Algorithmica
https://doi.org/10.1007/s00453-019-00553-0

Two Problems in Max-Size Popular Matchings

Florian Brandl[1] · Telikepalli Kavitha[2]

Received: 24 May 2018 / Accepted: 8 February 2019

Abstract

We study popular matchings in the many-to-many matching problem, which is given by a graph $G = (V, E)$ and, for every agent $u \in V$, a capacity $\mathsf{cap}(u) \geq 1$ and a preference list strictly ranking her neighbors. A matching in G is popular if it weakly beats every matching in a majority vote when agents cast votes for one matching versus the other according to their preferences. First, we show that when $G = (A \cup B, E)$ is bipartite, e.g., when matching students to courses, every pairwise stable matching is popular. In particular, popular matchings are guaranteed to exist. Our main contribution is to show that a max-size popular matching in G can be computed in linear time by the *2-level Gale–Shapley* algorithm, which is an extension of the classical Gale–Shapley algorithm. We prove its correctness via linear programming. Second, we consider the problem of computing a max-size popular matching in $G = (V, E)$ (not necessarily bipartite) when every agent has capacity 1, e.g., when matching students to dorm rooms. We show that even when G admits a stable matching, this problem is NP-hard, which is in contrast to the tractability result in bipartite graphs.

Keywords Matchings under preferences · Gale–Shapley algorithm · Linear programming duality · NP-hardness

1 Introduction

We study the many-to-many matching problem, also known as the *b-matching* problem, where every agent has a capacity and seeks to get matched to a subset of other

A preliminary version of this paper appeared in FSTTCS 2017 [5] and Sect. 4 is new.

✉ Telikepalli Kavitha
kavitha@tcs.tifr.res.in

Florian Brandl
brandlfl@in.tum.de

1 Technische Universität München, Munich, Germany

2 Tata Institute of Fundamental Research, Mumbai, India

agents that does not exceed her capacity. In addition to the capacity constraint, every agent can specify which agents are acceptable to her, together with a strict ranking over those. Hence, a many-to-many matching instance is formally given by a graph $G = (V, E)$, where V is the set of agents and E is the set of mutually acceptable pairs, i.e., if $(u, v) \in E$, then v is acceptable to u and *vice versa*. The capacity of an agent $u \in V$ is $\mathsf{cap}(u) \geq 1$ and $\succ_u$ is a strict ranking over the neighbors of u in G. We write the preferences of u as an ordered list of his neighbors, e.g., $u\colon v, v'$ denotes the preference $v \succ_u v'$, i.e., u prefers v to v'.

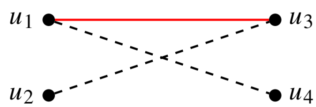
Definition 1 A matching M in $G = (V, E)$ is a subset of E such that $|M(u)| \leq \mathsf{cap}(u)$ for each $u \in V$, where $M(u) = \{v : (u, v) \in M\}$.

The goal is to compute an *optimal* matching in G. The usual definition of optimality in the many-to-many matching setting has been *pairwise-stability* [35]. A matching M in G is said to be pairwise-stable if there is no pair of agents (u, v) that "blocks" M. We say a pair $(u, v) \in E \setminus M$ blocks M if (1) either u has less than $\mathsf{cap}(u)$ partners in M or u prefers v to its worst partner in M and (2) either v has less than $\mathsf{cap}(v)$ partners in M or v prefers u to its worst partner in M.

When the graph G is bipartite, it is known that pairwise-stable matchings always exist [35] and the Gale–Shapley algorithm [12] for the one-to-one variant, or the *marriage* problem, can be easily generalized to find such a matching in $G = (A \cup B, E)$. The many-to-one variant in bipartite graphs, also called the *hospitals/residents* problem, where $\mathsf{cap}(a) = 1$ for every $a \in A$, was studied by Gale and Shapley [12] who showed that their algorithm for the marriage problem extends to the hospitals/residents problem. However for a general instance $G = (V, E)$, i.e., G is not necessarily bipartite, stable matchings need not exist even in the one-to-one variant, also known as the *roommates* problem. For instance, consider the triangle on 3 agents a, b, c where a prefers b to c, while b prefers c to a, and c prefers a to b. It is easy to check that this instance has no stable matching.

We focus on bipartite graphs first. Here (pairwise) stable matchings always exist and since these are maximal matchings in G, the size of such a matching 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 $V = \{u_1, u_2, u_3, u_4\}$ where each agent has capacity 1 and the edge set is $E = \{(u_1, u_3), (u_1, u_4), (u_2, u_3)\}$. The preferences are shown in the table below. Here the only stable matching (bold line) is $S = \{(u_1, u_3)\}$, which is of size 1. The max-size matching (dashed lines) $M_{\max} = \{(u_1, u_4), (u_2, u_3)\}$ is of size 2.

$u_1\colon u_3, u_4$	$u_3\colon u_1, u_2$
$u_2\colon u_3$	$u_4\colon u_1$



For many applications of matching problems, it seems desirable to have a high number of matched agents. For example, when matching students to courses, a larger number of matches makes effective use of the available resources. Similarly, when matching hospitals to residents, a larger number of matches keeps few residents unemployed and guarantees 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. [34,36]). This motivates relaxing the notion of "absence of blocking pairs" 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 the agents 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 pairs" 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 $\gamma > 1/2$.

1.1 Popularity

To this end, we consider the notion of popularity, which was introduced by Gärdenfors [14] in the one-to-one matching setting in bipartite graphs or the 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. Thus every agent has capacity 1 here. Popularity is based on voting by agents on the set of feasible matchings. The preferences of an agent over its neighbors are extended to preferences over matchings by postulating that every agent orders matchings in the order of its partners in these matchings. This postulates that agents 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 agent casts a vote. Thus popular matchings are (weak) *Condorcet winners* [7] in the corresponding voting instance. The definition of popular matchings, originally made in the marriage setting, holds in the roommates setting as well.

The Condorcet paradox shows that collective preferences can be cyclic and so there need not be a Condorcet winner in a general voting instance; 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 marriage problem, Gärdenfors [14] showed that every stable matching is popular. Hence the fact that stable matchings always exist here [12] implies that popular matchings always exist. This is quite remarkable given that popular matchings correspond to (weak) Condorcet winners. In the one-to-one matching setting in bipartite graphs, there is a vast literature on popular matchings [4, 10,18,20,21,25,26].

Here we generalize the notion of popularity to the many-to-many matching setting. This requires us to specify how agents vote over different subsets of their neighbors. In particular, one may want to allow a single agent to cast multiple votes if its capacity is greater than 1. Our definition of voting by an agent between two subsets of her neighbors is the following: first remove all agents that are contained in both sets; then find a bijection from the first set to the second set and compare every agent with its image under this bijection (if the sets are not of equal size, we add dummy agents that are less preferred than all non-dummy agents); the number of wins minus the number of losses is cast as the vote of the agent. 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 pairwise-stable matching is popular. In particular, popular matchings always exist in bipartite instances. As a consequence, popular matchings always exist in bipartite instances 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 all these notions below.

1.2 Definitions

In the one-to-one setting, given any two matchings M_0, M_1 and an agent 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 $i = 0, 1$. 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 agent. Define the function $\mathsf{vote}_u(v, v')$ for any agent u and neighbors v, v' of u as follows:

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

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 an agent to cast more than one vote. For instance, when we compare the preference of agent 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 agent 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 $\Delta_u(M_0, M_1) + \Delta_u(M_1, M_0) \leq 0$ and it can be *strictly negative*.

For instance, when an agent 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:

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

We say M_0 is at least as popular as M_1 if $\Delta(M_0, M_1) \geq 0$ and M_0 is more popular than M_1 if $\Delta(M_0, M_1) > 0$. If $\Delta(M_0, M_1) \geq 0$ then for every agent u in V: *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 agents, the votes for M_1 can *never* outnumber the votes for M_0.

Definition 2 M_0 is a popular matching in $G = (V, E)$ if $\Delta(M_0, M_1) \geq 0$ 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 agent 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 agents, 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 agents, 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 agent 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:

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

$$= \max_{\pi \in \Pi[k]} \sum_{i=1}^{k} \mathsf{vote}_u(S'_1[i], S'_0[\pi(i)]). \tag{4}$$

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

Thus it follows from (4) 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 agent 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" 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.

1.3 Results for Bipartite Graphs

We show that if $G = (A \cup B, E)$ is bipartite, every pairwise-stable matching in G is popular. Thus, our seemingly strong definition of popularity is a relaxation of pairwise-stability. We present a simple linear time algorithm for computing a max-size popular matching M_0 in $G = (A \cup B, E)$ and show that $|M_0| \geq \frac{2}{3} \cdot |M_{\max}|$. Our algorithm is an extension of the 2-level Gale–Shapley algorithm from [25] to find a max-size popular matching in a stable marriage instance, i.e., the one-to-one setting. While the analysis of the 2-level Gale–Shapley algorithm in [25] is based on a structural characterization of popular matchings (from [20]) 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 [20] and the proof of correctness of the algorithm in the one-to-one setting from [25] can be extended to show the correctness of the generalized algorithm in the many-to-many setting as well. However, our proof of correctness is much simpler, which yields a much easier proof of correctness of the algorithm in [25]. Our linear programming techniques are based on a linear program used in [28] to find a popular fractional matching in a bipartite graph with *1-sided preference lists*.

We also show that M_0 is more popular than every *larger* matching, i.e., for any matching M_1 larger than M_0, it holds that $\Delta(M_0, M_1) > 0$ (refer to (2)). 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 [13,36], we show that all max-size popular matchings have to match the same set of agents and every agent gets matched to the same number of agents 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 [36].

1.4 Results for Roommates Instances

As mentioned earlier, stable matchings need not always exist in a roommates instance $G = (V, E)$, i.e., if $\mathsf{cap}(u) = 1$ for all $u \in V$. There are several polynomial time algorithms [22,38,39] to determine if a stable matching exists or not in a given roommates instance. The problem we consider here is the complexity of finding a max-size popular matching in a roommates instance G (assuming G admits one or more popular matchings). We show that the max-size popular matching problem in a roommates instance with strict preference lists is NP-hard, even in instances that admit stable matchings. Note that a stable matching is a min-size popular matching in G [20].

The rural hospitals theorem does not necessarily hold for max-size popular matchings in roommates instances, i.e., the set of agents matched in every max-size popular matching need not be the same. An instance where different subsets of agents get matched in different max-size popular matchings forms the main gadget in the proof of NP-hardness for finding a max-size popular matching in a roommates instance.

1.5 Background and Related Work

The first algorithmic question studied for popular matchings was in the domain of 1-sided preference lists [1] in bipartite graphs where vertices on the left side are agents and only they 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 [28]. Popular matchings always exist in any instance of the stable marriage problem with strict preference lists since every stable matching is popular [14].

Efficient algorithms to find a max-size popular matching in a stable marriage instance are known [20,25] and a subclass of max-size popular matchings called dominant matchings was studied in [10] and it was shown that these matchings coincide with stable matchings in a larger graph. A polynomial time algorithm was shown in [26] to find a min-cost popular half-integral matching when there is a cost function on the edge set and it was shown in [21] that the popular fractional matching polytope here is half-integral. When preference lists admit ties, the problem of determining if a stable marriage instance $G = (A \cup B, E)$ admits a popular matching or not is NP-hard [4] and the NP-hardness of this problem holds even when ties are allowed on only one side (say, in the preference lists of agents in A) [9].

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 [32]. The notion of "more popular than" in [32] is weaker than ours: in order to compare matchings M_0 and M_1, in [32] 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 [10,20,25] 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.

Very recently, Király and Mészáros-Karkus [29] considered the problem of computing *strongly popular b*-matchings in the many-to-many setting in bipartite graphs where agents on one side have strict preferences while agents on the other side may have one tie of arbitrary length at the end of their preference list and showed a polynomial time algorithm for deciding if a strongly popular matching exists or not. A matching M is strongly popular if $\Delta(M, M') > 0$ for every matching $M' \neq M$ in the input instance. The definition of $\Delta(\cdot, \cdot)$ is as given in Definition 2.

The stable matching problem in a marriage instance has been extensively studied and we refer to the books [16,30] on this topic. The problem of computing stable matchings or its variants in the hospitals/residents setting is also well-studied [2,17,19, 23,24]. The stable matching algorithm in the hospitals/residents problem has several real-world applications and it is used to match residents to hospitals in Canada [6] and in the USA [33]. The many-to-many stable matching problem has also received considerable attention [3,35,37].

The popular roommates problem is to decide if a given instance $G = (V, E)$ admits a popular matching or not. The complexity of the popular roommates problem was an open problem for several years [4] and very recently, NP-hardness of the popular roommates problem has been shown [11,15]. The NP-hardness of the max-size popular matching problem in a roommates instance that we show here is independent of these hardness results as we are in an *easy* case of the popular roommates problem because our instance admits a stable matching.

Interestingly, the NP-hardness proof in [15] uses our reduction as one of their main gadgets—they cite the arxiv paper [27] where our hardness result was announced. The paper [11] also shows that it is NP-hard to decide if a roommates instance with a stable matching admits a *dominant* matching or not; the NP-hardness of the max-size popular matching problem in a roommates instance with stable matchings can be derived from this reduction. However this hardness result is not claimed in [11] since our result was announced earlier [27]. Our proof is much simpler than this proof in [11] and as mentioned earlier, it is based on the interesting observation that the rural hospitals theorem is not true for max-size popular matchings in roommates instances. The complexity of the max-size popular matching problem in a roommates instance (that admits stable matchings) was stated as an open problem in [4] and [8].

Organization of the paper We present the generalized 2-level Gale–Shapley algorithm for computing a max-size popular matching in a many-to-many bipartite instance in Sect. 2. We analyze this algorithm in Sect. 3 and prove its correctness. Our hardness result for max-size popular matchings in roommates instances is given in Sect. 4.

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 [25] and the obtained matching $\tilde{M}$ in G' can be mapped to a matching M in G. One should also ensure that there are no multi-edges in M.

The matching M will be popular in G, 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 $O(mn)$ (linear in the size of G') where $|E| = m$ and $|A| + |B| = n$.

2.1 A Linear Time Algorithm

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 $G = (A \cup B, E)$. It will be convenient to refer to vertices in A as *students* and to vertices in B as *courses*. Our 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 $E'' = \{(a^0, b), (a^1, b) : (a, b) \in E\}$.

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.

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 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 $\mathsf{cap}(a) - |M(a)|$, 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, i.e., $|M(a)| < \mathsf{cap}(a)$, 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.

We illustrate an example run of the algorithm on the instance on u_1, u_2, u_3, u_4 described in Sect. 1. Each vertex here has capacity 1. Initially $Q = \{u_1^0, u_2^0\}$: let u_2^0 be the first vertex in Q. So u_2^0 is deleted from Q and it proposes to u_3; this makes $M = \{(u_2^0, u_3)\}$. Then u_1^0 gets deleted from Q; as u_3 is its top neighbor, u_1^0 proposes to u_3. This results in $M = \{(u_1^0, u_3)\}$ as (u_2^0, u_3) gets deleted from M because u_3 prefers u_1^0 to u_2^0. Now u_2^0 has no other neighbors to propose to and $\mathsf{residual}(u_2) > 0$, so u_2^1 is added to Q. Then u_2^1 proposes to u_3 and this results in $M = \{(u_2^1, u_3)\}$ as u_3 prefers u_2^1 to u_1^0. Then u_1^0 proposes to its second choice neighbor u_4 and thus $M = \{(u_1^0, u_4), (u_2^1, u_3)\}$.

The matching M_0 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 $i \in \{0, 1\}$. Thus $M_0 = \{(u_1, u_4), (u_2, u_3)\}$ in our example. We will prove in Sect. 3 that M_0 is a max-size popular matching in G.

3 The Correctness of Our Algorithm

In this section we show a sufficient condition for a matching N in $G = (A \cup B, E)$ 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., $|N'(u_i)| \le 1$ for all vertices u_i in G'_N.

The vertex set of G'_N is $A' \cup B'$ 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 choose an $i \in \{1, \ldots, \mathsf{cap}(a)\}$ and a $j \in \{1, \ldots, \mathsf{cap}(b)\}$ such that i and j have not been chosen so far 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 $(a, b) \in N$, we will have the edge (a_i, b_j) in G'_N where $(a_i, b_j) \in N'$.
2. 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 $1 \le j \le \mathsf{cap}(b)$.

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 $j' \ne j$ and similarly, the course b_j is not adjacent in G'_N to student $a_{i'}$ for $i' \ne i$. Fig. 4 in the "Appendix" has an example of G'_N corresponding to a matching N in a many-to-one instance G.

3. The edge set of G'_N also contains self-loops (u_i, u_i) for all $u_i \in A' \cup B'$.

The purpose of these self-loops is to ensure that every matching in G gets mapped to a *perfect matching* in G'_N. The perfect matching N^* corresponding to N is

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

Thus if a vertex $u \in A \cup B$ was not fully matched in N, then some u_i's will be matched to themselves (along self-loops) 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 perfect matching N^*. Note that $\mathsf{vote}_u(v, v'_k) = \mathsf{vote}_u(v, v')$ for any vertex u and neighbors v, v'.
  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.
- For any edge $e = (u_i, u_i)$ where $u_i \in A' \cup B'$: the weight of edge e is $\mathsf{wt}_N(e) = \mathsf{vote}_u(u, N^*(u_i))$. Thus $\mathsf{wt}_N(u_i, u_i) = -1$ if u_i was matched in N' and $\mathsf{wt}_N(u_i, u_i) = 0$ otherwise (in which case $N^*(u_i) = u_i$).

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 1 below states that if *every* perfect matching in the graph G'_N has weight at most 0, then N is a popular matching in G.

3.1 A Sufficient Condition for Popularity

Theorem 1 *Let N be a matching in G such that every perfect matching in G'_N has weight at most 0. Then N is popular.*

Proof For any matching T in G, we will show a realization T^* of T in G'_N such that T^* is a perfect matching and $\mathsf{wt}_N(T^*) = -\Delta(N, T)$. Thus if every perfect matching in G'_N has weight at most 0, then $\mathsf{wt}_N(T^*) \le 0$, in other words, $\Delta(N, T) \ge 0$. Since $\Delta(N, T) \ge 0$ for all matchings T in G, the matching N will be popular.

In order to construct T^*, corresponding to each edge $(a, b) \in T$, we will find appropriate indices $s \in \{1, \dots, \mathsf{cap}(a)\}$ and $t \in \{1, \dots, \mathsf{cap}(b)\}$, where (a_s, b_t) is in G'_N, such that $(a_s, b_t) \in T^*$; there may also be some self-loops in T^*.

(i) For every edge $(a, b) \in N \cap T$ do: add the edge (a_i, b_j) to T^* where $(a_i, b_j) \in N^*$.

(ii) For every $(a, b) \in T \setminus N$, we have to decide the indices (s, t) such that $(a_s, b_t) \in T^*$. In the evaluation of $\Delta_a(N, T)$, while comparing the sets $N(a) \setminus T(a)$ and $T(a) \setminus N(a)$ [refer to Eq. (1) in Sect. 1]:

- Let b' be the course that a compares b with. So the matching N^* contains the edge (a_i, b'_j) for some (i, j) and we now need to decide the index k such that T^* will contain (a_i, b_k). In the evaluation of $\Delta_b(N, T)$, while comparing the sets $N(b) \setminus T(b)$ and $T(b) \setminus N(b)$:
    - Let a' be the student that b compares a with. So the matching N^* contains the edge $(a'_{i'}, b_{j'})$ for some (i', j'). We include the edge $(a_i, b_{j'})$ in T^*.
    - If a is compared with "null" by b (so b is not fully matched in N), then we include (a_i, b_k) in T^* for some k such that $(b_k, b_k) \in N^*$ and b_k is unmatched so far in T^*.
- Suppose b is compared with "null" by a (so a is not fully matched in N).
    - Let a' be the student that b compares a with in the evaluation of $\Delta_b(N, T)$ and so $(a'_{i'}, b_{j'}) \in N^*$ for some (i', j'). We include the edge $(a_k, b_{j'})$ in T^* for some a_k such that $(a_k, a_k) \in N^*$ and a_k is unmatched so far in T^*.
    - In case a is compared with "null" by b, then we include the edge $(a_{k'}, b_k)$ in T^* for some k' and k such that $(a_{k'}, a_{k'})$ and (b_k, b_k) are in N^* and $a_{k'}$ and b_k are unmatched so far in T^*.

(iii) For any vertex $u_k \in A' \cup B'$ that is left unmatched in steps (i)-(ii), include (u_k, u_k) in T^*.

It is easy to see that T^* is a valid matching in G'_N and it matches all vertices in $A' \cup B'$. We have $\mathsf{wt}_N(T^*) = \sum_{e \in T^*} \mathsf{wt}_N(e)$.

$$\sum_{e \in T^*} \mathsf{wt}_N(e) = \sum_{(a_i, b_j) \in T^*} \big(\mathsf{vote}_a(T^*(a_i), N^*(a_i)) + \mathsf{vote}_b(T^*(b_j), N^*(b_j))\big)$$

$$+ \sum_{(u_k, u_k) \in T^*} \mathsf{vote}_u(u_k, N^*(u_k)) \tag{5}$$

$$= \sum_{u \in A \cup B} \sum_{i=1}^{\mathsf{cap}(u)} \mathsf{vote}_u(T^*(u_i), N^*(u_i)) \tag{6}$$

$$= -\sum_{u \in A \cup B} \Delta_u(N, T) \tag{7}$$
$$= -\Delta(N, T). \tag{8}$$

We have $\mathsf{wt}_N(a_i, b_j) = \mathsf{vote}_a(b, N^*(a_i)) + \mathsf{vote}_b(a, N^*(b_j))$ from the definition of edge weights in G'_N. By grouping together for each vertex u, the edges $(u_i, v_j) \in T^*$ for all partners v of u in T and any possible (u_k, u_k) edges, we get the right side of Eqn. (6).

Crucially, Eqn. (7) follows from how we constructed the matching T^*: for each vertex u, we have $\sum_i \mathsf{vote}_u(N^*(u_i), T^*(u_i)) = \Delta_u(N, T)$ and so $\sum_i \mathsf{vote}_u(T^*(u_i), N^*(u_i)) = -\Delta_u(N, T)$. The total sum of all the terms $\Delta_u(N, T)$ for $u \in A' \cup B'$ is $\Delta(N, T)$. Thus it follows that $\mathsf{wt}_N(T^*) = -\Delta(N, T)$ and hence N is a popular matching. □

We now apply the above theorem to show that every pairwise-stable matching in G is also a popular matching.

Corollary 1 *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 $\mathsf{wt}_S(e) \le 0$. Thus every matching in G'_S has weight at most 0 and so by Theorem 1, we can conclude that S is popular. □

Our goal is to show that M_0 is a max-size popular matching in G. We will do this as follows in Sects. 3.2 and 3.3:

- We will show in Theorem 2 that M_0 satisfies the sufficient condition for popularity given in Theorem 1.
- Lemma 1 will show that no matching larger than M_0 can be a popular matching.

3.2 The Popularity of M_0

We will now use Theorem 1 to prove the popularity of the matching M_0 computed in Sect. 2. As described at the beginning of Sect. 3, we will construct the matchings M'_0, M^*_0 and the graph G'_{M_0} corresponding to the matching M_0. Our goal is to show that every perfect 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).

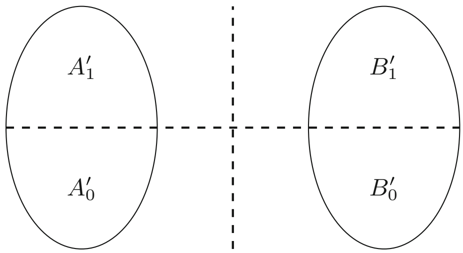


Fig. 1 $A' = A'_0 \cup A'_1$ and $B' = B'_0 \cup B'_1$: all courses b_j left unmatched in M'_0 are in B'_0 and all students a_i left unmatched in M'_0 are in A'_1. Note that $M'_0 \subseteq (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$

Theorem 2 will show that the matching M_0 satisfies the condition of Theorem 1, this will prove that M_0 is a popular matching in G. This proof is inspired by the proof in [26] 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 perfect matching in G'_{M_0} has weight at most 0, we consider the max-weight perfect matching problem in G'_{M_0} as our primal LP. We show a dual feasible solution 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 2 *Every perfect matching in G'_{M_0} has weight at most 0.*

Proof Let our primal LP be the max-weight perfect 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:

$$\begin{aligned} &\max \sum_{e \in G'_{M_0}} \mathsf{wt}_{M_0}(e) \cdot x_e \\ &\text{subject to} \quad \sum_{e \in E'(u_k)} x_e = 1 \qquad \text{for all } u_k \in A' \cup B', \\ &\qquad\qquad\qquad\qquad x_e \geq 0 \qquad \text{for all edges } e \in G'_{M_0}. \end{aligned}$$

Here $E'(u_k)$ 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 $u_k \in A' \cup B'$.

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

$$\text{subject to} \quad \alpha_{a_i} + \alpha_{b_j} \geq \mathsf{wt}_{M_0}(a_i, b_j) \qquad \text{for all edges } (a_i, b_j) \in G'_{M_0}, \tag{9}$$

$$\alpha_{u_k} \geq \mathsf{wt}_{M_0}(u_k, u_k) \qquad \text{for all } u_k \in A' \cup B'. \tag{10}$$

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:

$$\begin{aligned} \alpha_{u_k} &= 1 \quad \text{if } u_k \in A'_0 \cup B'_1 \\ \alpha_{u_k} &= -1 \quad \text{if } u_k \in A'_1 \cup B'_0. \end{aligned}$$

Observe that Inequality (10) holds for all vertices $u_k \in A' \cup B'$. This is because $\alpha_{u_k} = 0 = \mathsf{wt}_{M_0}(u_k, u_k)$ for all u_k unmatched in M'_0; similarly, for all u_k matched in M'_0 we have $\alpha_{u_k} \geq -1 = \mathsf{wt}_{M_0}(u_k, u_k)$. In order to show Inequality (9), we will use Claim 1 stated below.

Claim 1 *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 1 (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 (9) holds for all edges of G'_{M_0} in $A'_1 \times B'_0$.
- Claim 1 (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 (9) 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 (9) is satisfied for all edges of G'_{M_0} in $A'_0 \times B'_1$. Thus Inequality (9) 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 perfect matching in G'_{M_0} has weight at most 0. ◻

Our proof of Claim 1 is similar to its proof in the one-to-one setting from [25].

Proof of Claim 1 Consider any edge $(a_i, b_j) \in A'_1 \times B'_0$ in G'_{M_0}. Note that the matching M_0 does not contain the edge (a, b)—if it did, then G'_{M_0} would have only one copy of this edge, say (a_s, b_t), which being an edge of M'_0, has to be in either $A'_0 \times B'_0$ or $A'_1 \times B'_1$ whereas we are given that $(a_i, b_j) \in A'_1 \times B'_0$. The student $a_i \in A'_1$, i.e., a^1 got activated in our algorithm and recall that every course prefers level 1 neighbors to level 0 neighbors in our algorithm. So if a^1 had proposed to b, then this offer would have been accepted since b had at least one partner who was a level 0 student (since $b_j \in B'_0$). Thus a^1 (with its entire residual capacity) must have been accepted by neighbors that a prefers to b. Hence a_i prefers its partner in M^*_0 to b, so $\mathsf{vote}_a(b, M^*_0(a_i)) = -1$.

Since $(a, b) \notin M_0$ while a^1 got activated in our algorithm, along with the fact that $b_j \in B'_0$, it follows that the student a^0 was rejected by b. When b rejected a^0, the course b was matched to $\mathsf{cap}(b)$ neighbors, each of which was preferred by b to a^0. Thereafter, b may have received (and accepted) better offers from its neighbors and since $b_j \in B'_0$, the course b never received enough offers from level 1 neighbors to have all its partners as level 1 students. In particular, b_j is matched to a level 0 neighbor that is preferred to a^0. Thus b_j prefers its neighbor in M^*_0 to a, so $\mathsf{vote}_b(a, M^*_0(b_j)) = -1$. So it follows that $\mathsf{wt}_{M_0}(a_i, b_j) = -2$.

We will now show part (ii) of this lemma. In our algorithm, the preference order of each vertex, when restricted to level 0 neighbors, is its original preference order and similarly, its preference order when restricted to level 1 neighbors, is its original preference order. Thus for each edge (a_i, b_j) in G'_{M_0} where $(a_i, b_j) \in (A'_0 \times B'_0) \cup (A'_1 \times B'_1)$, either (1) the vertex b prefers $M'_0(b_j)$ to a or the vertex a prefers $M'_0(a_i)$ to b or (2) $(a_i, b_j) \in M'_0$. In both cases, we have $\mathsf{wt}_{M_0}(a_i, b_j) \le 0$. □

3.3 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 it follows quite easily from the proof of Theorem 2 that M is a max-size *weakly popular* matching in G. Since M is a popular matching (by Theorems 1 and 2), it would immediately follow that M is a max-size popular matching in G.

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 in the proof of Theorem 1. Thus T^* is a perfect matching in G'_{M_0} and $\mathsf{wt}_{M_0}(T^*) = -\Delta(M_0, T)$.

We know from Theorem 2 that $\mathsf{wt}_{M_0}(T^*) \le 0$. Suppose T is a weakly popular matching in G. Then $\mathsf{wt}_{M_0}(T^*)$ has to be 0, otherwise the "weak 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 weakly popular matching in G, then T^* is an optimal solution to the maximum weight perfect matching problem in G'_{M_0}. Recall that this is the primal LP in the proof of Theorem 2. We will use the dual feasible solution that we constructed in the proof of Theorem 2 and apply complementary slackness to show that if $(u_k, u_k) \in M^*_0$, i.e., if u_k is left unmatched in M'_0, then T^* also has to contain (u_k, u_k).

Lemma 1 *Let T be a weakly 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: $(u_k, u_k) \in M^*_0$ implies $(u_k, u_k) \in T^*$.*

Proof Consider the α-values assigned to vertices in $A' \cup B'$ in the proof of Theorem 2. This is an *optimal dual* solution since its value is 0 which is the value of the optimal primal solution since $\mathsf{wt}_{M_0}(M^*_0) = 0$. 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:

$$\text{either} \quad \alpha_{u_k} + \alpha_{v_t} = \mathsf{wt}_{M_0}(u_k, v_t) \quad \text{or} \quad T^*_{(u_k, v_t)} = 0. \tag{11}$$

Let $u_k \in A' \cup B'$ be a vertex such that $(u_k, u_k) \in M^*_0$, 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 non-trivial neighbor v_t of u_k, we have $\mathsf{vote}_u(v, u_k) = 1$ and so $\mathsf{wt}_{M_0}(u_k, v_t) \ge 0$. Claim 1 (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 1 (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

$\mathsf{wt}_{M_0}(u_k, v_t) = \mathsf{vote}_u(v, u_k) + \mathsf{vote}_v(u, M_0'(v_t)) = 1 - 1 = 0$. Thus it follows from Equation (11) that $T^*_{(u_k,v_t)} = 0$ for any non-trivial neighbor v_t. Since T^* has to match all in $A' \cup B'$, we have $(u_k, u_k) \in T^*$. $\square$

Let T be any weakly popular matching in G. Consider the matching $T' = T^* \setminus \{(u_k, u_k) : u_k \in A' \cup B'\}$. Lemma 1 implies that $|T'| \leq |M_0'|$ 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 $|T| \leq |M_0|$. As this holds for any weakly popular matching T in G, we can conclude that M_0 is a max-size weakly popular matching in G. Since M_0 is a popular matching in G (by Theorems 1 and 2), we can conclude that M_0 is a max-size popular matching in G.

Interestingly, Lemma 1 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 is a max-size maximal element of both $\succsim_p$ and $\succsim_{wp}$. 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 1 *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.*

3.4 Running Time of Our Algorithm

It is known that the Gale–Shapley algorithm for the hospitals/residents problem (and thus in the many-to-many setting) can be implemented to run in linear time [31]. Our algorithm is very similar to the Gale–Shapley algorithm in the many-to-many setting and we describe a simple linear time implementation of our algorithm.

Recall that our algorithm runs in the graph H whose vertex set is $A'' \cup B$. For each vertex $u \in A'' \cup B$, right at the beginning of the algorithm, we will construct an array D_u that contains all neighbors of u in decreasing order of preference. More precisely, D_u is an array of length $\deg_H(u)$ where $D_u[i]$ contains the identity of the i-th ranked neighbor of u. For $a \in A''$ and $1 \leq i \leq \deg_H(a)$, the cell $D_a[i]$ also stores the value $\mathsf{rank}_b(a)$, where b is a's i-th ranked neighbor in H and $\mathsf{rank}_b(a)$ is the rank of a in b's preference order in H.

It is easy to see that all the arrays D_u can be constructed in $O(m + n)$ time, where $m = |E|$ and $n = |A| + |B|$. We will maintain the function $\mathsf{MaxRank}(\cdot)$ defined below

[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 $M_0 \sim M_1$.

for each $b \in B$:

$$\mathsf{MaxRank}(b) = \begin{cases} \text{rank of } b\text{'s worst neighbor in } M & \text{if } b \text{ is fully matched in } M, \\ \infty & \text{otherwise.} \end{cases}$$

For any $a \in A''$, before a proposes to a neighbor b, the vertex a first checks if $\mathsf{rank}_b(a) < \mathsf{MaxRank}(b)$ and goes ahead with the proposal only if this condition is satisfied. If this condition is not satisfied, i.e., if b ranks a worse than $\mathsf{MaxRank}(b)$, then the edge (a, b) is actually missing in the current graph H and so a does not propose to b.

For any $b \in B$, when b receives a's proposal (we assume that a's proposal also contains the value $j = \mathsf{rank}_b(a)$), b accepts this proposal by setting a flag in the cell $D_b[j]$ to *true*. Thus for each $b \in B$ and $1 \le j \le \deg_H(b)$, along with the identity a of b's j-th ranked neighbor in H, the cell $D_b[j]$ also contains a flag set to *true* if $(a, b) \in M$, else the flag is set to *false*. Thus the entire matching M is stored via these flags in the arrays D_b for $b \in B$.

If $|M(b)| > \mathsf{cap}(b)$ after b accepts a's proposal, then we set the flag in the cell $D_b[\mathsf{MaxRank}(b)]$ to *false*. Also, $\mathsf{MaxRank}(b)$ needs to be updated and this is done by using a pointer to traverse the array D_b leftward from its current location to find the rightmost cell whose flag is set to *true*.

Consider the step where we check if b is already matched to u^0 when u^1 proposes to b. This check can be easily performed by comparing u^0's rank in b's preference order with $\mathsf{MaxRank}(b)$. Note that u^0's rank in b's preference order is $\mathsf{rank}_b(u^1) + \deg_G(b)$. If $(u^0, b) \in M$, then we need to replace (u^0, b) in M with (u^1, b)—this step is implemented by setting the flag in $D_b[\mathsf{rank}_b(u^0)]$ to *false* and the flag in $D_b[\mathsf{rank}_b(u^1)]$ to *true*, so this takes unit time.

It is easy to see that our entire algorithm can be implemented to run in linear time. Hence we can conclude the following theorem.

Theorem 3 *A max-size popular matching in a many-to-many instance $G = (A \cup B, E)$ can be computed in linear time.*

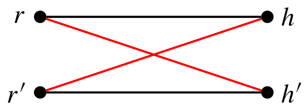
3.5 The Rural Hospitals Theorem for Max-Size Popular Matchings

The strong version of the rural hospitals theorem for stable matchings [36] does not necessarily hold for max-size popular matchings. The strong version of the rural hospitals theorem for max-size popular matchings would say that a hospital that is not matched up to capacity in some max-size popular matching has to be 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 $M = \{(r, h), (r', h')\}$ and $M' = \{(r, h'), (r', h)\}$. 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.

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

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



However the weaker version of the rural hospitals theorem holds here, i.e., every max-size popular matching has to match the same set of vertices. Such a result for max-size popular matchings in the one-to-one setting was shown in [18]. Our proof is based on linear programming and is different from the combinatorial proof in [18].

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

Proof Consider the realization T^* of T such that T^* is a perfect matching in G'_{M_0} and $\mathsf{wt}_{M_0}(T^*) = -\Delta(M_0, T)$. Since T is popular, we know that T^* has to include all the edges (u_k, u_k) for vertices u_k left unmatched in M'_0 (by Lemma 1). Let $T' = T^* \setminus \{(u_k, u_k) : u_k \in A' \cup B'\}$. Every vertex in G'_{M_0} that is unmatched in M'_0 is left unmatched in T' also. We also have $|M'_0| = |M_0| = |T| = |T'|$ as both M_0 and T are max-size popular matchings in G. Hence T' and M'_0 match the same vertices in G'_{M_0}, i.e., every vertex v in G is matched in T to the same capacity as it gets matched to in M_0. □

We also show the following results here: Lemma 3 shows that a pairwise-stable matching is a min-size popular matching in G and Lemma 4 bounds the size of M_0, where M_0 is a max-size popular matching in G. The proofs of Lemmas 3 and 4 are inspired by analogous proofs in the one-to-one setting shown in [20] and in [25], respectively.

Lemma 3 *A pairwise-stable matching S is a min-size weakly popular matching in G.*

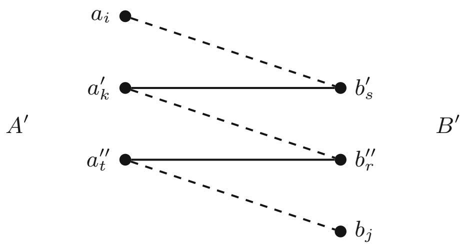
Proof Let T be a matching in G such that $|T| < |S|$. Consider a realization T^* of the matching T in the graph G'_S as described in the proof of Theorem 1 such that T^* is a perfect matching and $\mathsf{wt}_S(T^*) = -\Delta(S, T)$. Recall that S is a pairwise-stable matching in G. Hence for each edge e in G'_S, we have $\mathsf{wt}_S(e) \leq 0$. Moreover, because $|T| < |S|$, there is a vertex u_i that is matched to a non-trivial neighbor in S, however T^* contains the self-loop $e = (u_i, u_i)$. We have $\mathsf{wt}_S(e) = -1$. Thus $\mathsf{wt}_S(T^*) < 0$.

In other words, $\Delta(S, T) > 0$ and T cannot be weakly popular. Since S is a popular matching in G, it means that S is a min-size popular matching in G, in fact, S is a min-size weakly popular matching in G. □

Lemma 4 $|M_0| \geq \frac{2}{3}|M_{\max}|$, *where M_{max} is a max-size matching in G.*

Proof The size of the matching M_0 is exactly the same as that of M'_0 (see the beginning of Sect. 3). Consider the graph G'_{M_0} without self-loops (call this graph G''_{M_0}). We will show that every augmenting path with respect to M'_0 here has length at least 5. This will immediately imply that $|M'_0| \geq 2c/3$ where c is the size of a max-size matching in G''_{M_0}. The value $c \geq |M_{\max}|$ since corresponding to any matching T in G, we have

Fig. 2 An augmenting path with respect to M'_0 in the graph G''_{M_0}: the vertices a_i, a'_k have to belong to A'_1 while the vertices b_j, b''_r have to belong to B'_0. Thus the length of this path is ≥ 5



a matching T' in G''_{M_0} such that $|T| = |T'|$. Thus the size of a max-size matching in G''_{M_0} is at least $|M_{\max}|$ and so we get $|M_0| \geq 2|M_{\max}|/3$.

Consider any augmenting path p with respect to M'_0 in the graph G''_{M_0} (see Fig. 2) and let the endpoints of p be a_i and b_j. Since these vertices are left unmatched in M'_0, it follows that $a_i \in A'_1$ and $b_j \in B'_0$. As seen in the proof of Lemma 1, the vertex a_i is adjacent only to vertices in B'_1 in the graph G''_{M_0} and the unmatched vertex b_j is adjacent only to vertices in A'_0 in the graph G''_{M_0}. Every vertex in A'_0 is matched in M_0 to a neighbor in B'_0 and every vertex in B'_1 is matched in M_0 to a neighbor in A'_1. Thus the shortest augmenting path with respect to M'_0 has the following structure with respect to the sets in $\{A'_0, A'_1, B'_0, B'_1\}$ that its vertices belong to: A'_1-B'_1-A'_1-B'_0-A'_0-B'_0, i.e., its length is at least 5. $\square$

4 The Max-Size Popular Matching Problem in a Roommates Instance

In this section we consider the max-size popular matching problem in a roommates instance $G = (V, E)$. Here every vertex $u \in V$ has a strict preference list ranking its neighbors and $\mathsf{cap}(u) = 1$ for all vertices u here. Not every roommates instance admits a popular matching; for instance, the triangle on vertices a, b, c described in Sect. 1 has no popular matching.

We will assume that our input instance G admits a stable matching. The proof of popularity of a stable matching for the bipartite case [14] holds here as well, so there exists at least one popular matching in our roommates instance G. Our problem is to compute a *max-size* popular matching in G. In this section we prove this problem is NP-hard.

We will show a reduction from the vertex cover problem to prove the hardness of the above problem. Let $H = (V_H, E_H)$ be an instance of the vertex cover problem and let $V_H = \{1, \ldots, n_H\}$, i.e., $V_H = [n_H]$. We will build a roommates instance G as follows: (see Fig. 3)

- Corresponding to every vertex $i \in V_H$, there will be 4 vertices a_i, b_i, c_i, d_i in G, and
- Corresponding to every edge $e = (i, j) \in E_H$, there will be 2 vertices u^e_i and u^e_j in G.

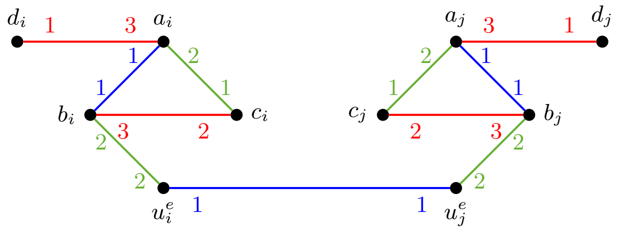


Fig. 3 The graph G restricted to the adjacent vertices i and j in H: the vertices a_t, b_t, c_t, d_t in G correspond to vertex $t \in \{i, j\}$ in H and the vertices u_i^e and u_j^e in G correspond to the edge $e = (i, j)$ in H. Vertex preferences in G are indicated on the edges

The preferences of the vertices a_i, b_i, c_i, d_i are as follows, where $e_1, \ldots, e_k$ are all the edges in H with vertex i as an endpoint.

$$a_i: \ b_i \succ c_i \succ d_i \qquad b_i: \ a_i \succ u_i^{e_1} \cdots \succ u_i^{e_k} \succ c_i \qquad c_i: \ a_i \succ b_i \qquad d_i: \ a_i.$$

The order among the vertices $u_i^{e_1}, \ldots, u_i^{e_k}$ in the preference list of b_i is arbitrary. The preference list of vertex u_i^e is $u_j^e \succ b_i$, where $e = (i, j)$ (see Fig. 3).

Observe that G admits a stable matching $S = \{(a_i, b_i) : 1 \le i \le n_H\} \cup \{(u_i^e, u_j^e) : e = (i, j) \in E_H\}$.

Lemma 5 *Let M be a popular matching in G.*

1. *For any $i \in [n_H]$, either $(a_i, b_i) \in M$ or $\{(a_i, d_i), (b_i, c_i)\} \subseteq M$.*
2. *The set $U = \{i \in [n_H] : (a_i, b_i) \in M\}$ is a vertex cover of H.*

Proof The vertex a_i is the top choice neighbor of all its neighbors b_i, c_i, d_i. Thus a_i has to be matched in the popular matching M.

1. If a_i is matched to b_i then $(a_i, b_i) \in M$.
2. If $(a_i, d_i) \in M$, then c_i also has to be matched—otherwise we get a more popular matching by replacing the edge (a_i, d_i) with (a_i, c_i). Since c_i has degree 2, it has to be the case that $\{(a_i, d_i), (b_i, c_i)\} \subseteq M$.
3. Suppose $(a_i, c_i) \in M$. Since a_i prefers b_i to c_i, this means that b_i has to be matched in M. Other than a_i and c_i (which are matched to each other), b_i's neighbors are u_i^e for all edges e incident to i in H. So $(b_i, u_i^e) \in M$ for some $e = (i, j) \in E_H$. We will now construct a new matching M' as follows:
   - Replace the edges (a_i, c_i), (b_i, u_i^e), $(u_j^e, M(u_j^e))$ in M with the edges (a_i, b_i) and (u_i^e, u_j^e). So c_i and $M(u_j^e)$ are unmatched in M', hence these 2 vertices prefer M to M'; however the 4 vertices a_i, b_i, u_i^e, u_j^e prefer M' to M. Thus M' is more popular than M, a contradiction to the popularity of M. Hence $(a_i, c_i) \notin M$.

We now show the second part of the lemma. Let $(i, j) \in E_H$. We need to show that either $(a_i, b_i) \in M$ or $(a_j, b_j) \in M$. Suppose not. Then by the first part of this lemma, (a_i, d_i), (b_i, c_i) are in M and similarly, (a_j, d_j), (b_j, c_j) are in M. Also $(u_i^e, u_j^e) \in M$.

Consider the matching M' obtained by replacing the 5 edges (a_i, d_i), (b_i, c_i), (a_j, d_j), (b_j, c_j), and (u_i^e, u_j^e) in M with the 4 edges (a_i, c_i), (b_i, u_i^e), (a_j, c_j), and (b_j, u_j^e) (see Fig. 3). Among the 10 vertices involved here, the 6 vertices a_i, b_i, c_i and a_j, b_j, c_j prefer M' to M while the 4 vertices d_i, u_i^e and d_j, u_j^e prefer M to M'. Thus M' is more popular than M, a contradiction to the popularity of M.

Hence for each edge $(i, j) \in E_H$, either $(a_i, b_i) \in M$ or $(a_j, b_j) \in M$. In other words, the set $U = \{i \in [n_H] : (a_i, b_i) \in M\}$ is a vertex cover of H. □

The following theorem uses the above lemma to show a polynomial time reduction from the vertex cover problem in H to the max-size popular matching problem in G.

Theorem 4 *For any $1 \leq k \leq n_H$, the graph $H = (V_H, E_H)$ admits a vertex cover of size k if and only if G has a popular matching of size at least $m_H + 2n_H - k$, where $|E_H| = m_H$.*

A sufficient condition for popularity In order to prove Theorem 4, we will generalize the linear programming approach used in Sect. 3 to prove the popularity of a matching in a roommates instance. Given a matching N in $G = (V, E)$, the graph $G'_N = (V, E')$ is the graph G augmented with self-loops. Corresponding to any matching N in G, we have the perfect matching $N^* = N \cup \{(u, u) : u \text{ is unmatched in } N\}$ in G'_N.

The weight function wt_N on the edge set of G'_N is exactly the same as before. It is easy to see from Eqs. (5)–(8) that for any matching T in G, we have $\mathsf{wt}_N(T^*) = -\Delta(N, T)$. Thus if a max-weight perfect matching in G'_N has weight at most 0 then N is a popular matching. The max-weight perfect matching LP in G'_N is described below.

$$\max \sum_{e \in E'} \mathsf{wt}_N(e) \cdot x_e \qquad \text{(LP1)}$$

$$\begin{aligned} \text{subject to} \quad \sum_{e \in E'(u)} x_e &= 1 && \text{for all vertices } u \in V \\ \sum_{e \in E[S]} x_e &\leq \lfloor |S|/2 \rfloor && \text{for all odd } S \subseteq V,\ |S| \geq 3 \\ x_e &\geq 0 && \text{for all edges } e \in E'. \end{aligned}$$

As before, $E'(u)$ is the set of edges incident to u in G'_N. Also $E[S]$ is the set of all $(u, v) \in E$ such that $u, v \in S$. Note that $E[S]$ does not contain any self-loops.

The dual LP (see LP2) will be useful. Here Ω is the collection of all odd subsets S of V of size at least 3. If the optimal value of LP2 is at most 0 then the optimal value of LP1 is at most 0 (by weak duality) and hence $-\Delta(N, T) \leq 0$ or $\Delta(N, T) \geq 0$ for all matchings T. In other words, N is a popular matching in G.

$$\min \sum_{u \in V} \alpha_u + \sum_{S \in \Omega} \lfloor |S|/2 \rfloor \cdot z_S \qquad \text{(LP2)}$$

$$\text{subject to} \quad \alpha_u + \alpha_v + \sum_{\substack{S \in \Omega \\ u, v \in S}} z_S \geq \mathsf{wt}_N(u, v) \qquad \text{for all edges } (u, v) \in E$$

$$\begin{aligned} \alpha_u &\geq \mathsf{wt}_N(u,u) && \text{for all } u \in V \\ z_S &\geq 0 && \text{for all } S \in \Omega. \end{aligned}$$

Proof of Theorem 4 Suppose $H = (V_H, E_H)$ admits a vertex cover U of size k. We will now build a popular matching N in G of size $m_H + 2n_H - k$.

- Add all edges (u_i^e, u_j^e) in G to N.
- For every $i \in U$, add the edge (a_i, b_i) to N.
- For every $i \notin U$, add the edges (a_i, d_i), (b_i, c_i) to N.

The size of N is $m_H + |U| + 2(n_H - |U|) = m_H + 2n_H - k$. We will prove N is popular by showing a solution $(\boldsymbol{\alpha}, \mathbf{z})$ to LP2 such that $\sum_u \alpha_u + \sum_{S\in\Omega} \lfloor |S|/2 \rfloor \cdot z_S = 0$. To begin with, initialize $z_S = 0$ for all $S \in \Omega$.

- For every $i \in U$: set $\alpha_{a_i} = \alpha_{b_i} = \alpha_{c_i} = \alpha_{d_i} = 0$.
- For every $i \notin U$: set $\alpha_{a_i} = 1$ and $\alpha_{b_i} = \alpha_{c_i} = \alpha_{d_i} = -1$; also set $z_{S_i} = 2$ where $S_i = \{a_i, b_i, c_i\}$.
- For every edge $e = (i, j) \in E_H$ do:
  - If both i and j are in U then arbitrarily set either (i) $\alpha_{u_i^e} = 1$ and $\alpha_{u_j^e} = -1$ or (ii) $\alpha_{u_i^e} = -1$ and $\alpha_{u_j^e} = 1$, i.e., $\{\alpha_{u_i^e}, \alpha_{u_j^e}\} = \{-1, 1\}$.
  - Else if $i \notin U$ then set $\alpha_{u_i^e} = 1$ and $\alpha_{u_j^e} = -1$;
  - Else (so $j \notin U$) set $\alpha_{u_i^e} = -1$ and $\alpha_{u_j^e} = 1$.

It is easy to check that the above setting of $(\boldsymbol{\alpha}, \mathbf{z})$ is a feasible solution to LP2. In particular, for $i \in U$, we have $\mathsf{wt}_N(b_i, u_i^e) = -1 - 1 = -2$ while $\alpha_{b_i} = 0$ and $\alpha_{u_i^e} \in \{\pm 1\}$, thus $\alpha_{b_i} + \alpha_{u_i^e} \geq \mathsf{wt}_N(b_i, u_i^e)$; also $\{\alpha_{u_i^e}, \alpha_{u_j^e}\} = \{-1, 1\}$, thus $\alpha_{u_i^e} + \alpha_{u_j^e} = 0 = \mathsf{wt}_N(u_i^e, u_j^e)$. Similarly, when $i \notin U$, $\mathsf{wt}_N(b_i, u_i^e) = 1 - 1 = 0$ and we have $\alpha_{b_i} = -1$ and $\alpha_{u_i^e} = 1$ here; also $\alpha_{u_i^e} = 1$ and $\alpha_{u_j^e} = -1$, thus $\alpha_{u_i^e} + \alpha_{u_j^e} = 0 = \mathsf{wt}_N(u_i^e, u_j^e)$ again. Moreover,

$$\sum_{v\in V} \alpha_v \; + \sum_{S\in\Omega} \lfloor |S|/2 \rfloor \cdot z_S \;=\; \sum_{i\notin U} -2 \; + \sum_{i\notin U} 2 \;=\; 0. \tag{12}$$

This is because $\alpha_v = 0$ for all vertices v unmatched in N and $\alpha_u + \alpha_v = 0$ for all edges $(u, v) \in N$ except the edges (b_i, c_i) where $i \notin U$. For each $i \notin U$, we have $\alpha_{b_i} + \alpha_{c_i} = -2$ and we also have $z_{S_i} = 2$ where $S_i = \{a_i, b_i, c_i\}$. The sum in Equation (12) is 0 and so N is a popular matching. Hence G has a popular matching of size $m_H + 2n_H - k$.

We will now show the converse. Let M be a popular matching in G of size at least $m_H + 2n_H - k$. We know that $U = \{i \in [n_H] : (a_i, b_i) \in M\}$ is a vertex cover of H (by Lemma 5, part 2). We will show that $|U| \leq k$.

It follows from part 1 of Lemma 5 that all edges (u_i^e, u_j^e) belong to every popular matching. So these account for m_H many edges in M. We also know that for every $i \in V_H$, either $(a_i, b_i) \in M$ or $\{(a_i, d_i), (b_i, c_i)\} \subseteq M$. Thus $|M| = m_H + |U| + 2(n_H - |U|)$. Since $|M| \geq m_H + 2n_H - k$, it follows that $|U| \leq k$. Thus the graph G has a vertex cover of size k. □

We can now conclude the following theorem.

Theorem 5 *Let $G = (V, E)$ be a roommates instance with strict preference lists that admits a stable matching. The max-size popular matching problem in G is* NP*-hard.*

Remark The rural hospitals theorem [16] for stable matchings in a roommates instance G says that every stable matching in G matches the same subset of vertices. Such a statement is not true for max-size popular matchings in a roommates instance as seen in the instance in Fig. 3. This instance has two max-size popular matchings (these are of size 4): one leaves c_i and d_i unmatched while another leaves c_j and d_j unmatched.

5 Conclusions

We considered the max-size popular matching problem in a bipartite instance $G = (A \cup B, E)$ where vertex u has a capacity $\mathsf{cap}(u)$, for each $u \in A \cup B$. We generalized the notion of popularity from the one-to-one setting to define *popular* and *weakly popular* in the many-to-many setting. We showed a linear time algorithm to compute a max-size popular matching here and showed that it is also a max-size weakly popular matching. We then considered the max-size popular matching problem in a roommates instance $G = (V, E)$ that admits stable matchings and proved this problem to be NP-hard.

Acknowledgements We thank the reviewers for their helpful comments. 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.

Appendix: An Interesting Example

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 $v_1 \succ_{u_k} \cdots \succ_{u_k} v_{\mathsf{cap}(v)}$, for each vertex u_k in G'.

Given any matching $\tilde{M}$ in G', we define $\mathsf{proj}(\tilde{M})$ as the projection of $\tilde{M}$, which is the matching obtained by dropping the subscripts of all vertices. Observe that $\mathsf{proj}(\tilde{M})$ obeys all capacity bounds in G. We will now consider the *many-to-one* or the hospitals/residents setting: so there are no multi-edges in $\mathsf{proj}(\tilde{M})$. 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'.

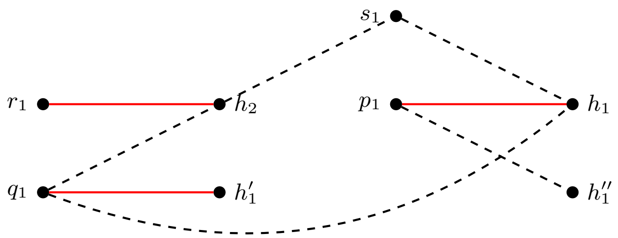


Fig. 4 The edges of the matching N' are bold and the non-matching edges in G'_N are dashed. For simplicity, we have not included self-loops here. Note that the edge (q_1, h_2) is a blocking edge to N' as both q_1 and h_2 prefer each other to their respective partners in N', i.e., $\mathsf{wt}_N(q_1, h_2) = 2$ and $\mathsf{wt}_N(e) = 0$ for all other edges e in G'_N

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:

$$
\begin{array}{ll}
p: h, h'' & h: p, q, r, s \\
q: h, h' & h': q \\
r: h & h'': p \\
s: h &
\end{array}
$$

Consider the matching $N = \{(p, h), (q, h'), (r, h)\}$. Claim 2 below shows that N is a popular matching. Our proof of popularity of N will use the sufficient condition for popularity shown in Theorem 1 and follows the approach used in Theorem 2.

Claim 2 *The matching N is popular in G.*

Proof We will use the notation introduced at the beginning of Sect. 3. So we have $R' = \{p_1, q_1, r_1, s_1\}$ and $H' = \{h_1, h_2, h'_1, h''_1\}$. Let $N' = \{(p_1, h_1), (q_1, h'_1), (r_1, h_2)\}$ (see Fig. 4).

We need to show that every perfect 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 max-weight perfect 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. $\square$

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 3 *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'. Hence neither N_1 nor N_2 is popular in G'. □

Note that the above instance can easily be transformed to another instance G with a *max-size* popular matching that cannot be realized as a popular matching in G'. In fact, it is easy to show that every popular matching in G' gets projected to a popular matching in G. However as illustrated by the example above, there may exist popular matchings in G that cannot be realized as popular matchings in G'.

References

1. Abraham, D.J., Irving, R.W., Kavitha, T., Mehlhorn, K.: Popular matchings. SIAM J. Comput. **37**(4), 1030–1045 (2007)
2. Askalidis, G., Immorlica, N., Kwanashie, A., Manlove, D., Pountourakis, E.: Socially Stable matchings in the hospitals/residents problem. In: the 13th International Symposium on Algorithms and Data Structures (WADS), pp. 85–96 (2013)
3. Blair, C.: The lattice structure of the set of stable matchings with multiple partners. Math. Oper. Res. **13**, 619–628 (1988)
4. Biro, P., Irving, R.W., Manlove, D.F.: Popular matchings in the marriage and roommates problems. In: the 7th International Conference in Algorithms and Complexity (CIAC), pp. 97–108 (2010)
5. Brandl, F., Kavitha, T.: Popular matchings with multiple partners. In: the 37th Foundations of Software Technology and Theoretical Computer Science (FSTTCS) (2017)
6. Canadian Resident Matching Service. How the matching algorithm works. Web document available at http://carms.ca/algorithm.htm
7. Condorcet, J.A.N.C.: Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Imprimerie Royale, Paris (1785)
8. Cseh, Á.: Popular matchings. In: Trends in Computational Social Choice, Edited by Ulle Endriss, COST (European Cooperation in Science and Technology), pp. 105–122 (2017)
9. Cseh, Á., Huang, C.-C., Kavitha, T.: Popular matchings with two-sided preferences and one-sided ties. In: the 42nd International Colloquium on Automata, Languages, and Programming (ICALP): Part I, pp. 367–379 (2015)
10. Cseh, Á., Kavitha, T.: Popular edges and dominant matchings. In: the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO), pp. 138–151 (2016)
11. Faenza, Y., Kavitha, T., Powers, V., Zhang, X.: Popular matchings and limits to tractability. In: The Conference is the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, pp. 2790–2809 (2019)
12. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Month. **69**(1), 9–15 (1962)
13. Gale, D., Sotomayor, M.: Some remarks on the stable matching problem. Discrete Appl. Math. **11**(3), 223–232 (1985)
14. Gärdenfors, P.: Match making: assignments based on bilateral preferences. Behav. Sci. **20**(3), 166–173 (1975)
15. Gupta, S., Misra, P., Saurabh, S., Zehavi, M.: Popular matching in roommates setting is NP-hard. The Conference is the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, pp. 2810–2822 (2019)
16. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Boston (1989)
17. Hamada, K., Iwama, K., Miyazaki, S.: The hospitals/residents problem with lower quotas. Algorithmica **74**(1), 440–465 (2016)

18. Hirakawa, M., Yamauchi, Y., Kijima, S., Yamashita, M.: 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)
19. Huang, C.-C.: Classified stable matching. In: the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1235–1253 (2010)
20. Huang, C.-C., Kavitha, T.: Popular matchings in the stable marriage problem. Inf. Comput. **222**, 180–194 (2013)
21. Huang, C.-C., Kavitha, T.: Popularity, self-duality, and mixed matchings. In: the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 2294–2310 (2017)
22. Irving, R.W.: An efficient algorithm for the stable roommates problem. J. Algorithms **6**, 577–595 (1985)
23. Irving, R.W., Manlove, D.F., Scott, S.: The hospitals/residents problem with ties. In: the 7th Scandinavian Workshop on Algorithm Theory (SWAT), pp. 259–271 (2000)
24. Irving, R.W., Manlove, D.F., Scott, S.: Strong stability in the hospitals/residents problem. In the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pp. 439–450 (2003)
25. Kavitha, T.: A size-popularity tradeoff in the stable marriage problem. SIAM J. Comput. **43**(1), 52–71 (2014)
26. Kavitha, T.: Popular half-integral matchings. In: the 43rd International Colloquium on Automata, Languages, and Programming (ICALP), pp. 22.1–22.13 (2016)
27. Kavitha, T.: Max-size popular matchings and extensions. http://arxiv.org/pdf/1802.07440.pdf
28. Kavitha, T., Mestre, J., Nasre, M.: Popular mixed matchings. Theor. Comput. Sci. **412**(24), 2679–2690 (2011)
29. Király, T., Mészáros-Karkus, Zs: Finding strongly popular b-matchings in bipartite graphs. Electron. Notes Discrete Math. **61**, 735–741 (2017)
30. Manlove, D.F.: Algorithmics of Matching Under Preferences. World Scientific Publishing Company Incorporated, Singapore (2013)
31. Manlove, D.F.: The Hospitals/Residents Problem. Encyclopedia of Algorithms, pp. 926–930. Springer, Berlin (2015)
32. Nasre, M., Rawat, A.: Popularity in the generalized hospital residents setting. In: the 12th International Computer Science Symposium in Russia (CSR), pp. 245–259 (2017)
33. National Resident Matching Program. Why the Match? Web document available at http://www.nrmp.org/whythematch.pdf
34. Roth, A.E.: The evolution of the labor market for medical interns and residents: a case study in game theory. J. Polit. Econ. **92**(6), 991–1016 (1984)
35. Roth, A.E.: Stability and polarization of interest in job matching. Econometrica **53**, 47–57 (1984)
36. Roth, A.E.: On the allocation of residents to rural hospitals: a general property of two-sided matching markets. Econometrica **54**(2), 425–427 (1986)
37. Sotomayor, M.: Three remarks on the many-to-many stable matching problem. Math. Soc. Sci. **38**, 55–70 (1999)
38. Subramanian, A.: A new approach to stable matching problems. SIAM J. Comput. **23**(4), 671–700 (1994)
39. Teo, C.-P., Sethuraman, J.: The geometry of fractional stable matchings and its applications. Math. Oper. Res. **23**(4), 874–891 (1998)