M_0 \succsim_{wp} M_1$ if $\Delta(M_1, M_0) \leq 0$, induced by popularity and weak popularity, respectively. Clearly, $\succsim_p \subseteq \succsim_{wp}$. Note that popular matchings and weakly popular matchings correspond to maximal elements of $\succsim_p$ and $\succsim_{wp}$, respectively.[1] We showed that M_0, which is a max-size maximal element of $\succsim_p$, is also a max-size maximal element of $\succsim_{wp}$. This implies that if $\succsim$ is a relation on matchings (induced by an alternative notion of popularity) such that $\succsim_p \subseteq \succsim \subseteq \succsim_{wp}$, then M_0 is also a max-size maximal element of $\succsim$. This allows us to conclude the following proposition which even allows for different vertices to compare sets of neighbors in different ways.

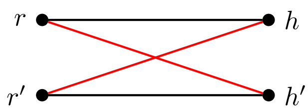
▶ **Proposition 11.** The size of a max-size popular matching in $G = (A \cup B, E)$ is invariant to the way vertices compare sets of neighbors as long as it is in between the most adversarial and the most favorable comparison.

We now briefly discuss some other results that we show here. The rural hospitals theorem for stable matchings [27] does not necessarily hold for max-size popular matchings. That is, a hospital that is not matched up to capacity in some max-size popular matching is not necessarily matched to the same set of residents in every max-size popular matching.

Consider the instance $G = (R \cup H, E)$ with $R = \{r, r'\}$ and $H = \{h, h'\}$ and $\mathsf{cap}(h) = 1$ and $\mathsf{cap}(h') = 2$. The edge set is $R \times H$. The preferences are shown in the table below. The (max-size) popular matchings are $M = \{(r, h), (r', h')\}$ (in black) and $M' = \{(r, h'), (r', h)\}$ (in red). So h' is matched to a different resident in the two max-size popular matchings M and M'. Note that M' is not stable, as (r, h) is a blocking pair.

$r: h, h'$ $\qquad\qquad$ $h: r, r'$

$r': h, h'$ $\qquad\qquad$ $h': r, r'$



However Lemma 12 (proved in the full version) holds here. Such a result for max-size popular matchings in the one-to-one setting (that every max-size popular matching has to match the same set of vertices) was shown in [13]. Our proof is based on linear programming and is different from the combinatorial proof in [13].

▶ **Lemma 12.** *Let T be a max-size popular matching in G. Then T matches the same vertices as M_0 (the matching computed in Section 2) and moreover, every vertex u is matched in T to the same capacity as it gets matched to in M_0.*

The following results are also included in the full version. These proofs are inspired by analogous proofs in the one-to-one setting shown in [19] and in [15], respectively.

▶ **Lemma 13.** *We have $|M_0| \geq \frac{2}{3}|M_{\max}|$, where M_0 is a max-size popular matching in G and M_{max} is a max-size matching in G.*

▶ **Lemma 14.** *A pairwise-stable matching S is a min-size weakly popular matching in G.*

[1] M_0 is a maximal element of a relation $\succsim$ if for all elements M_1 we have: $M_1 \succsim M_0$ implies $M_0 \sim M_1$.

Acknowledgments. The first author wishes to thank Larry Samuelson for comments on the motivation for popular matchings. The second author wishes to thank David Manlove and Bruno Escoffier for asking her about popular matchings in the hospitals/residents setting.

References

1 D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.

2 G. Askalidis, N. Immorlica, A. Kwanashie, D. Manlove, and E. Pountourakis. Socially stable matchings in the hospitals/residents problem. In *the 13th International Symposium on Algorithms and Data Structures (WADS)*, pages 85–96, 2013.

3 P. Biró, R. W. Irving, and D. F. Manlove. Popular matchings in the marriage and roommates problems. In *the 7th International Conference on Algorithms and Complexity (CIAC)*, pages 97–108, 2010.

4 C. Blair. The lattice structure of the set of stable matchings with multiple partners. *Mathematics of Operations Research*, 13:619–628, 1988.

5 Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, 1785. Facsimile published in 1972 by Chelsea Publishing Company, New York.

6 Á. Cseh, C.-C. Huang, and T. Kavitha. Popular matchings with two-sided preferences and one-sided ties. In *the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 367–379, 2015.

7 Á. Cseh and T. Kavitha. Popular edges and dominant matchings. In *the 18th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 138–151, 2016.

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

9 D. Gale and M. Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223–232, 1985.

10 P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Sciences*, 20(3):166–173, 1975.

11 D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* MIT Press, Boston, MA, 1989.

12 K. Hamada, K. Iwama, and Shuichi Miyazaki. The hospitals/residents problem with lower quotas. *Algorithmica*, 74(1):440–465, 2016.

13 M. Hirakawa, Y. Yamauchi, S. Kijima, and M. Yamashita. On the structure of popular matchings in the stable marriage problem - who can join a popular matching? In the 3rd International Workshop on Matching Under Preferences (MATCH-UP), 2015.

14 C.-C. Huang. Classified stable matching. In *the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1235–1253, 2010.

15 C.-C. Huang and T. Kavitha. Popular matchings in the stable marriage problem. *Information and Computation*, 222:180–194, 2013.

16 C.-C. Huang and T. Kavitha. Popularity, self-duality, and mixed matchings. In *the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2294–2310, 2017.

17 R. W. Irving, D. F. Manlove, and S. Scott. The hospitals/residents problem with ties. In *the 7th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 259–271, 2000.

18 R. W. Irving, D. F. Manlove, and S. Scott. Strong stability in the hospitals/residents problem. In *the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 439–450, 2003.

19 T. Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43(1):52–71, 2014.

20 T. Kavitha. Popular half-integral matchings. In *the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 22.1–22.13, 2016.

21 T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. *Theoretical Computer Science*, 412(24):2679–2690, 2011.

22 D. F. Manlove. *Algorithmics of Matching Under Preferences.* World Scientific Publishing Company, 2013.

23 M. Nasre and A. Rawat. Popularity in the generalized hospital residents setting. In *the 12th International Computer Science Symposium in Russia (CSR)*, pages 245–259, 2017.

24 National Resident Matching Program. Why the match? Web document available at `http://www.nrmp.org/whythematch.pdf`.

25 A. E. Roth. The evolution of the labor market for medical interns and resident: a case study in game theory. *The Journal of Political Economy*, 92(6):991–1016, 1984.

26 A. E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52:47–57, 1984.

27 A. E. Roth. On the allocation of residents to rural hospitals: A general property of two-sided matching markets. *Econometrica*, 54(2):425–427, 1986.

28 Canadian Resident Matching Service. How the matching algorithm works. Web document available at `http://carms.ca/algorithm.htm`.

29 M. Sotomayor. Three remarks on the many-to-many stable matching problem. *Mathematical Social Sciences*, 38:55–70, 1999.

A A naive approach for finding max-size popular matchings

Given a many-to-many matching instance $G = (A \cup B, E)$, we investigate the possibility of constructing a corresponding one-to-one matching instance $G' = (A' \cup B', E')$ (with *strict* preference lists) in order to show a reduction from the max-size popular matching problem in G to one in G'. The vertex set A' will have $\mathsf{cap}(a)$ many copies $a_1, a_2, \ldots$ of every $a \in A$ and B' will have $\mathsf{cap}(b)$ many copies $b_1, b_2, \ldots$ of every $b \in B$; the edge set E' has $\mathsf{cap}(a) \cdot \mathsf{cap}(b)$ many copies of edge (a, b) in E. If $v \succ_u v'$ in G then we have $v_i \succ_{u_k} v'_j$ for each $i \in \{1, \ldots, \mathsf{cap}(v)\}$, $j \in \{1, \ldots, \mathsf{cap}(v')\}$, and $k \in \{1, \ldots, \mathsf{cap}(u)\}$. Among the copies $v_1, \ldots, v_{\mathsf{cap}(v)}$ of the same vertex v, we will set $v_1 \succ_{u_k} \cdots \succ_{u_k} v_{\mathsf{cap}(v)}$.

Given any matching $\tilde{M}$ in G', we define $\mathsf{proj}(\tilde{M})$ as the projection of $\tilde{M}$, which is obtained by dropping the subscripts of all vertices. We will now consider the *many-to-one* or the hospitals/ residents setting: so there are no multi-edges in $\mathsf{proj}(\tilde{M})$. As $\mathsf{proj}(\tilde{M})$ obeys all capacity bounds, it is a valid matching in G. It would be interesting to be able to show that every popular matching M in G has a *realization* $\tilde{M}$ in G' (i.e., $\mathsf{proj}(\tilde{M}) = M$) such that $\tilde{M}$ is a popular matching in G'.

However the above statement is not true as shown by the following example. Let $G = (R \cup H, E)$ where $R = \{p, q, r, s\}$ and $H = \{h, h', h''\}$ where $\mathsf{cap}(h) = 2$ and $\mathsf{cap}(u) = 1$ for all other vertices u. The preference lists are as follows:

$p\colon h, h''$	$h\colon p, q, r, s$
$q\colon h, h'$	$h'\colon q$
$r\colon h$	$h''\colon p$
$s\colon h$	

Consider the matching $N = \{(p, h), (q, h'), (r, h)\}$. We show below that N satisfies the sufficient condition for popularity as given in Theorem 4. The proof of Claim 15 follows the same approach as used in the proof of Theorem 6.

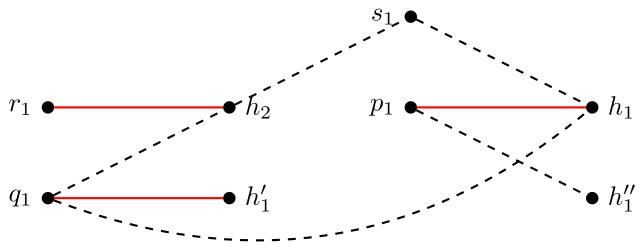


Figure 2 The edges of the matching N' are in red and the non-matching edges in G'_N are dashed. For simplicity, we have not included last resort neighbors here. Note that the edge (q_1, h_2) is a blocking edge to N' as both q_1 and h_2 prefer each other to their respective partners in N', i.e., $\mathsf{wt}_N(q_1, h_2) = 2$ and $\mathsf{wt}_N(e) = 0$ for all other edges e in G'_N.

▶ **Claim 15.** *N is popular in G.*

Proof. We have $R' = \{p_1, q_1, r_1, s_1\}$ and $H' = \{h_1, h_2, h'_1, h''_1\}$. Here we use the notation introduced at the beginning of Section 3: let $N' = \{(p_1, h_1), (q_1, h'_1), (r_1, h_2)\}$ (see Fig. 2).

We need to show that every $(R' \cup H')$-complete matching in the weighted graph G'_N has weight at most 0. We will show this by constructing a *witness* or a solution to the dual LP corresponding to the primal LP which is the $(R' \cup H')$-complete max-weight matching problem in G'_N. This solution is the following: $\alpha_{p_1} = \alpha_{h_1} = \alpha_{s_1} = \alpha_{h''_1} = 0$ while $\alpha_{q_1} = \alpha_{h_2} = 1$ and $\alpha_{r_1} = \alpha_{h'_1} = -1$. The above solution is dual-feasible since every edge in G'_N is covered by the sum of α-values of its endpoints – in particular, note that $\alpha_{q_1} + \alpha_{h_2} = 2 = \mathsf{wt}_N(q_1, h_2)$. The dual optimal solution is at most $\sum_{u \in R' \cup H'} \alpha_u = 0$. So the primal optimal solution is also at most 0, in other words, N is a popular matching in G. ◀

Note that the graph G' has two *extra* edges relative to G'_N: these are (p_1, h_2) and (r_1, h_1). With respect to realizations of N in G', there are 2 candidates: these are $N_1 = \{(p_1, h_1), (q_1, h'_1), (r_1, h_2)\}$ and $N_2 = \{(p_1, h_2), (q_1, h'_1), (r_1, h_1)\}$.

▶ **Claim 16.** *Neither N_1 nor N_2 is popular in G'.*

Proof. Consider the matching $M_1 = \{(p_1, h''_1), (q_1, h_2), (r_1, h_1)\}$. The vertices p_1, h_1, and h'_1 prefer N_1 to M_1 while the vertices q_1, h_2, r_1, and h''_1 prefer M_1 to N_1 and s_1 is indifferent. Thus M_1 is more popular than N_1, i.e., N_1 is not a popular matching in G'.

Consider the matching $M_2 = \{(p_1, h_1), (q_1, h'_1), (s_1, h_2)\}$. The vertices r_1 and h_2 prefer N_2 to M_2 while the vertices p_1, h_1, and s_1 prefer M_2 to N_2 and q_1, h'_1, and h''_1 are indifferent. Thus M_2 is more popular than N_2, i.e., N_2 is not a popular matching in G'. ◀

Summarizing, there may exist popular matchings in G that cannot be realized as popular matchings in G'. Thus in order to claim that that $\mathsf{proj}(\tilde{M})$ is a *max-size* popular matching in G when $\tilde{M}$ is a max-size popular matching in G', it needs to be shown that there is at least one max-size popular matching in G that can be realized as a popular matching in G'.