Popular Matchings with One-Sided Bias

Telikepalli Kavitha
Tata Institute of Fundamental Research, Mumbai, India
kavitha@tifr.res.in

Abstract

Let $G = (A \cup B, E)$ be a bipartite graph where A consists of agents or *main players* and B consists of jobs or *secondary players*. Every vertex has a strict ranking of its neighbors. A matching M is *popular* if for any matching N, the number of vertices that prefer M to N is at least the number that prefer N to M. Popular matchings always exist in G since every stable matching is popular.

A matching M is A-*popular* if for any matching N, the number of *agents* (i.e., vertices in A) that prefer M to N is at least the number of agents that prefer N to M. Unlike popular matchings, A-popular matchings need not exist in a given instance G and there is a simple linear time algorithm to decide if G admits an A-popular matching and compute one, if so.

We consider the problem of deciding if G admits a matching that is both popular and A-popular and finding one, if so. We call such matchings *fully popular*. A fully popular matching is useful when A is the more important side – so along with overall popularity, we would like to maintain "popularity within the set A". A fully popular matching is not necessarily a min-size/max-size popular matching and all known polynomial time algorithms for popular matching problems compute either min-size or max-size popular matchings. Here we show a linear time algorithm for the fully popular matching problem, thus our result shows a new tractable subclass of popular matchings.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Bipartite graphs, Stable matchings, Gale-Shapley algorithm, LP-duality

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.70

Category Track A: Algorithms, Complexity and Games

Funding Supported by the DAE, Government of India, under project no. 12-R&D-TFR-5.01-0500.

Acknowledgements Work done at MPI for Informatics, Saarland Informatics Campus, Germany. Thanks to Yuri Faenza for discussions that led to this problem and his helpful comments on the manuscript. Thanks to the reviewers for their valuable suggestions on improving the presentation.

1 Introduction

Let $G = (A \cup B, E)$ be a bipartite graph where vertices in A are called *agents* and those in B are called *jobs*. Every vertex has a strict ranking of its neighbors. Such a graph, also called a *marriage* instance, is a very well-studied model in two-sided matching markets. A matching M in G is *stable* if there is no *blocking pair* with respect to M, i.e., no pair (a, b) such that a and b prefer each other to their respective assignments in M. Gale and Shapley [10] in 1962 showed that stable matchings always exist in G and can be efficiently computed.

Stable matching algorithms have applications in several real-world problems. For instance, stable matchings have been extensively used to match students to schools and colleges [1, 3] and one of the oldest applications here is to match medical residents to hospitals [4, 21]. It is known that all stable matchings in G have the same size [11] and this may only be half the size of a max-size matching in G. Consider the following instance on 4 vertices a_0, a_1, b_0, b_1.

$$a_0 : b_1 \qquad a_1 : b_1 \succ b_0 \qquad b_0 : a_1 \qquad b_1 : a_1 \succ a_0.$$

Here a_1 and b_1 are each other's top choices. There is no edge between a_0, b_0. Note that $M_{\max} = \{(a_0, b_1), (a_1, b_0)\}$ has size 2 while the only stable matching $S = \{(a_1, b_1)\}$ has size 1.

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 70; pp. 70:1–70:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Hence forbidding blocking edges constrains the size of the resulting matching. Rather than empower every edge with a "veto power" to block matchings (this is the notion of stability), we would like to relax stability so that only a *strict majority vote* from the entire vertex set has the power to block matchings. The motivation is to obtain a larger pool of feasible matchings so as to allow better matchings with respect to our objective.

The notion of *popularity* is a natural relaxation of stability that captures the notion of "majority preference". Preferences of a vertex over its neighbors extend naturally to preferences over matchings: consider an election between 2 matchings M and N where vertices are voters. In this M versus N election, every vertex (say, u) votes for the matching in $\{M, N\}$ that it prefers, i.e., where it gets a better assignment (being unmatched is its worst choice), and u abstains from voting if it has the same assignment in both M and N. Let $\phi(M, N)$ (resp., $\phi(N, M)$) be the number of votes for M (resp., N) in this election.

▶ **Definition 1.** *A matching M is* popular *if $\phi(M, N) \geq \phi(N, M)$ $\forall$ matchings N in G.*

So a popular matching never loses a head-to-head election against any matching, i.e., it is a weak *Condorcet winner* [5, 6] in the voting instance where matchings are candidates and vertices are voters. The notion of popularity was introduced by Gärdenfors [12] who showed that every stable matching is popular. So popular matchings always exist in any marriage instance. In fact, every stable matching is a *min-size* popular matching [14] and there are efficient algorithms to compute a *max-size* popular matching [14, 16].

Popular matchings are suitable for applications such as matching students to projects (where students and project advisers have strict preferences) – by relaxing stability to popularity, we can obtain better matchings in terms of size or our desired objective. We consider a natural and relevant objective here: observe that the two sides of $G = (A \cup B, E)$ are *asymmetric* in this application - students are *doers* of the projects, i.e., they are the main or more active players while project advisers are the secondary or more passive players. So along with overall popularity, we would like to maintain "popularity within the set A".

That is, we would like the popular matching that we compute to be popular even when we *only* count the votes of vertices in A, i.e., there should be no matching that is preferred by more vertices in A. Popularity within the set A is the notion of popularity with *one-sided* preferences and we will refer to this as *A-popularity* here. In the M versus N election, let $\phi_A(M, N)$ (resp., $\phi_A(N, M)$) be the number of vertices in A that vote for M (resp., N).

▶ **Definition 2.** *A matching M is A-*popular *if $\phi_A(M, N) \geq \phi_A(N, M)$ $\forall$ matchings N in G.*

A-popular matchings have been well-studied and are relevant in applications such as assigning training posts to applicants [2] and housing allocation schemes [19] where vertices on only one side of the graph have preferences over their neighbors. An A-popular matching need not necessarily exist in a given instance as shown below.

$$a_1 : b_1 \succ b_2 \succ b_3 \qquad\qquad a_2 : b_1 \succ b_2 \succ b_3 \qquad\qquad a_3 : b_1 \succ b_2 \succ b_3.$$

There are 3 agents here and they have identical preferences. It is easy to check that none of the matchings in this instance is A-popular. Let $M_0 = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$ and $M_1 = \{(a_1, b_3), (a_2, b_1), (a_3, b_2)\}$, we have $\phi_A(M_1, M_0) = 2 > 1 = \phi_A(M_0, M_1)$ since a_2, a_3 prefer M_1 to M_0 while a_1 prefers M_0 to M_1. Here we are interested in matchings that are both popular *and* A-popular.

▶ **Definition 3.** *A popular matching M is* fully popular *if M is also A-popular. So for any matching N in G, we have: $\phi(M, N) \geq \phi(N, M)$ and $\phi_A(M, N) \geq \phi_A(N, M)$.*

There may be exponentially many popular matchings in $G = (A \cup B, E)$. So when A is the more important/active side, say it consists of those doing their projects/internships/jobs, it is natural to seek a popular matching that is A-popular as well, i.e., a *fully popular* matching. Thus we seek a matching M such that (1) a majority of the vertices weakly prefer M to any matching, i.e., $\phi(M, N) \geq \phi(N, M)$ for all matchings N, and moreover, (2) a majority of the agents (those in A) weakly prefer M to any matching, i.e., $\phi_A(M, N) \geq \phi_A(N, M)$ for all N.

We show the following result here.

▶ **Theorem 4.** *There is a linear time algorithm to decide if a marriage instance $G = (A \cup B, E)$ with strict preferences admits a fully popular matching or not. If so, our algorithm returns a max-size fully popular matching.*

Another model for popularity in matchings with main players in A and secondary players in B is to scale the votes of those in A by a suitable factor $c \geq 1$ and count the weighted sum of votes in favor of M versus the weighted sum of votes in favor of N in the M versus N election. That is, M is *weighted popular* if $c \cdot \phi_A(M, N) + \phi_B(M, N) \geq c \cdot \phi_A(N, M) + \phi_B(N, M)$ for every matching N, where c is the scaling factor and $\phi_B(\cdot, \cdot)$ is analogous to $\phi_A(\cdot, \cdot)$. No results are currently known for the weighted popular matching problem with $c > 1$ in a marriage instance. Observe that a fully popular matching is a weighted popular matching for every scaling factor $c \geq 1$.

1.1 Background and Related results

The notion of popularity was proposed by Gärdenfors [12] in 1975. Algorithms in the domain of popular matchings were first studied in 2005 for *one-sided* preferences or the A-popular matching problem. Efficient algorithms were given in [2] to decide if a given instance (with ties permitted in preference lists) admits an A-popular matching or not; in particular, a linear time algorithm was given for the case with strict preference lists. An efficient algorithm for the weighted A-popular matching problem, where each agent's vote is scaled by its weight (these weights are a part of the input), was given in [20].

Algorithms for popular matchings in a marriage instance $G = (A \cup B, E)$ or *two-sided* preferences have been well-studied in the last decade. The max-size popular matching algorithms in [14, 16] compute special popular matchings called *dominant* matchings. A linear time algorithm for finding a popular matching with a given edge e was given in [7] (such an edge is called a popular edge). It was shown in [7] that if e is a popular edge then there is either a stable matching or a dominant matching with e. Popular half-integral matchings in $G = (A \cup B, E)$ were characterized in [17] as stable matchings in a larger graph related to G. The popular fractional matching polytope was analyzed in [15] where the half-integrality of this polytope was shown. Other than algorithms for min-size/max-size popular matchings and for popular edge, no other polynomial time algorithms are currently known for finding popular matchings with special properties.

To complete the picture, it was shown in [9] that it is NP-hard to decide if G admits a popular matching that is neither a min-size nor a max-size popular matching. A host of hardness results in [9] painted a bleak picture for efficient algorithms for popular matching problems (other than what is already known). For instance, it is NP-hard to find a popular matching in G with a given pair of edges. Thus finding a max-weight (similarly, min-cost) popular matching is NP-hard when there are weights (resp., costs) on edges.

1.2 Our Result and Techniques

It may be the case that no min-size/max-size popular matching in G is A-popular, however G admits a fully popular matching: Section 2 has such an example. As there are instances where it is NP-hard to decide if there is a popular matching that is neither a min-size nor a max-size popular matching [9], a first guess may be that the fully popular matching problem is NP-hard.

Though an A-popular matching is constrained to use only some special edges in G (see Theorem 5), this does not seem very helpful since it is NP-hard to solve the popular matching problem with forced/forbidden edges [9]. Note that a rival matching (wrt popularity) is free to use any edge in G. It was not known if there was any tractable subclass of popular matchings other than the classes of *stable matchings* [10] and *dominant matchings* [7, 14, 16].

We show the set of fully popular matchings is a new tractable subclass of popular matchings: unlike the classes of stable matchings and dominant matchings which are always non-empty, there need not be a fully popular matching in G. Our algorithm for finding a fully popular matching is based on the classical Gale-Shapley algorithm and works in a new graph H; this graph is essentially two copies of G and is a variant of the graph seen in [17] to study popular *half-integral* matchings. There is a natural map from stable matchings in H to popular half-integral matchings in G. Our goal is to compute a stable matching with sufficient symmetry in H so that we can obtain a popular *integral* matching in G.

We achieve this symmetry by using properties of both popular and A-popular matchings. These properties allow us to identify certain edges that have to be excluded from our matching. If there is no stable matching in H without these edges then we use the lattice structure on stable matchings [13] to show that G has no fully popular matching. Else we obtain a matching M in G from this "partially symmetric" stable matching in H. The most technical part of our analysis is to prove M's popularity in G.

2 Preliminaries

Our input is a bipartite graph $G = (A \cup B, E)$ where every vertex has a strict preference list ranking its neighbors. Let us augment G with self-loops, i.e., each vertex is assumed to be at the bottom of its own preference list.

We will first present the characterization of A-popular matchings in G – note that preferences of vertices in B play no role here. For each $a \in A$, define the vertex $f(a)$ to be a's top choice neighbor and let $s(a)$ be a's most preferred neighbor that is nobody's top choice neighbor. We assume every $a \in A$ has at least one neighbor other than itself, so $f(a) \in B$, however it may be the case that $s(a) = a$. Let $E' = E \cup \{(u, u) : u \in A \cup B\}$.

▶ **Theorem 5** ([2]). *A matching M in $G = (A \cup B, E')$ is A-popular if and only if:*
1. *$M \subseteq \{(a, f(a)), (a, s(a)) : a \in A\}$.*
2. *M matches all in A and all in $\{f(a) : a \in A\}$.*

Popular matchings. We will use an LP-based characterization of popular matchings [17, 18] in a marriage instance $G = (A \cup B, E')$. Observe that any matching in G can be regarded as a perfect matching by including self-loops for all vertices left unmatched. Let M be any perfect matching in G. For any pair of adjacent vertices u, v, let u's vote for v (vs its partner $M(u)$) be 1 if u prefers v to $M(u)$, it is -1 if u prefers $M(u)$ to v, else 0 (i.e., $M(u) = v$). In order to check if M is popular or not in G, the following edge weight function wt_M will be useful. Here $\mathsf{wt}_M(a, b)$ is the sum of votes of a and b for each other.

$$\text{Let } \mathsf{wt}_M(a,b) = \begin{cases} 2 & \text{if } (a,b) \text{ is a blocking edge to } M; \\ -2 & \text{if both } a \text{ and } b \text{ prefer their partners in } M \text{ to each other;} \\ 0 & \text{otherwise.} \end{cases}$$

Thus $\mathsf{wt}_M(a,b) = 0$ for every $(a,b) \in M$. We need to define wt_M for self-loops as well. For any $u \in A \cup B$, let $\mathsf{wt}_M(u,u) = 0$ if $(u,u) \in M$, else $\mathsf{wt}_M(u,u) = -1$. For any perfect matching N in G, observe that $\mathsf{wt}_M(N) = \sum_{e \in N} \mathsf{wt}_M(e) = \phi(N,M) - \phi(M,N)$. Thus M is popular if and only if $\mathsf{wt}_M(N) \leq 0$ for every perfect matching N in G.

Consider the max-weight perfect matching LP in G with the edge weight function wt_M. This linear program is (LP1) given below and (LP2) is the dual of (LP1). The variables x_e for $e \in E'$ are primal variables and the variables y_u for $u \in A \cup B$ are dual variables. Here $\delta'(u) = \delta(u) \cup \{(u,u)\}$.

$$\begin{aligned} \max \sum_{e \in E'} \mathsf{wt}_M(e) \cdot x_e & & \text{(LP1)} \\ \text{s.t.} \quad \sum_{e \in \delta'(u)} x_e = 1 & \quad \forall\, u \in A \cup B \\ x_e \geq 0 & \quad \forall\, e \in E'. \end{aligned}$$

$$\begin{aligned} \min \sum_{u \in A \cup B} y_u & & \text{(LP2)} \\ \text{s.t.} \quad y_a + y_b \geq \mathsf{wt}_M(a,b) & \quad \forall\, (a,b) \in E \\ y_u \geq \mathsf{wt}_M(u,u) & \quad \forall\, u \in A \cup B. \end{aligned}$$

M is popular if and only if the optimal value of (LP1) is at most 0. In fact, the optimal value is exactly 0 since M is a perfect matching in G and $\mathsf{wt}_M(M) = 0$. Thus M is popular if and only if the optimal value of (LP2) is 0 (by LP-duality).

▶ **Theorem 6** **([17, 18]).** *A matching M in $G = (A \cup B, E)$ is popular if and only if there exists $\vec{\alpha} \in \{0, \pm 1\}^n$ (where $|A \cup B| = n$) such that $\sum_{u \in A \cup B} \alpha_u = 0$ along with*

$$\alpha_a + \alpha_b \geq \mathsf{wt}_M(a,b) \quad \forall (a,b) \in E \qquad \text{and} \qquad \alpha_u \geq \mathsf{wt}_M(u,u) \quad \forall u \in A \cup B.$$

Proof. Since E is the edge set of a bipartite graph, the constraint matrix of (LP2) is totally unimodular. So (LP2) admits an optimal solution that is integral. The vector $\vec{\alpha}$ is an integral optimal solution of (LP2). We have $\alpha_u \geq \mathsf{wt}_M(u,u) \geq -1$ for all u.

We now claim $\alpha_u \in \{0, \pm 1\}$ for all vertices u. Since M is an optimal solution to (LP1), complementary slackness implies $\alpha_u + \alpha_v = \mathsf{wt}_M(u,v) = 0$ for each edge $(u,v) \in M$. Thus $\alpha_u = -\alpha_v \leq 1$ for every vertex u matched to a non-trivial neighbor v in M. Regarding any vertex u such that $(u,u) \in M$, we have $\alpha_u = \mathsf{wt}_M(u,u) = 0$ (by complementary slackness). Hence $\vec{\alpha} \in \{0, \pm 1\}^n$. ◀

For any popular matching M, a vector $\vec{\alpha}$ as given in Theorem 6 will be called a *witness* of M's popularity. Note that a popular matching may have several witnesses. A stable matching S has 0^n as a witness since $\mathsf{wt}_S(e) \leq 0$ for all $e \in E'$.

Recall that our problem is to compute a *fully popular* matching, i.e., a popular matching that is also A-popular. It is easy to construct instances that admit A-popular matchings but admit no fully popular matching. It could also be the case that no min-size or max-size popular matching in $G = (A \cup B, E)$ is A-popular, however G has a fully popular matching. Consider the instance G given in Fig. 1. Vertex preferences are indicated on edges: 1 denotes top choice, and so on.

We list the vertices $f(u)$ and $s(u)$ for each $u \in A$ in this instance. The vertex b' is *not* $s(a)$ since a prefers q' to b' and $q' \neq f(u)$ for any $u \in A$.

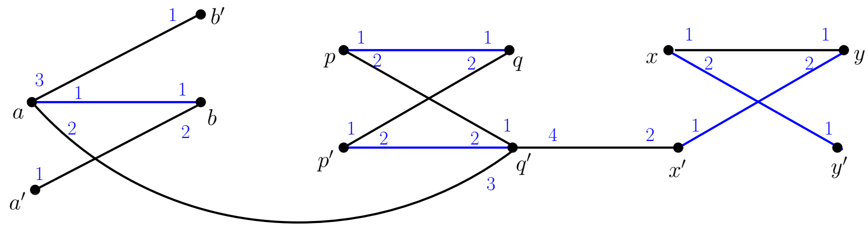

Figure 1 An instance on $A = \{a, a', p, p', x, x'\}$ and $B = \{b, b', q, q', y, y'\}$ where no min-size/max-size popular matching is A-popular. There is a fully popular matching (on blue edges) here.

- We have $f(a) = f(a') = b$, $f(p) = f(p') = q$, and $f(x) = f(x') = y$.
- We have $s(a) = s(p) = s(p') = s(x') = q'$, $s(x) = y'$, and $s(a') = a'$.

Since $s(u) \neq u$ for $u \in \{a, p, p', x, x'\}$, any A-popular matching M has to match these 5 vertices to neighbors in B (by Theorem 5). So $M(a) = f(a) = b$ which implies $M(a') = s(a') = a'$, i.e., after pruning self-loops from M, the vertex a' has to be left unmatched in M. So M has size 5. It is easy to check that a stable matching (thus any min-size popular matching) in G has size 4; also any max-size popular matching has size 6. Thus no min-size or max-size popular matching in G can be fully popular. Interestingly, this instance admits a fully popular matching: $M = \{(a, b), (p, q), (p', q'), (x, y'), (x', y)\}$ is fully popular.

3 Fully Popular Matchings

The input is a marriage instance $G = (A \cup B, E)$. Our algorithm will work in a bipartite graph H which is essentially 2 copies of the graph G as shown in Fig. 2. The vertex set of H is $A_L \cup B_L$ on the left and $B_R \cup A_R$ on the right. Here $A_L = \{a_\ell : a \in A\}$ and $A_R = \{a_r : a \in A\}$. Similarly, $B_L = \{b_\ell : b \in B\}$ and $B_R = \{b_r : b \in B\}$.

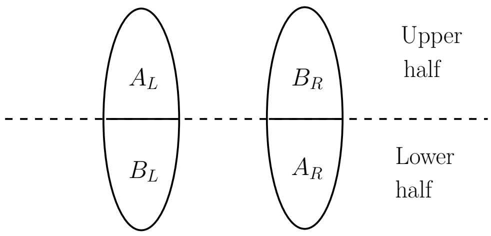

Figure 2 The bipartite graph H consists of 2 copies of the graph $G = (A \cup B, E)$.

The *upper half* of H consists of the set A_L of agents on the left and the set B_R of jobs on the right while the *lower half* of H consists of the set B_L of jobs on the left and the set A_R of agents on the right. Thus every vertex $u \in A \cup B$ has two copies in H: one as u_ℓ on the left of H and another as u_r on the right of H.

For every edge in E, there will be four edges in H: a pair of parallel edges in the upper half and a pair of parallel edges in the lower half. In order to distinguish two parallel edges with the same endpoints, we use superscripts $+$ and $-$ on the endpoints. For $(a, b) \in E$:

- in the upper half, we have two parallel edges (a_ℓ^+, b_r^-) and (a_ℓ^-, b_r^+) between a_ℓ and b_r;
- in the lower half, we have two parallel edges (b_ℓ^+, a_r^-) and (b_ℓ^-, a_r^+) between b_ℓ and a_r.

Corresponding to every vertex $u \in A \cup B$, there will be a *single* edge (u_ℓ^-, u_r^+) in H: this edge corresponds to the self-loop (u, u) and it will be convenient to use $+/-$ superscripts on the endpoints of this edge also. These edges (u_ℓ^-, u_r^+) for all u are the only edges in H that go across the two halves of H.

Vertices in H have preferences on their incident *edges* rather than their neighbors. However it would be more convenient to say u prefers v^- to w^+ rather than say u prefers (u^+, v^-) to (u^-, w^+). In fact, H is equivalent to a conventional graph H^* (with preferences on neighbors) that was used to study popular *half-integral* matchings in [17]: there were 4 vertices in H^* for each $u \in A \cup B$. The graph H is a sparser version of H^* with only 2 vertices u_ℓ and u_r for each $u \in A \cup B$ and a pair of parallel edges between every pair of adjacent vertices. We now describe the preferences of vertices in H.

Every vertex prefers superscript $-$ neighbors to superscript $+$ neighbors: among superscript $-$ neighbors (similarly, superscript $+$ neighbors), it will be its original preference order. Consider any vertex $u \in A \cup B$. Suppose u's preference list in G is $v \succ v' \succ \cdots \succ v''$, i.e., v is u's top choice, next comes v', and so on. In H, the preference list of u_ℓ is as follows:

$$\underbrace{v_r^- \succ v_r'^- \succ \cdots \succ v_r''^-}_{\text{superscript } - \text{ neighbors}} \succ \underbrace{v_r^+ \succ v_r'^+ \succ \cdots \succ v_r''^+ \succ u_r^+}_{\text{superscript } + \text{ neighbors}},$$

where $v_r, v'_r, \ldots$ correspond to the copies of $v, v', \ldots$ on the right side of H.

The vertex u_r^+ is the last choice of u_ℓ. In H, the preference list of u_r is as follows:

$$\underbrace{v_\ell^- \succ v_\ell'^- \succ \cdots \succ v_\ell''^- \succ u_\ell^-}_{\text{superscript } - \text{ neighbors}} \succ \underbrace{v_\ell^+ \succ v_\ell'^+ \succ \cdots \succ v_\ell''^+}_{\text{superscript } + \text{ neighbors}},$$

where $v_\ell, v'_\ell, \ldots$ correspond to the copies of $v, v', \ldots$ on the left side of H. This is analogous to u_ℓ's preference list: the main difference is in the position of its "twin" - the vertex u_r prefers u_ℓ^- to all its superscript $+$ neighbors.

Blocking edges. Let $u_\ell \in A_L \cup B_L$ and $v_r \in A_R \cup B_R$. For any matching M in H, we say an edge (u_ℓ^+, v_r^-) *blocks* M if (i) u_ℓ prefers v_r^- to its assignment in M and (ii) v_r prefers u_ℓ^+ to its assignment in M. Similarly, we say edge (u_ℓ^-, v_r^+) *blocks* M if (i) u_ℓ prefers v_r^+ to its assignment in M and (ii) v_r prefers u_ℓ^- to its assignment in M.

▶ **Definition 7.** *A matching M in H is* stable *if no edge in H blocks M.*

▷ Claim 8. Let S be a stable matching in $G = (A \cup B, E)$. The matching $S' = \{(a_\ell^-, b_r^+), (b_\ell^-, a_r^+) : (a,b) \in S\} \cup \{(u_\ell^-, u_r^+) : u \text{ is unmatched in } S\}$ is stable in H.

Proof. We need to show that no edge in H blocks S'. Note that S' is a perfect matching in H. Consider any edge (c_ℓ^+, d_r^-) in H where $c \in A$ and $d \in B$. Since every vertex prefers superscript $-$ neighbors to superscript $+$ neighbors, the vertex d_r prefers its partner in S' to c_ℓ^+. So consider any edge (c_ℓ^-, d_r^+) in H. If $(c,d) \in S$ then $(c_\ell^-, d_r^+) \in S'$ and so it does not block S'. If $(c,d) \notin S$ then it follows from the stability of S in G that either (1) c is matched to a neighbor b preferred to d or (2) d is matched to a neighbor a preferred to c. In case (1), c_ℓ is matched in S' to a neighbor b_r^+ preferred to d_r^+; in case (2), d_r is matched in S' to a neighbor a_ℓ^- preferred to c_ℓ^-. Thus (c_ℓ^-, d_r^+) does not block S'.

It can analogously be shown that neither (d_ℓ^-, c_r^+) nor (d_ℓ^+, c_r^-) blocks S'. Also (u_ℓ^-, u_r^+) for any $u \in A \cup B$ does not block S' since u_r^+ is u_ℓ's least preferred neighbor in H. Hence S' is a stable matching in H. ◁

Thus the graph H admits a perfect stable matching. Since all stable matchings in H have the same size [11], every stable matching in H has to be perfect. We seek to compute a "special" stable matching in H: one that has no edge that is *forbidden*. The edges marked forbidden are those that no fully popular matching uses. The definition of *valid* edges below is as given in Theorem 5: any A-popular matching in G can contain only these edges/self-loops.

▶ **Definition 9.** *Edges/self-loops in* $\{(a, f(a)), (a, s(a)) : a \in A\}$ *are* valid*. So are self-loops in* $\{(b, b) : b \neq f(a)$ for any $a \in A\}$.

All other edges and self-loops are *invalid*. Thus every $a \in A$ has exactly 2 valid edges incident to it: one of these may be the self-loop (a, a).

An edge e in G is *popular* if there exists a popular matching M in G such that $e \in M$. Call a vertex *stable* if it is matched in some (equivalently, every [11]) stable matching. It is known that every popular matching in G has to match all stable vertices to non-trivial neighbors [14]. So the self-loop (u, u) is popular if and only if u is unstable.

▶ **Definition 10.** *Call an edge e in $E \cup \{(u, u) : u \in A \cup B\}$* legal *if e is valid and popular.*

Forbidden edges. A fully popular matching, by definition, has to contain only legal edges. So if (a, b) is not legal then (a^+_ℓ, b^-_r), (a^-_ℓ, b^+_r), (b^+_ℓ, a^-_r), and (b^-_ℓ, a^+_r) are *forbidden* edges in the stable matching that we seek to compute in H. Similarly, for any $u \in A \cup B$, if (u, u) is not legal then (u^-_ℓ, u^+_r) is a forbidden edge in our algorithm.

▶ **Definition 11.** *A matching M in H is* legal *if M has no forbidden edge.*

Call a matching M in H *symmetric* if for each edge (a, b) in E, we have both (a_ℓ, b_r) and (b_ℓ, a_r) in M or neither (for convenience, we are ignoring $+/-$ superscripts on a_ℓ, a_r, b_ℓ, b_r), i.e., loosely speaking, M has the same edges in the upper and lower halves of H. A symmetric matching M in H will be called a *realization* of $\tilde{M} = \{(a, b) : (a_\ell, b_r)$ and (b_ℓ, a_r) are in $M\}$. Note that $\tilde{M}$ is a matching in G.

▶ **Lemma 12.** *Every fully popular matching in G has a realization as a legal stable matching in the instance H.*

Proof. Let N be a fully popular matching in G and let $\vec{\alpha} \in \{0, \pm 1\}^n$ be a witness of N's popularity (see Theorem 6). For $i \in \{0, \pm 1\}$, let A_i be the set of vertices $a \in A$ with $\alpha_a = i$ and let B_i be the set of vertices $b \in B$ with $\alpha_b = i$. Thus we have $A = A_0 \cup A_1 \cup A_{-1}$ and $B = B_0 \cup B_1 \cup B_{-1}$. Note that $\alpha_a + \alpha_b = \mathsf{wt}_N(a, b) = 0$ for each edge $(a, b) \in N$: this is by complementary slackness on (LP2) corresponding to N. So the matching $N \subseteq (A_0 \times B_0) \cup (A_{-1} \times B_1) \cup (A_1 \times B_{-1})$ (see Fig. 3).

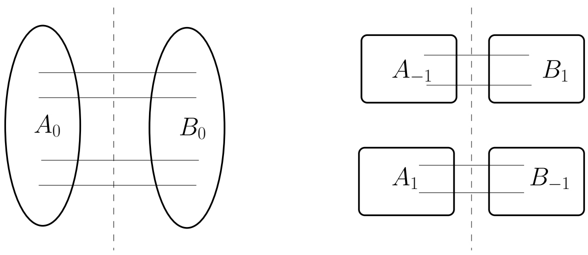

Figure 3 The partition $A_0 \cup A_1 \cup A_{-1}$ of A and $B_0 \cup B_{-1} \cup B_1$ of B.

We need to show a realization of N in H that is stable. We will use N's witness $\vec{\alpha}$ in G to define the following matching N^*_α in H: this is similar to how popular half-integral matchings were realized as stable matchings in a larger graph H^* in [17].

- For all $(a, b) \in N \cap (A_{-1} \times B_1)$ do: add edges (a^-_ℓ, b^+_r) and (b^+_ℓ, a^-_r) to N^*_α.
- For all $(a, b) \in N \cap (A_1 \times B_{-1})$ do: add edges (a^+_ℓ, b^-_r) and (b^-_ℓ, a^+_r) to N^*_α.
- For all $(a, b) \in N \cap (A_0 \times B_0)$ do: add edges (a^-_ℓ, b^+_r) and (b^-_ℓ, a^+_r) to N^*_α.

For each u such that $(u,u) \in N$, add (u_ℓ^-, u_r^+) to N_α^*. Using the constraints that $\vec{\alpha}$ has to satisfy, it is easy to argue that N_α^* is a stable matching in H. Moreover, the fact that N is a fully popular matching in G implies that N_α^* is a *legal* matching in H: since N is A-popular (resp., popular) in G, every edge used in N is valid (resp., popular). So N_α^* has no forbidden edge. Thus N_α^* is a legal stable matching in H. ◀

A variant of Gale-Shapley algorithm. A stable matching that avoids all forbidden edges (if such a matching exists) can be computed in linear time by running a variant of Gale-Shapley algorithm in H where proposals made along forbidden edges are rejected. Once a proposal made along a forbidden edge is rejected by a vertex, all further proposals made on *worse* edges also have to rejected by this vertex. If some vertex is left unmatched at the end of this algorithm, then there is no stable matching in H that avoids all forbidden edges; else we have a desired stable matching in H. We refer to [13] for details on this variant of Gale-Shapley algorithm.

Thus it can be efficiently checked if H admits a legal stable matching or not. If such a matching does not exist in H then there is no fully popular matching in G (by Lemma 12). So we will assume henceforth that there exists a legal stable matching in H. However the fact that such a stable matching exists in H does not imply that G admits a fully popular matching. This is because any matching M^* in H can only be mapped to a *half-integral* matching in G.

In order to claim the resulting matching in G is integral, we need M^* to be symmetric, i.e., have the same edges in both halves of H. We will not construct such a symmetric stable matching in H. The matching we compute will have a certain amount of symmetry and this will be enough to obtain a fully popular matching in G. If H does not admit such a "partially symmetric" stable matching, then we show that G has *no* fully popular matching.

3.1 Two partitions of the vertex set

We run Gale-Shapley algorithm that avoids all forbidden edges [13] in H. In this algorithm, vertices on the left of H propose and vertices on the right of H dispose. When $u_\ell \in A_L \cup B_L$ proposes to v_r^-, this proposal is made along (u_ℓ^+, v_r^-): so v_r sees this as u_ℓ^+'s proposal; when u_ℓ proposes to v_r^+, this proposal is made along (u_ℓ^-, v_r^+): so v_r sees this as u_ℓ^-'s proposal. If u_ℓ proposes to a neighbor v_r along (u_ℓ^+, v_r^-) or (u_ℓ^-, v_r^+), then v_r accepts u_ℓ's proposal only if the edge (u,v) is legal; otherwise v_r rejects u_ℓ's proposal since this is a forbidden edge. Edges ranked worse than $(u_\ell^+, v_r^-)/(u_\ell^-, v_r^+)$ (as the case may be) will be deleted from the current instance on whose edges proposals are made – this ensures that once v_r receives a proposal along a certain edge, whether this proposal is accepted or not, v_r cannot accept proposals made along *worse* edges. Let S_0 be the legal stable matching in H that is obtained.

- Let $U_A \subseteq A$ be the set of agents a such that $(a_\ell^-, a_r^+) \in S_0$ and let $U_B \subseteq B$ be the set of jobs b such that $(b_\ell^-, b_r^+) \in S_0$.

We know that H is made up of two halves: the upper half and the lower half. Since S_0 is stable and thus perfect (recall that any stable matching in H is perfect), the set of agents matched to *genuine* neighbors – not to their twins – in each half of H is $A \setminus U_A$ and similarly, the set of jobs matched to genuine neighbors in each half of H is $B \setminus U_B$. We now form sets $A_+, A_-, B_+, B_-, A'_+, A'_-, B'_+, B'_-$ corresponding to S_0: initially they are empty.

- For every $(a_\ell^+, b_r^-) \in S_0$ where $a \in A$ and $b \in B$: add a to A_+ and b to B_-.
- For every $(a_\ell^-, b_r^+) \in S_0$ where $a \in A$ and $b \in B$: add a to A_- and b to B_+.
- For every $(b_\ell^+, a_r^-) \in S_0$ where $a \in A$ and $b \in B$: add b to B'_+ and a to A'_-.
- For every $(b_\ell^-, a_r^+) \in S_0$ where $a \in A$ and $b \in B$: add b to B'_- and a to A'_+.

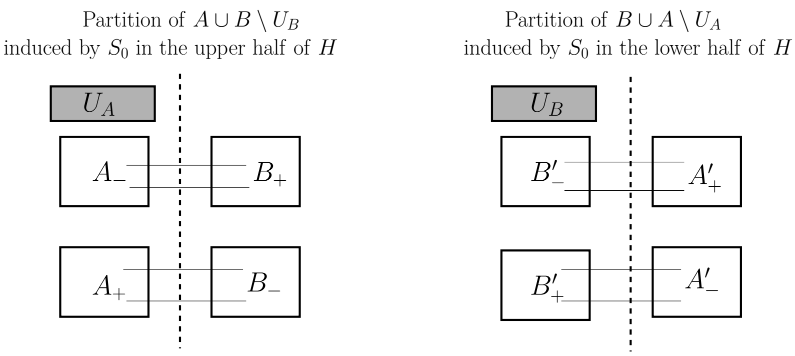

Figure 4 Two partitions of the set $(A \cup B) \setminus (U_A \cup U_B)$ induced by the matching S_0.

We have $A \setminus U_A = A_+ \cup A_- = A'_+ \cup A'_-$ and $B \setminus U_B = B_+ \cup B_- = B'_+ \cup B'_-$. Fig. 4 denotes these partitions of $A \setminus U_A$ and $B \setminus U_B$ induced by the matching S_0 in the upper and lower halves of H. In this figure, the set U_A has been included in the upper half and the set U_B in the lower half.

We will use $(a_\ell^-, *)$ to denote any edge in the set $\{(a_\ell^-, b_r^+) : b \in B\} \cup \{(a_\ell^-, a_r^+)\}$. Similarly $(*, a_r^+)$ denotes any edge in the set $\{(b_\ell^-, a_r^+) : b \in B\} \cup \{(a_\ell^-, a_r^+)\}$. Similarly for $(b_\ell^-, *)$ and $(*, b_r^+)$. Recall that every popular matching has a *witness* $\vec{\alpha} \in \{0, \pm 1\}^n$ (see Theorem 6).

▶ **Lemma 13.** *Let N be a fully popular matching in G and let $\vec{\alpha}$ be any witness of N. If $a \in A_- \cap A'_+$ then $\alpha_a = 0$.*

Proof. We have vertex $a \in A_- \cap A'_+$, where the sets A_- and A'_+ are defined above. Let $\mathcal{D}_0$ be the set of legal stable matchings in H. The set $\mathcal{D}_0$ forms a sublattice of the lattice[1] of stable matchings in H and the matching S_0 is the $(A_L \cup B_L)$-*optimal* matching in $\mathcal{D}_0$ [13]. Since $a \in A_-$, we have $(a_\ell^-, c_r^+) \in S_0$ for some neighbor c_r^+ of a_ℓ. Thus c_r^+ is the most preferred partner for $a_\ell \in A_L$ in all matchings in $\mathcal{D}_0$. Recall that every vertex prefers superscript $-$ neighbors to superscript $+$ neighbors. Hence no matching in $\mathcal{D}_0$ matches a_ℓ to a superscript $-$ neighbor, i.e., every legal stable matching in H has to contain $(a_\ell^-, *)$.

S_0 is also the $(A_R \cup B_R)$-*pessimal* matching in $\mathcal{D}_0$ [13]. Since $a \in A'_+$, we have $(d_\ell^-, a_r^+) \in S_0$ for some neighbor d_ℓ^- of a_r. So every matching in $\mathcal{D}_0$ has to match $a_r \in A_R$ to a neighbor at least as good as d_ℓ^-, i.e., every legal stable matching in H has to contain $(*, a_r^+)$.

Suppose N is a fully popular matching with a witness $\vec{\alpha}$ such that $\alpha_a \in \{\pm 1\}$. If $\alpha_a = 1$, i.e., if $a \in A_1$ (see Fig. 3), then there is a legal stable matching N^*_α in H such that $(a_\ell^+, *) \in N^*_\alpha$ (see the proof of Lemma 12). This contradicts our claim above that every legal stable matching in H has to contain $(a_\ell^-, *)$. So $\alpha_a = -1$, i.e., $a \in A_{-1}$. Then there is a legal stable matching N^*_α in H such that $(*, a_r^-) \in N^*_\alpha$ (by the proof of Lemma 12). This again contradicts our claim above that every legal stable matching in H has to contain $(*, a_r^+)$. Thus $\alpha_a \notin \{\pm 1\}$, hence $\alpha_a = 0$. ◀

[1] The meet of 2 stable matchings M and M' is the stable matching where every u in $A_L \cup B_L$ (resp., $A_R \cup B_R$) is matched to its *more* (resp., *less*) preferred partner in $\{M(u), M'(u)\}$. The join of M and M' is the stable matching where every u in $A_L \cup B_L$ (resp., $A_R \cup B_R$) is matched to its *less* (resp., *more*) preferred partner in $\{M(u), M'(u)\}$.

The proof of Lemma 14 is analogous to the proof of Lemma 13.

▶ **Lemma 14.** *Let N be a fully popular matching in G and let $\vec{\alpha}$ be any witness of N. If $b \in B_+ \cap B'_-$ then $\alpha_b = 0$.*

We will use $G_0 = (A \cup B, E_0)$ to denote the *popular subgraph* of G. The edge set E_0 of G_0 is the set of popular edges, i.e., those present in some popular matching in G. The subgraph G_0 need not be connected and Lemma 15 will be useful to us.

▶ **Lemma 15** ([8]). *Let C be any connected component in the popular subgraph G_0. For any popular matching N in G and any witness $\vec{\alpha}$ of N: if $\alpha_v = 0$ for some $v \in C$ then $\alpha_u = 0$ for all $u \in C$.*

Proof. Consider any popular edge (a, b). So there is some popular matching M with the edge (a, b). The matching M is an optimal solution to the max-weight perfect matching LP with edge weight function wt_N since $\mathsf{wt}_N(M) = \phi(M, N) - \phi(N, M) = 0$: recall that M and N are popular matchings in G. We know that $\vec{\alpha}$ is an optimal solution to the dual LP. So it follows from complementary slackness conditions that $\alpha_a + \alpha_b = \mathsf{wt}_N(a, b)$. Since $\mathsf{wt}_N(a, b) \in \{\pm 2, 0\}$ (an even number), the integers α_a and α_b have the same parity.

Let u and v be any 2 vertices in the same connected component in the popular subgraph G_0. So there is a u-v path ρ in G such that every edge in ρ is a popular edge. We have just seen that the endpoints of each popular edge have the same parity in $\vec{\alpha}$. Hence α_u and α_v have the same parity. Thus $\alpha_v = 0$ implies $\alpha_u = 0$. ◀

3.2 Our algorithm

Lemmas 13-15 motivate our algorithm which is described as Algorithm 1. The main step of the algorithm is the *while* loop that takes any unmarked vertex v in $(A_- \cap A'_+) \cup (B_+ \cap B'_-)$. Initially all vertices are unmarked. Consider the first iteration of the algorithm: let $v \in A$.

Lemma 13 tells us that for any fully popular matching N and any witness $\vec{\alpha}$ of N, we have $\alpha_v = 0$. Lemma 15 tells us that $\alpha_u = 0$ for every vertex u in the component C, where C is v's connected component in G_0. The proof of Lemma 12 shows N has a realization N^*_α in H such that N^*_α contains $(a^-_\ell, *)$ and $(*, a^+_r)$ for every agent $a \in C$.

Thus we are interested in those legal stable matchings in H that contain $(a^-_\ell, *)$ and $(*, a^+_r)$ for every agent $a \in C$. Hence our algorithm *forbids* all edges $(a^+_\ell, *)$ and $(*, a^-_r)$ for every agent $a \in C$ in the stable matching that we compute here. This step is implemented by making every neighbor reject offers from a^+_ℓ (this may induce other rejections) and symmetrically, a_r rejects all offers from superscript $+$ neighbors. Note that the resulting matching may contain (a^-_ℓ, a^+_r) for some of the agents a in C. All vertices in C get marked in this iteration.

Let $\mathcal{D}_1 \subseteq \mathcal{D}_0$ be the set of all legal stable matchings in H that contain $(a^-_\ell, *)$ and $(*, a^+_r)$ for every agent $a \in C$. Thus $\mathcal{D}_1$ is a sublattice of $\mathcal{D}_0$. We know from the proof of Lemma 12 that $N^*_\alpha \in \mathcal{D}_1$ where N is a fully popular matching in G and $\vec{\alpha}$ any witness of N. So if $\mathcal{D}_1$ is empty then we can conclude that G has *no* fully popular matching.

Algorithm 1 *Input:* $G = (A \cup B, E)$ *with strict preferences.*

1: Compute a legal stable matching S_0 in H by running Gale-Shapley algorithm with forbidden edges.
   {*Vertices in* $A_L \cup B_L$ *propose and those in* $B_R \cup A_R$ *dispose.*}
2: Let A_-, A'_+ and B_+, B'_- be as defined earlier (see the start of Section 3.1).
3: Initially all vertices are unmarked and $i = 0$.
4: **while** there exists an unmarked vertex $v \in (A_- \cap A'_+) \cup (B_+ \cap B'_-)$ **do**
5:     $i = i + 1$.
6:     Modify S_{i-1} to S_i so as to forbid all edges $(a^+_\ell, *)$ and $(*, a^-_r)$ for every agent a in v's component in the popular subgraph G_0.
       (S_i *is the* $(A_L \cup B_L)$*-optimal legal stable matching in* H *that avoids all forbidden edges identified in the first* i *iterations of the while-loop.*)
7:     **if** there is no such legal stable matching S_i in H **then**
8:         Return "*No fully popular matching in* G".
9:     **end if**
10:    Update the sets A_-, A'_+ and B_+, B'_-: these correspond to S_i now.
11:    Mark all vertices in v's component in the popular subgraph G_0.
12: **end while**
13: Return $M = \{(a, b) \in E : (a^+_\ell, b^-_r) \text{ or } (a^-_\ell, b^+_r) \text{ is in } S_i\}$.

Let us assume we are now in the i-th iteration and let $\mathcal{D}_i$ be the set of legal stable matchings in H that avoid all edges forbidden by our algorithm in the first i iterations. In other words, $\mathcal{D}_i$ is the set of those matchings in $\mathcal{D}_{i-1}$ where no edge forbidden in the i-th iteration is present. We have $\mathcal{D}_0 \supseteq \mathcal{D}_1 \supseteq \cdots \supseteq \mathcal{D}_{i-1} \supseteq \mathcal{D}_i$. For all $0 \le j \le i$, the set $\mathcal{D}_j$ forms a sublattice of the lattice of all stable matchings in H [13].

▶ **Lemma 16.** *For every fully popular matching* N *in* G *and every witness* $\vec{\alpha}$ *of* N*, the realization* N^*_α *is an element of* $\mathcal{D}_i$*.*

Proof. We will prove the lemma by induction. We know from Lemma 12 that the base case is true, i.e., $N^*_\alpha \in \mathcal{D}_0$. By induction hypothesis, let us assume that for every fully popular matching N and every witness $\vec{\alpha}$, the realization N^*_α is an element of $\mathcal{D}_{i-1}$. Since the algorithm entered the i-th iteration of the while loop, there was an unmarked vertex x in $(A_- \cap A'_+) \cup (B_+ \cap B'_-)$ at the start of this iteration.

▷ Claim 17. For any fully popular matching N and any witness $\vec{\alpha}$ of N, we have $\alpha_x = 0$.

Proof. The matching S_{i-1} computed in Step 6 of the $(i-1)$-th iteration is the $(A_L \cup B_L)$-optimal matching in the lattice $\mathcal{D}_{i-1}$. Hence if $(x^-_\ell, *) \in S_{i-1}$ for some $x_\ell \in A_L \cup B_L$ then $(x^-_\ell, *)$ belongs to every matching in $\mathcal{D}_{i-1}$. The matching S_{i-1} is also the $(A_R \cup B_R)$-pessimal matching in the set $\mathcal{D}_{i-1}$. Hence if $(*, x^+_r) \in S_{i-1}$ for some $x_r \in A_R \cup B_R$ then $(*, x^+_r)$ belongs to every matching in $\mathcal{D}_{i-1}$.

If the above claim is false then there is a fully popular matching N and a witness $\vec{\alpha}$ of N with $\alpha_x \in \{\pm 1\}$. If $\alpha_x = 1$ then there is a legal stable matching N^*_α in H such that $(x^+_\ell, *) \in N^*_\alpha$. If $\alpha_x = -1$ then there is a legal stable matching N^*_α in H such that $(*, x^-_r) \in N^*_\alpha$. Since $N^*_\alpha \in \mathcal{D}_{i-1}$, both cases contradict our earlier observation that every matching in $\mathcal{D}_{i-1}$ has to contain $(x^-_\ell, *)$ and $(*, x^+_r)$. Thus for any fully popular matching N and any witness $\vec{\alpha}$ of N, we have $\alpha_x = 0$. ◁

Claim 17 along with Lemma 15 tells us that for all vertices u in x's component C' in G_0, we have $\alpha_u = 0$. The proof of Lemma 12 shows us that N^*_α contains $(a^-_\ell, *)$ and $(*, a^+_r)$ for every agent $a \in C'$. Since $N^*_\alpha \in \mathcal{D}_{i-1}$, it follows that N^*_α is an element in $\mathcal{D}_i$. This finishes the proof of this lemma. ◀

Hence if $\mathcal{D}_i = \emptyset$, i.e., if the algorithm says "no" in Step 8, then there is indeed no fully popular matching in G. This finishes one part of our proof of correctness. We now need to show that if our algorithm returns a matching M, then M is a fully popular matching in G.

4 Correctness of our algorithm

In this section we show that the matching returned by Algorithm 1 is a fully popular matching in G. Let S_i be the matching in H computed in the final iteration of Algorithm 1. Let M be the matching (in G) induced by S_i in the upper half of H: this is as defined in Step 13 of Algorithm 1.

Note that $M \subseteq (A_+ \times B_-) \cup (A_- \times B_+)$, where the sets A_+, B_-, A_-, B_+ are defined at the beginning of Section 3.1: the matching S_i replaces S_0 in these definitions now. Similarly, let L be the matching (in G) induced by S_i in the lower half of H. So

$$L = \{(a,b) \in E : (b^+_\ell, a^-_r) \text{ or } (b^-_\ell, a^+_r) \text{ is in } S_i\}.$$

Thus $L \subseteq (A'_+ \times B'_-) \cup (A'_- \times B'_+)$. Let U_A (resp., U_B) be the set of vertices u in A (resp., B) such that $(u^-_\ell, u^+_r) \in S_i$. The vertices in $U_A \cup U_B$ are unmatched in both M and L. Since S_i is a legal stable matching in H, it matches all vertices in H using valid edges. Thus by Theorem 5, M is A-popular.[2] We need to show that M is popular in G. Theorem 18 will be our starting point.

▶ **Theorem 18.** *The matching M is popular in the subgraph $G \setminus U_B$. Also, the matching L is popular in the subgraph $G \setminus U_A$.*

Proof. The popularity of L in $G \setminus U_A$ will be shown using the witness $\vec{\beta}$ defined below and the popularity of M in $G \setminus U_B$ will be shown using the witness $\vec{\gamma}$ defined below.

1. $\beta_u = 1$ for $u \in A'_+ \cup B'_+$, $\beta_u = -1$ for $u \in A'_- \cup B'_-$, and $\beta_u = 0$ for $u \in U_B$.
2. $\gamma_u = 1$ for $u \in A_+ \cup B_+$, $\gamma_u = -1$ for $u \in A_- \cup B_-$, and $\gamma_u = 0$ for $u \in U_A$.

Since $L \subseteq (A'_+ \times B'_-) \cup (A'_- \times B'_+)$, we have $\sum_{u \in (A \cup B) \setminus U_A} \beta_u = 0$. Note that $\mathsf{wt}_L(u,u) = 0$ for $u \in U_B$ and $\mathsf{wt}_L(u,u) = -1$ for all other u. Thus $\beta_u \geq \mathsf{wt}_L(u,u)$ for all $u \in (A \cup B) \setminus U_A$.

Similarly, $\sum_{u \in (A \cup B) \setminus U_B} \gamma_u = 0$. Also, $\gamma_u \geq \mathsf{wt}_M(u,u)$ for all $u \in (A \cup B) \setminus U_B$.

▷ Claim 19. $\beta_a + \beta_b \geq \mathsf{wt}_L(a,b)$ for all edges (a,b) where $a \in A \setminus U_A$ and $b \in B$.

▷ Claim 20. $\gamma_a + \gamma_b \geq \mathsf{wt}_M(a,b)$ for all edges (a,b) where $a \in A$ and $b \in B \setminus U_B$.

We will prove Claim 20 below. The proof of Claim 19 is analogous.

- Consider $a \in U_A$. We set $\gamma_a = 0$ and we know that $(a^-_\ell, a^+_r) \in S_i$. Recall that a^+_r is a_ℓ's least preferred neighbor, thus a_ℓ must have been rejected by all its more preferred neighbors in our variant of the Gale-Shapley algorithm in H. That is, every neighbor b^+_r of a_ℓ received a proposal from a^-_ℓ. Since b_r prefers superscript $-$ neighbors to superscript $+$

[2] In order to apply Theorem 5, we ought to say $M \cup \{(u,u) : u \in U_A \cup U_B\}$ is A-popular.

neighbors, this means $(d^-_\ell, b^+_r) \in S_i$ for some neighbor d^-_ℓ that b_r prefers to a^-_ℓ, i.e., b prefers d to a. Thus $b \in B_1$ (so $\gamma_b = 1$) and moreover, $\mathsf{wt}_M(a,b) = 0$. Hence we have $\gamma_a + \gamma_b = 1 > \mathsf{wt}_M(a,b)$.

- We will next show this constraint holds for all edges (a,b) incident to $a \in A_-$. There are 2 cases: (1) $b \in B_-$ and (2) $b \in B_+$. In case (1), we have (a^-_ℓ, c^+_r) and (d^+_ℓ, b^-_r) in S_i. Since every vertex prefers superscript $-$ neighbors to superscript $+$ neighbors, it means a_ℓ proposed to b^-_r and got rejected, i.e., b_r prefers its partner d^+_ℓ to a^+_ℓ. We also claim a_ℓ prefers its partner c^+_r to b^+_r. This is because b_r prefers a^-_ℓ to d^+_ℓ (superscript $-$ neighbors over superscript $+$ neighbors): so if a^-_ℓ had proposed to b_r, then b_r would have rejected its partner d^+_ℓ. This means both a and b prefers their partners in M to each other. Thus $\mathsf{wt}_M(a,b) = -2 = \gamma_a + \gamma_b$.
  In case (2) above, either (i) $(a^-_\ell, b^+_r) \in S_i$ or (ii) (a^-_ℓ, c^+_r) and (d^-_ℓ, b^+_r) are in S_i: since S_i is stable, a_ℓ prefers c^+_r to b^+_r or b_r prefers d^-_ℓ to a^-_ℓ. Thus $\mathsf{wt}_M(a,b) \le 0 = \gamma_a + \gamma_b$.
- Finally, we will show this constraint holds for all edges (a,b) incident to $a \in A_+$. There are 2 cases: $b \in B_+$ and $b \in B_-$. In the first case, we have $\gamma_a + \gamma_b = 2$ and since $\mathsf{wt}_M(a,b) \le 2$, the constraint $\mathsf{wt}_M(a,b) \le \gamma_a + \gamma_b$ obviously holds.
  In the second case, either (i) $(a^+_\ell, b^-_r) \in S_i$ or (ii) (a^+_ℓ, c^-_r) and (d^+_ℓ, b^-_r) are in S_i: since S_i is stable, a_ℓ prefers c^-_r to b^-_r or b_r prefers d^+_ℓ to a^+_ℓ. Thus $\mathsf{wt}_M(a,b) \le 0 = \gamma_a + \gamma_b$.

This finishes the proof of M's popularity in $G \setminus U_B$ (by Theorem 6). ◀

Thus the matching M is popular in the subgraph $G \setminus U_B$. However we need to prove the popularity of M in the *entire* graph G, i.e., we need to include vertices in U_B as well. Setting $\gamma_b = 0$ for $b \in U_B$ will not cover edges in $A_- \times U_B$. Now we will use the fact that L is popular in $G \setminus U_A$ and that M and L have several edges in common (as shown in Lemma 22).

Let Z be the set of all vertices outside $U_A \cup U_B$ that got marked in our algorithm. So these are the marked vertices that are matched in S_i to genuine neighbors (not to their twins). Since we marked entire connected components in the popular subgraph G_0 in Algorithm 1, both M and L match vertices in Z to each other.

Lemma 22 shows that the matching S_i has "partial symmetry" across the upper and lower halves of the graph H; more precisely, M and L are identical on the set Z. This will be key to showing M's popularity. The following lemma will be useful in proving Lemma 22.

▶ **Lemma 21.** *M and L are stable matchings when restricted to vertices in $Z \cup U_A \cup U_B$.*

Proof. Let $Z_A = Z \cap A$ and let $Z_B = Z \cap B$. It follows from our algorithm that $Z_A \subseteq A_- \cap A'_+$ and $Z_B \subseteq B_+ \cap B'_-$ (see Fig. 5).

We need to show that M (similarly, L) has no blocking edge in $(Z_A \cup U_A) \times (Z_B \cup U_B)$. Consider any edge $(a,b) \in Z_A \times Z_B$. We know from Claim 20 (in the proof of Theorem 18) that $\mathsf{wt}_M(a,b) \le \gamma_a + \gamma_b = -1 + 1 = 0$. Similarly, $\mathsf{wt}_L(a,b) \le \beta_a + \beta_b = 1 - 1 = 0$. Thus (a,b) is not a blocking edge to either M or L.

Thus neither M nor L has a blocking edge in $Z_A \times Z_B$. Moreover, G has no edge in $U_A \times U_B$. This is because each vertex $u \in U_A \cup U_B$ is unstable – otherwise (u^-_ℓ, u^+_r) is an unpopular edge and thus forbidden. Consider any edge $(a,b) \in U_A \times Z_B$. We have $\mathsf{wt}_M(a,b) \le \gamma_a + \gamma_b = 0 + 1 = 1$. Since $\mathsf{wt}_M(a,b)$ is an even number, this means $\mathsf{wt}_M(a,b) \le 0$. Thus (a,b) is not a blocking edge to M.

We will now show that (a,b) is not a blocking edge to L. Since $a \in U_A$ and $b \in Z_B \subseteq B'_-$, we have (a^-_ℓ, a^+_r) and (b^-_ℓ, c^+_r) in S_i, where c is some neighbor of b. Note that a^-_ℓ is a^+_r's least preferred superscript $-$ neighbor in H. Thus a^+_r did not receive any offer from b^-_ℓ in Algorithm 1. Because S_i is stable, it has to be the case that b_ℓ prefers c^+_r to a^+_r. Since $(c,b) \in L$, we have $\mathsf{wt}_L(a,b) = 0$. Thus (a,b) is not a blocking edge to L.

It can similarly be shown that no edge in $Z_A \times U_B$ blocks either M or L. Thus M and L are stable matchings when restricted to vertices in $Z \cup U_A \cup U_B$. ◀

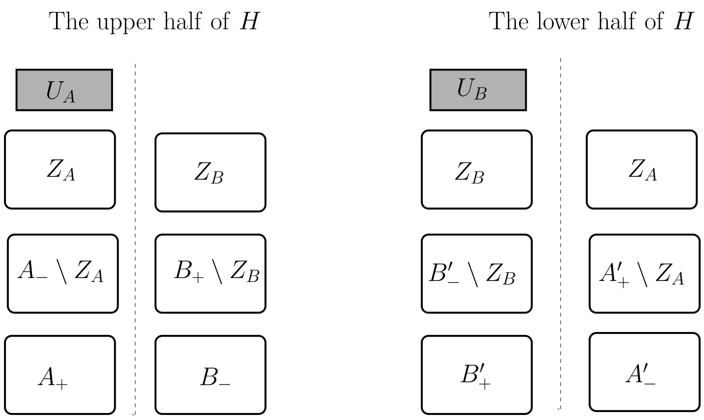

Figure 5 The final picture of the partitions created by M and L in the upper and lower halves of H, resp. The while-loop termination condition implies $(A_- \setminus Z_A) \subseteq A'_-$ and $(A'_+ \setminus Z_A) \subseteq A_+$, so on.

▶ **Lemma 22.** *The matching M restricted to vertices in Z is the same as the matching L restricted to vertices in Z.*

Proof. Consider any connected component C in the popular subgraph G_0. The component C splits into sub-components $C'_1, \ldots, C'_t$ when we restrict edges to only those marked "valid". We claim there is exactly *one* stable matching $T_{C'_j}$ in each such sub-component C'_j. Assume C'_j contains a job b that is a top choice neighbor for some agent.[3] Then b has to be matched in $T_{C'_j}$ to its most preferred neighbor a in C'_j, otherwise (a, b) would be a blocking edge to $T_{C'_j}$. Recall that every agent has exactly 2 valid edges incident to it. So fixing one edge (a, b) in the matching fixes $T_{C'_j}$.

In more detail, every agent $a' \neq a$ in C'_j such that $f(a') = b$ has to be matched in $T_{C'_j}$ to $s(a')$ (call it b'). Given that a' is matched to b', every agent $a'' \neq a'$ in C'_j such that $s(a'') = b'$ has to be matched in $T_{C'_j}$ to $f(a'')$ and so on. Thus the matching $T_{C'_j}$ gets fixed. The same happens with every sub-component in C and so the only stable matching in C is $T_C = \cup_{j=1}^{t} T_{C'_j}$.

Let $C_1, \ldots, C_r$ be the connected components of G_0 that contain vertices in Z. So all vertices in $\cup_{i=1}^{r} C_i$ are marked, thus $\cup_{i=1}^{r} C_i \subseteq Z \cup U_A \cup U_B$. We know from Lemma 21 that both M and L are stable matchings in each C_i, where $1 \le i \le r$. So M (similarly, L) restricted to $\cup_{i=1}^{r} C_i$ is $\cup_{i=1}^{r} T_{C_i}$. Thus M and L have the same edges on Z. ◀

Lemma 22 helps us in defining an appropriate witness $\vec{\alpha}$ to show M's popularity in G. Recall $\vec{\gamma}$ from Theorem 18: we will set $\alpha_u = 0$ for all $u \in Z \cup U_B$ and $\alpha_u = \gamma_u$ otherwise. Before we prove the popularity of M in Theorem 25, we need the following two lemmas.

▶ **Lemma 23.** *For every $a \in A_- \setminus Z_A$, a likes $M(a)$ at least as much as $L(a)$.*

Proof. Suppose not. Then $M(a) = s(a)$ while $L(a) = f(a)$. We claim $f(a) \in B_+$. Otherwise $f(a) \in B_-$, however for every edge $(x, y) \in A_- \times B_-$, we have $\mathsf{wt}_M(x, y) \le \gamma_x + \gamma_y = -2$. But a prefers $f(a)$ to its partner in M, thus $\mathsf{wt}_M(a, f(a)) \ge 0$. Hence $f(a) \in B_+$. Since $\mathsf{wt}_M(x, y) \le 0$ for every edge $(x, y) \in A_- \times B_+$, we can conclude that $\mathsf{wt}_M(a, f(a)) = 0$, i.e., $f(a)$ is matched in M to a neighbor $a' \in A_-$ that it prefers to a. Since S_i uses only valid edges, this means $f(a) = f(a')$, i.e., $f(a)$ is the top choice neighbor of a'.

[3] Otherwise C'_j consists of a single edge $(a, s(a))$ for some $a \in A$; if there was another agent u in C'_j then $s(u) = s(a)$ and so one of a, u would be left unmatched in S_i, a contradiction to S_i's stability in H.

We now move to the lower half of H: observe that both a and a' are in A'_-. This is because there is no unmarked vertex in $A_- \cap A'_+$ by the termination condition of our while-loop. Note that a is unmarked since $a \notin Z_A$. Thus a' is also unmarked since $(a, f(a))$ and $(a', f(a))$ are popular edges – hence a and a' are in the same connected component in G_0. Since $a \in A'_-$, $L(a) = f(a)$ is in B'_+. Consider the edge $(a', f(a)) \in A'_- \times B'_+$: both a' and $f(a)$ prefer each other to their respective partners in L. This means $\mathsf{wt}_L(a', f(a)) = 2$. However for each edge $(x, y) \in A'_- \times B'_+$, we have $\mathsf{wt}_L(x, y) \leq \beta_x + \beta_y = 0$, a contradiction. So any $a \in A_- \setminus Z_A$ likes $M(a)$ at least as much as $L(a)$. ◀

▶ **Lemma 24.** *For every $a \in A_+ \cap A'_+$, a likes $M(a)$ at least as much as $L(a)$.*

Proof. Suppose not. Then $M(a) = s(a)$ while $L(a) = f(a)$. Since $a \in A'_+$, $L(a) = f(a) \in B'_-$. This implies $f(a) \in B_-$ since there is no unmarked vertex in $B_+ \cap B'_-$ by the termination condition of our while-loop. We know $f(a)$ is unmarked since a (its partner in L) is unmarked and this is because $a \in A_+$. Since $a \in A_+$ and $f(a) \in B_-$, we have $\mathsf{wt}_M(a, f(a)) \leq \gamma_a + \gamma_b = 0$ and so $f(a)$ has to be matched in M to a more preferred neighbor $a' \in A_+$. As argued in the proof of Lemma 23, it follows from the legality of S_i that $f(a)$ is the top choice neighbor of a'.

Consider the matching L in the lower half of H. Since $L(a) = f(a)$, $\mathsf{wt}_L(a', f(a)) = 2$. That is, $(a', f(a))$ is a blocking edge to L. We need $\beta_{a'} = \beta_{f(a)} = 1$ to ensure $\beta_{a'} + \beta_{f(a)} \geq \mathsf{wt}_L(a', f(a)) = 2$. However $f(a) \in B'_-$ since $a \in A'_+$. This means $\beta_{f(a)} = -1$, a contradiction. Thus any $a \in A_+ \cap A'_+$ likes $M(a)$ at least as much as $L(a)$. ◀

▶ **Theorem 25.** *The matching M is popular in G.*

Proof. The popularity of M in G will be shown using $\vec{\alpha}$ defined below: (recall that $Z_A = Z \cap A$ and $Z_B = Z \cap B$)

- set $\alpha_u = 0\ \forall u \in Z \cup U_A \cup U_B$.
- set $\alpha_u = 1\ \forall u \in A_+ \cup (B_+ \setminus Z_B)$ and $\alpha_u = -1\ \forall u \in B_- \cup (A_- \setminus Z_A)$.

Since $M \subseteq (A_+ \times B_-) \cup (Z_A \times Z_B) \cup ((A_- \setminus Z_A) \times (B_+ \setminus Z_B))$ (see Fig. 5), we have $\sum_{u \in A \cup B} \alpha_u = 0$. Also $\alpha_u \geq \mathsf{wt}_M(u, u)$ for all vertices $u \in A \cup B$ since $\alpha_u = 0 = \mathsf{wt}_M(u, u)$ for $u \in U_A \cup U_B$ and $\alpha_u \geq -1 = \mathsf{wt}_M(u, u)$ for all other u. To use Theorem 6, we need to show $\alpha_a + \alpha_b \geq \mathsf{wt}_M(a, b)$ for all edges (a, b).

We will first show this constraint holds for edges incident to vertices in U_B. It is easy to show that the neighborhood of U_B is in A'_+ and also that each $a \in A'_+$ prefers its partner in L to $b \in U_B$. This is because $(b^-_\ell, b^+_r) \in S_i$ and b^+_r is b_ℓ's least preferred neighbor, thus b_ℓ must have been rejected by all its more preferred neighbors in our algorithm, i.e., every neighbor a^+_r of b_ℓ received a proposal from b^-_ℓ. Since a_r prefers superscript $-$ neighbors to superscript $+$ neighbors, this means $(c^-_\ell, a^+_r) \in S_i$ for some neighbor c^-_ℓ that a_r prefers to b^-_ℓ, i.e., a prefers c to b. Thus $a \in A'_+$.

We have $A'_+ = Z_A \cup (A'_+ \setminus Z_A)$ and $A'_+ \setminus Z_A \subseteq A_+$ (by the while-loop termination condition). Lemma 22 and Lemma 24 showed that for $a \in Z_A \cup (A_+ \cap A'_+)$, we have $M(a) \succeq_a L(a)$, i.e., a likes $M(a)$ at least as much as $L(a)$, and we showed above that each $a \in A'_+$ prefers $L(a)$ to b. Thus $\mathsf{wt}_M(a, b) = 0$. Since we set $\alpha_a = 0$ for $a \in Z_A$ and $\alpha_a = 1$ for $a \in A_+$, we have $\alpha_a + \alpha_b \geq 0 = \mathsf{wt}_M(a, b)$.

We now need to show $\alpha_a + \alpha_b \geq \mathsf{wt}_M(a, b)$ holds for all edges (a, b) in $G \setminus U_B$. Recall the witness $\vec{\gamma}$ defined in the proof of Theorem 18 to show the popularity of M in the subgraph $G \setminus U_B$. Observe that it is only for vertices u in Z that we have $\alpha_u \neq \gamma_u$. Moreover, $\alpha_a > \gamma_a$ for $a \in Z_A$. Thus we only have to worry about edges (a, b) in $G \setminus U_B$ where $b \in Z_B$ and check that $\mathsf{wt}_M(a, b) \leq \alpha_a + \alpha_b$. All other edges in $G \setminus U_B$ are covered by $\vec{\alpha}$ (since $\vec{\gamma}$ covers these edges). Let $b \in Z_B \subseteq B_+ \cap B'_-$.

1. Suppose $a \in U_A \cup Z_A$. For any $(a, b) \in (U_A \cup A_-) \times B_+$, we have $\mathsf{wt}_M(a, b) \le \gamma_a + \gamma_b \le 0 + 1$. Since $\mathsf{wt}_M(a, b)$ is an even number, this means $\mathsf{wt}_M(a, b) \le 0 = \alpha_a + \alpha_b$.
2. Suppose $a \in A_- \setminus Z_A$. Then $a \in A'_-$ by the termination condition of the while-loop in our algorithm. Since $\mathsf{wt}_L(x, y) \le \beta_x + \beta_y = -2$ for every edge $(x, y) \in A'_- \times B'_-$, it follows that $b \in Z_B \subseteq B'_-$ prefers $L(b)$ to a and similarly, $a \in A'_-$ prefers $L(a)$ to b.
   We know from Lemma 22 that $M(b) = L(b)$, so b prefers $M(b)$ to a. We know from Lemma 23 that $M(a) \succeq_a L(a)$ (i.e., a likes $M(a)$ at least as much as $L(a)$), so a prefers $M(a)$ to b. Thus $\mathsf{wt}_M(a, b) = -2 < \alpha_a + \alpha_b$ since $\alpha_a = -1$ and $\alpha_b = 0$.
3. Suppose $a \in A_+$. There are two sub-cases here: (i) $a \in A'_-$ and (ii) $a \in A'_+$. In sub-case (i), $\mathsf{wt}_L(a, b) \le \beta_a + \beta_b = -2$. Since $M(b) = L(b)$ (by Lemma 22), it means that b prefers $M(b)$ to a. Hence $\mathsf{wt}_M(a, b) \le 0 < \alpha_a + \alpha_b$ since $\alpha_a = 1$ and $\alpha_b = 0$ here.
   Consider sub-case (ii). We have $\mathsf{wt}_L(a, b) \le \beta_a + \beta_b = 0$. So either (1) b prefers $L(b)$ to a or (2) a prefers $L(a)$ to b. In case (1), we have $\mathsf{wt}_M(a, b) \le 0$ since $M(b) = L(b)$ (by Lemma 22). In case (2) also, we have $\mathsf{wt}_M(a, b) \le 0$ since $M(a) \succeq_a L(a)$ (by Lemma 24). So in both cases we have: $\mathsf{wt}_M(a, b) \le 0 < \alpha_a + \alpha_b$ since $\alpha_a = 1$ and $\alpha_b = 0$ here.

Thus $\vec{\alpha}$ is a witness of M's popularity (by Theorem 6). So M is popular in G. ◀

Since M is A-popular, Theorem 25 immediately implies that M is fully popular in G. Moreover, M is a *max-size* fully popular matching in G, as shown below.

▶ **Lemma 26.** *The matching M is a max-size fully popular matching in G.*

Proof. Observe that $U_A \cup U_B$ is the set of vertices left unmatched in the matching M. We claim the vertices in $U_A \cup U_B$ are left unmatched in any fully popular matching N. This claim holds because the matching S_i is the $(A_L \cup B_L)$-optimal matching in the lattice $\mathcal{D}_i$. Thus if $(a^-_\ell, a^+_r) \in S_i$, i.e., if a_ℓ is matched to its least preferred neighbor a^+_r in S_i then a_ℓ has to be matched to a^+_r in the realization N^*_α of N as well (for any witness $\vec{\alpha}$ of N), i.e., $(a^-_\ell, a^+_r) \in N^*_\alpha$; equivalently, a is left unmatched in N (after removing self-loops from N). Similarly, if $(b^-_\ell, b^+_r) \in S_i$ then b_ℓ has to be matched to its least preferred neighbor b^+_r in N^*_α as well, i.e., $(b^-_\ell, b^+_r) \in N^*_\alpha$; equivalently, b is left unmatched in N. ◀

Running time of the algorithm. The set of popular edges can be computed in linear time [7]. Similarly, the set of valid edges can be computed in linear time [2]. Gale-Shapley algorithm and in particular, the variant of Gale-Shapley algorithm used here – to compute a stable matching that avoids all forbidden edges – can be implemented to run in linear time [13]. Hence it can be shown that Algorithm 1 runs in linear time. Thus Theorem 4 stated in Section 1 follows.

References

1 A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

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

3 S. Baswana, P. P. Chakrabarti, S. Chandran, Y. Kanoria, and U. Patange. Centralized admissions for engineering colleges in india. *INFORMS Journal on Applied Analytics*, 49(5):338–354, 2019.

4 Canadian Resident Matching Service. How the matching algorithm works. `http://carms.ca/algorithm.htm`.

5 M.-J.-A.-N. de C. (Marquis de) Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie Royale, 1785.

6 Condorcet method. https://en.wikipedia.org/wiki/Condorcet_method.

7 Á. Cseh and T. Kavitha. Popular edges and dominant matchings. *Mathematical Programming*, 172(1):209–229, 2018.

8 Y. Faenza and T. Kavitha. Quasi-popular matchings, optimality, and extended formulations. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 325–344, 2020.

9 Y. Faenza, T. Kavitha, V. Powers, and X. Zhang. Popular matchings and limits to tractability. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2790–2809, 2019.

10 D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

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

12 P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Science*, 20:166–173, 1975.

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

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

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

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

17 T. Kavitha. Popular half-integral matchings. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 22:1–22:13, 2016.

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

19 D. F. Manlove and C. Sng. Popular matchings in the capacitated house allocation problem. In *Proceedings of the 14th Annual European Symposium on Algorithms (ESA)*, pages 492–503, 2006.

20 J. Mestre. Weighted popular matchings. *ACM Transactions on Algorithms*, 10(1):2:1–2:16, 2014.

21 National Resident Matching Program. Why the match? http://www.nrmp.org/whythematch.pdf.