An input to the Popular Matching problem, in the roommates setting (as opposed to the marriage setting), consists of a graph $G$ (not necessarily bipartite) where each vertex ranks its neighbors in strict order, known as its preference. In the Popular Matching problem the objective is to test whether there exists a matching $M^*$ such that there is no matching $M$ where more vertices prefer their matched status in $M$ (in terms of their preferences) over their matched status in $M^*$. In this article, we settle the computational complexity of the Popular Matching problem in the roommates setting by showing that the problem is NP-complete. Thus, we resolve an open question that has been repeatedly and explicitly asked over the last decade.

CCS Concepts: • Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability;

Additional Key Words and Phrases: Popular matching, NP-hard

1 INTRODUCTION

Matching problems with preferences are ubiquitous in everyday-life scenarios. They arise in applications such as the assignment of students to universities, doctors to hospitals, students to campus housing, pairing up police officers, kidney donor-recipient pairs, and so on. The common theme is that individuals have preferences over the possible outcomes, and the task is to find a matching of the participants that is in some sense optimal with respect to these preferences. In this arti-
In this paper, we study the computational complexity of computing one such solution concept, namely the Popular Matching problem. The input to the Popular Matching problem consists of a graph on \( n \) vertices and the preferences of the vertices represented as a ranked list of the neighbors of every vertex, said to be the preference list of the vertex. The goal is to find a popular matching—a matching that is preferred over any other matching (in terms of the preference lists) by at least half of the vertices in the graph.

Popular matching finds real-life applications in avenues as diverse as the organ-donor exchange markets, spectrum sharing in cellular networks, barter exchanges, to just name a few [27, 32]. Specifically, situations in which a stable matching—a matching that does not admit a blocking edge, i.e., an edge whose endpoints prefer each other to their respective “situation” in the current matching—is too restrictive, popular matching finds applicability. We note that a stable matching is the smallest sized popular matching. A simple “vote counting” argument toward this is as follows. Let \( X \) denote a popular matching that is smaller than a stable matching, denoted by \( S \). Then, the symmetric difference between the two matchings must contain a path that starts and ends with an \( S \)-edge and alternates between \( X \) and \( S \)-edges. Let \( P = u_1 v_1 \ldots u_i v_i \ldots u_n v_n \) denote such a path, where set of edges \( \{u_i v_i : 1 \leq i \leq n\} \subseteq S \) and \( \{v_i u_{i+1} : 1 \leq i \leq n - 1\} \subseteq X \). If we were to compare the preferences of the vertices in \( P \) toward the matchings \( X \) and \( S \), then we observe that both the end points of an \( X \)-edge cannot prefer \( X \) over \( S \); else that will be a blocking edge with respect to \( S \), contradiction to the stability of \( S \). Thus, \( S \) gets the vote of at least half of the endpoints of the \( X \)-edges, in addition to that of \( u_1 \) and \( v_n \) who are only matched in \( S \). Hence, \( S \) gets at least \( n + 1 \) votes among the \( 2n \) vertices in \( P \). Thus, the matching \( X' = X \oplus P \) (consisting of the \( S \)-edges in \( P \) and all the \( X \)-edges in \( E(G) \setminus E(P) \)) is strictly more popular than \( X \) in \( G \), a contradiction to the popularity of \( X \). So for applications where it is desirable to have matchings of larger size than a stable matching—for instance, allocating projects to students, or pairing up police officers, where the absence of blocking edges is not mandatory—popular matching may be a suitable alternative. The notion of popularity captures a natural relaxation of the notion of stability: Blocking edges are permitted but the matching, nevertheless, has “global stability.”

To define the Popular Matching problem formally, we first need few definitions. Let \( G \) denote a graph with vertex set \( V(G) \) and edge set \( E(G) \). Let \( N_G(v) \) denote the neighborhood of a vertex \( v \in V(G) \). Given a vertex \( v \in V(G) \), a preference list of \( v \) in \( G \) is a bijective function \( \ell_v : N_G(v) \rightarrow \{1, 2, \ldots, |N_G(v)|\} \). Informally, the smaller the number a vertex \( v \in V(G) \) assigns to a vertex \( u \in N_G(v) \), the more \( v \) prefers to be matched to \( u \). In particular, for all \( u, w \in N_G(v) \), if \( \ell_v(u) < \ell_v(w) \), then \( v \) prefers \( u \) over \( w \). A matching \( M \) in \( G \) is a subset of \( E(G) \) whose edges are pairwise disjoint. We say that a vertex \( v \in V(G) \) is matched by a matching \( M \) if there exists a (unique) vertex \( u \in V(G) \) such that \( \{u, v\} \in M \), which we denote by \( u = M(v) \).

In literature, the terminology related to Popular Matching is closely related to that of the Stable Marriage problem. When the input graph is (bipartite) arbitrary, the instance is said to be that of the (stable marriage) roommates setting of the problem. We denote an instance of Popular Matching (in the roommates setting) by \( I = (G, L = \{\ell_v : v \in V(G)\}) \). Formally, the notion of preference over matchings is defined as follows. Given two matchings in \( G \), denoted by \( M \) and \( M' \), we say that a vertex \( v \in V(G) \) prefers \( M \) over \( M' \) if one of the following conditions is satisfied: (i) \( v \) in matched by \( M \) but not matched by \( M' \); (ii) \( v \) is matched by both \( M \) and \( M' \), and \( \ell_v(M(v)) < \ell_v(M'(v)) \). We say that \( M' \) is more popular than \( M \), if the number of vertices that prefer \( M' \) to \( M \) exceeds the number of vertices that prefer \( M \) to \( M' \). A matching \( M \) is popular if and only if there is no matching \( M' \) that is more popular than \( M \). In the decision version of the Popular Matching problem, given an instance \( I = (G, L = \{\ell_v : v \in V(G)\}) \), the question is whether there exists a popular matching? There exists a rather simple instance—a triangle with cyclic preferences—presented in Figure 1, for which the answer to the question is “no.” Any maximal matching in this...
Popular Matching in Roommates Setting Is NP-hard

Fig. 1. A simple example of a No-instance of Popular Matching. The values on the edges denote preferences.

graph consists of exactly one edge and no matter which edge we pick as the matching, there is another matching (an edge) that is preferred by two vertices over one. Consequently, this instance cannot have a popular matching. It is worth noting here, that this object plays a crucial role in our gadget construction, Section 4, that yields the NP-hardness result for Popular Matching. We discuss its role in “causing trouble” toward finding a popular matching in the reduced instance.

History of the problem and our result. The provenance of the notion of a popular matching can be dated to the work of Condorcet in 1785 on the subject of a Condorcet winner [9]. In the last century, however, the notion was introduced as the majority assignment by Gärdenfors [12] in 1975. Anecdotal retelling ascribes the coinage of the term “popular matching” and the associated question of whether there exists a polynomial time algorithm for Popular Matching in the “housing allocation” setting, to Robert Irving during a talk in University of Glasgow in 2002 [27, p. 333]. In 2005, Abraham et al. [1] (also see Reference [2]) were the first to discuss an efficient algorithm for computing a popular matching albeit for the case where the graph is bipartite and only the vertices in one of the partitions have a preference list, a setting known as the housing allocation. The persuasive motivation and elegant analysis of Abraham et al. led to a spate of papers on popular matching [3, 16, 17, 21–24, 26] covering diverse settings that include strict preferences as well as one with ties. It is well known that when the input graph is bipartite and the preferences are strict—the stable marriage setting—Popular Matching problem can always be decided affirmatively in polynomial time [15]. It is equally well known that when the graph is arbitrary, the computational complexity of deciding whether a popular matching exists is unknown. In particular, whether Popular Matching is NP-hard has been repeatedly, explicitly asked as an open question over the past decade [3, 5, 6, 8, 14, 15, 17–19, 24, 25, 27, 31]. Indeed, it has been stated as one of the main open problem in the area (see the aforementioned citations). In this article, we settle this question by proving the following result.

Theorem 1. The Popular Matching problem is NP-complete.¹

Related results. Chung [4] was the first to study the Popular Matching problem in arbitrary graphs (i.e., the roommates setting). He observed that every stable matching is a popular matching. In the midst of a long series of articles, the issue of the computational complexity of Popular Matching in an arbitrary graph remained unsettled, leading various researchers to devise notions such as the unpopularity factor and unpopularity margin [14, 16, 28] in the hope of capturing the essence of popular matchings. A solution concept that emerged from this search is the maximum sized popular matching, motivated by the fact that unlike stable matchings (as given by the Rural Hospital Theorem [29]), all popular matchings in an instance do not match the same set of vertices or even have the same size. Thus, it is natural to focus on the size of a popular matching. There

¹After a preprint of this article appeared on arXiv, Kavitha [20] also published a pre-print showing this theorem. Her reduction is quite different from this article, and it also appears in the proceedings of SODA 2019 [10].
is a series of papers that focus on the MAX Sized Popular Matching problem in bipartite graphs (without ties in the preference lists) [8, 15, 18] and (with ties) [7]. When preferences are strict, there are various polynomial time algorithms that solve MAX Sized Popular Matching in bipartite graphs: Huang and Kavitha [14] give an \(O(mn)\) algorithm that is improved by Kavitha to \(O(m)\) [18], where \(m\) and \(n\) denote the number of edges in the bipartite graph and the size of the smaller vertex partition, respectively. In the presence of ties (even on one side), the MAX Sized Popular Matching was shown to be \(NP\)-hard [7]. It is worth noting that every stable matching is popular, but the converse is not true. As a consequence of the former, every bipartite graph has a popular matching that is computable in polynomial-time, because it has a stable matching computable by the famous Gale-Shapley algorithm described by the eponymous authors in their seminal paper [11].

2 PRELIMINARIES

Standard Definitions and Our Notation. Given a graph \(G\), we let \(V(G)\) and \(E(G)\) denote the vertex set and edge set of \(G\), respectively. Throughout the article, we consider undirected simple graphs. We view an edge as a set and edge set of \(G\). A triangle in \(G\) is a cycle in \(G\) on exactly three vertices. The neighborhood of a vertex \(v \in V(G)\) in \(G\) is denoted by \(N_G(v) = \{u \in V(G) : (u,v) \in E(G)\}\), and the set of edges incident to \(v\) in \(G\) is denoted by \(E_G(v)\). Given a vertex \(v \in V(G)\), a preference list of \(v\) in \(G\) is a bijective function \(\ell_v : N_G(v) \rightarrow \{1,2,\ldots,|N_G(v)|\}\). Informally, the smaller the number a vertex \(v \in V(G)\) assigns to a vertex \(u \in N_G(v)\), the more \(v\) prefers to be matched to \(u\).

In particular, for all \(u, w \in N_G(v)\), if \(\ell_v(u) < \ell_v(w)\), then \(v\) prefers \(u\) over \(w\). A matching \(M\) in \(G\) is a subset of \(E(G)\) whose edges are pairwise disjoint. We say that a vertex \(v \in V(G)\) is matched by a matching \(M\) if there exists a (unique) vertex \(u \in V(G)\) such that \(\{u,v\} \in M\), which we denote by \(u = M(v)\). Moreover, \(M\) is maximal if there is no edge in \(E(G)\) such that both endpoints of that edge are not matched by \(M\).

We denote an instance of Popular Matching (in the roommates setting) by \(I = (G, L = \{\ell_v : v \in V(G)\})\). A vertex \(v \in V(G)\) prefers a matching \(M\) over a matching \(M'\) if its “status” in \(M\) is better than the one in \(M'\), where being not matched is the least preferred status. Formally, the notion of preference over matchings is defined as follows.

**Definition 2.1.** Let \(I = (G, L = \{\ell_v : v \in V(G)\})\) be an instance of Popular Matching. Given two matchings in \(G\), denoted by \(M\) and \(M'\), we say that a vertex \(v \in V(G)\) prefers \(M\) over \(M'\) if one of the following conditions is satisfied: (i) \(v\) is matched by \(M\) but not matched by \(M'\); (ii) \(v\) is matched by both \(M\) and \(M'\), and \(\ell_v(M(v)) < \ell_v(M'(v))\). The number of vertices in \(V(G)\) that prefer \(M\) over \(M'\) is denoted by \(\text{vote}(M, M')\).

For notational convenience, given a vertex \(v \in V(G)\), we denote \(\ell_v(v) = |N_G(v)| + 1\), and given a matching \(M\) where \(v\) is not matched, we use the notation \(M(v) = v\). Then, for example, the first condition in Definition 2.1 is subsumed by the second one. We now also formally define the notion of popularity.

**Definition 2.2.** Let \(I = (G, L = \{\ell_v : v \in V(G)\})\) be an instance of Popular Matching. We say that a matching \(M\) in \(G\) is popular if \(\text{vote}(M', M) - \text{vote}(M, M') \leq 0\) for any other matching \(M'\) in \(G\).

Intuitively, the meaning of the definition above is that when the vertices are asked whether we should replace \(M\) by \(M'\), for any other matching \(M'\), the number of vertices that will vote against the swap is at least as large as the number of vertices that will vote in favor of it. Let us recall that in the Popular Matching problem, the objective is to decide whether there exists a popular matching.
Given a graph $G$, we say that a vertex $v \in V(G)$ covers an edge $e \in E(G)$ if $v$ is incident to $e$, that is, $v \in e$. A vertex cover $U$ in $G$ is a subset of $V(G)$ such that every edge in $E(G)$ is covered by at least one vertex in $U$. In the VERTEX COVER problem, we are given a graph $G$ and an integer $k$, and the objective is to decide whether $G$ has a vertex cover of size at most $k$.

**Known characterization of popular matchings.** We need to present (known) definitions of a labeling of the edges in $E(G)$ as well as of a special graph derived from $G$ and a matching $M$ in $G$, which will give rise to a characterization of popular matchings.

**Definition 2.3 (Definition 2 in Reference [15], Rephrased).** Let $I = (G, L = \{\ell_v : v \in V(G)\})$ be an instance of POPULAR MATCHING. Given a matching $M$ in $G$, the edge labeling $\text{label}_M : (E(G) \setminus M) \to \{-2, 0, +2\}$ is defined as follows:

$$\text{label}_M([u, v]) = \begin{cases} -2 & \text{if } \ell_u(M(u)) < \ell_u(v) \text{ and } \ell_v(M(v)) < \ell_v(u) \\ +2 & \text{if } \ell_u(M(u)) > \ell_u(v) \text{ and } \ell_v(M(v)) > \ell_v(u) \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, an edge in the definition above is assigned $-2$ if both its endpoints do not prefer being matched to each other over their status in $M$, and it is assigned $+2$ if both its endpoints prefer being matched to each other over their status in $M$.

**Definition 2.4 ([15]).** Let $I = (G, L = \{\ell_v : v \in V(G)\})$ be an instance of POPULAR MATCHING. Given a matching $M$ in $G$, the graph $G_M$ is the subgraph of $G$ with $V(G_M) = V(G)$ and $E(G_M) = \{[u, v] \in E(G) : [u, v] \in M \text{ or } \text{label}_M([u, v]) \neq -2\}$.

Before we can present the characterization, we need to define the notions of an alternating path and an alternating cycle in $G_M$. First, an alternating cycle in $G_M$ is a cycle in $G_M$ (with an even number of edges) such that if we traverse the edges of the cycle (in any direction), then every edge in $M$ is followed by an edge outside $M$, and every edge outside $M$ is followed by an edge in $M$. Similarly, an alternating path in $G_M$ is a path in $G_M$ such that if we traverse the edges of the path (in any direction), then every edge in $M$ is followed by an edge outside $M$ (with the exception of the last edge), and every edge outside $M$ is followed by an edge in $M$ (with the same exception), and in addition, if the edge incident to the first or last vertex on the path is not in $M$, then that vertex is not matched by $M$. Now, the characterization is given by the following proposition.

**Proposition 2.1 (Theorem 1 in Reference [15], Rephrased).** Let $I = (G, L = \{\ell_v : v \in V(G)\})$ be an instance of POPULAR MATCHING. A matching $M$ in $G$ is popular if and only if the following conditions hold in $G_M$.

- There is no alternating cycle in $G_M$ that contains one or more edge labeled $+2$ by $\text{label}_M$.
- There is no alternating path in $G_M$ that starts from a vertex not matched by $M$ and contains one or more edge labeled $+2$ by $\text{label}_M$.
- There is no alternating path in $G_M$ that contains two or more edges labeled $+2$ by $\text{label}_M$.

We note that Theorem 1 in Reference [15] holds for general graphs (its statement refers to bipartite graphs). The usefulness of Proposition 2.1 for us is that it will help us verify that the matching we construct when we prove the forward direction of the correctness of our reduction is indeed popular. Note that if we are to prove the popularity of a matching by using only the definition of popularity, then we need to compare the matching to potentially $n!$ other matchings. Thus, Proposition 2.1 will come in handy.
3 DEFINITION OF PARTITIONED VERTEX COVER

The correctness of our reduction will crucially rely on the fact that our source problem will not be VERTEX COVER, but a variant of it that we call PARTITIONED VERTEX COVER. This variant is defined as follows.

Problem definition. The input of PARTITIONED VERTEX COVER consists of a graph \( G \), a collection \( \mathcal{P} \) of pairwise disjoint sets of size two that induce edges in \( G \), and a collection \( \mathcal{T} \) of pairwise disjoint sets of size three of vertices that induce triangles in \( G \), such that every vertex in \( V(G) \) occurs in either a triangle in \( \mathcal{T} \) or an edge in \( \mathcal{P} \) (but not in both). In other words, \( \mathcal{T} \cup \mathcal{P} \) forms a partition of \( V(G) \) into sets of sizes 3 and 2.

To ease readability, we will refer to a set (edge) in \( \mathcal{P} \) as a pair and to a set in \( \mathcal{T} \) as a triple.

The objective of PARTITIONED VERTEX COVER is to decide whether \( G \) has a vertex cover \( U \) such that the two following conditions hold.

1. For every \( P \in \mathcal{P} \), it holds that \( |U \cap P| = 1 \).
2. For every \( T \in \mathcal{T} \), it holds that \( |U \cap T| = 2 \).

A vertex cover \( U \) with the properties above will be referred to as a solution.

Remark and Hardness. We remark that it will be crucial for us that (i) the sets in \( \mathcal{T} \cup \mathcal{P} \) are all pairwise disjoint, (ii) the maximum size of a set in \( \mathcal{T} \cup \mathcal{P} \) is only 3 and all but one of the vertices of a set in \( \mathcal{T} \cup \mathcal{P} \) must be selected, and (iii) all solutions must have the same size, where the implicit size requirement (that is, being of size exactly \(|\mathcal{P}| + 2|\mathcal{T}|\)) is automatically satisfied if Conditions 1 and 2 are satisfied.

Now, we claim that PARTITIONED VERTEX COVER is \( \text{NP} \)-hard. The correctness of this claim directly follows from a classic reduction from 3-SAT to VERTEX COVER (see, e.g., Reference [30]). For the sake of completeness, we present this reduction and argue formally why its output can be viewed correctly as an instance of PARTITIONED VERTEX COVER (rather than an instance of VERTEX COVER).

We begin our discussion by formally defining the 3-SAT problem: In an instance of 3-SAT, we are given a set of variables \( X \), and a formula encoded as a collection of clauses, denoted by \( C \), in the conjunctive normal form. Each clause \( C \in C \) is a set of exactly three literals, where each literal is either a variable \( x \in X \) or the negation of a variable \( x \in X \) that is denoted by \( \overline{x} \). A truth assignment \( \alpha : X \rightarrow \{T, F\} \) satisfies a literal \( \ell \) if either it is positive and assigned truth, or negative and assigned false. Now, \( \alpha \) satisfies a clause if it satisfies at least one of its literals, and it satisfies \( C \) if it satisfy every clause in \( C \). The objective is to decide whether there exists a truth assignment that satisfies \( C \).

Lemma 3.1. PARTITIONED VERTEX COVER is \( \text{NP} \)-hard.

Proof. We will reduce 3-SAT to PARTITIONED VERTEX COVER. Below we present the well-known classic reduction from 3-SAT to VERTEX COVER that shows that VERTEX COVER is \( \text{NP} \)-hard. Following that we argue how the instance outputted by the reduction can be viewed as an instance of PARTITIONED VERTEX COVER, and thus conclude the proof of Lemma 3.1.

Let \( I = (X, C) \). Then, we construct an instance reduction \( (I) = (G, k) \) of VERTEX COVER as follows. First, define \( k = |X| + 2|C| \). Now, for every \( x \in X \), add two new vertices \( v_x \) and \( v_{\overline{x}} \), along with the edge \( \{v_x, v_{\overline{x}}\} \), to \( G \). For every clause \( C = \{p, q, r\} \in C \), add three new vertices \( u_p^C, u_q^C \) and \( u_r^C \), along with the edges \( \{u_p^C, u_q^C\}, \{u_q^C, u_r^C\} \) and \( \{u_r^C, u_p^C\} \) to \( G \). Finally, for every \( C \in C \) and \( \ell \in C \), add the edge \( \{u_r^C, v_\ell\} \) to \( G \). It is easy to verify and has been shown in Reference [30] that

\[ \text{That is, for all } \{x, y, z\} \in \mathcal{T}, \text{ we have that } \{x, y\}, \{y, z\}, \{z, x\} \in E(G). \]
I has a satisfying assignment if and only if the G has a vertex cover of size at most k where (G, k) = reduction(I).

Next, we discuss how this reduction also creates an instance of Partitioned Vertex Cover. Given the output instance (G, k) of the reduction, we define \( P = \{(u_x, v_x) : x \in X\} \) and \( T = \{u^C_p, u^C_q, u^C_r : C = \{p, q, r\} \in C\} \).

**Claim 3.1.1.** Let \( I = (X, C) \) be an instance of 3-SAT. Then, I has a satisfying assignment if and only if \( J = (G, P, T, k) \) has a partitioned vertex cover of size at most k.

Proof. Clearly, the sets in \( P \cup T \) are pairwise disjoint, every set in \( P \) is an edge in G, and every set in \( T \) induces a triangle in G. Moreover, every vertex cover of G must select at least one vertex of each edge in \( E(G) \), and at least two vertices of every triangle in G. Since \( |P| = |X| \) and \( |T| = 2|C| \), this means that G has a vertex cover of size at most k if and only if G has a vertex cover of size exactly k, and the latter statement holds if and only if \( (G, P, T, k) \) has a solution.

Thus, the lemma is proved.

\[ \square \]

4 REDUCING PARTITIONED VERTEX COVER TO POPULAR MATCHING

Let \( I = (G, P, T) \) be an instance of Partitioned Vertex Cover. In this section, we construct an instance reduction(I) = (H, L = \{\ell_v : v \in V(H)\}) of Popular Matching. Note that, to avoid confusion, we denote the graph in reduction(I) by H rather than G, since the latter already denotes the graph in I. We remark that the Edge Coverage gadget below is in fact the entire reduction from (standard) Vertex Cover to an optimization variant of Popular Matching recently given by Kavitha [19] (in that context, we will use notation consistent with this work). Our two other gadgets are completely new. After describing the Edge Coverage gadget, we briefly discuss why that alone does not yield the hardness result for Popular Matching. In particular, this brief discussion sheds light on the jump in understanding the Popular Matching problem that we had to perform to employ this known gadget (or any other similar gadget in the literature on popular matchings) to prove the hardness of Popular Matching.

4.1 Edge Coverage
For every vertex \( i \in V(G) \), we add four new vertices (to H), denoted by \( a_i, b_i, c_i \) and \( d_i \). In addition, we add the edges \{a_i, a_j\}, \{a_i, b_j\}, \{a_i, c_i\}, and \{b_i, c_i\} (see Figure 2). Now, for every edge \( e = \{i, j\} \in E(G) \), we add two vertices, \( u^e_i \) and \( u^e_j \), and the edges \{u^e_i, u^e_j\}, \{b_i, u^e_j\}, and \{b_j, u^e_i\}.

Let us now give a partial definition of the preference lists of the vertices added so far (see Figure 2). When we will add neighbors to some of these vertices, they will be appended to the end of these partial lists, and we will not change the values that we are about to define. For every vertex \( i \in V(G) \), we have the following definitions:

- **Vertex a_i**: \( \ell_{a_i}(b_i) = 1;\ \ell_{a_i}(c_i) = 2;\ \ell_{a_i}(d_i) = 3.\)
- **Vertex b_i**: \( \ell_{b_i}(a_i) = 1; \ \ell_{b_i} \) restricted to \( \{u^e_i : e \in E(G)\} \) is an arbitrary bijection into \( \{2, 3, \ldots, |E_G(i)| + 1\} \); \( \ell_{b_i}(c_i) = |N_G(i)| + 2.\)
- **Vertex c_i**: \( \ell_{c_i}(a_i) = 1; \ \ell_{c_i}(b_i) = 2.\)

\[ 3 \text{That is, every vertex in } \{u^e_i : e \in E_G(i)\} \text{ is assigned a unique integer from } \{2, 3, \ldots, |E_G(i)| + 1\}, \text{ and it is immaterial to us which bijection to choose to achieve this.} \]

ACM Transactions on Computation Theory, Vol. 13, No. 2, Article 9. Publication date: March 2021.
Fig. 2. The Edge Coverage gadget. Here, $x \in \{2, 3, \ldots, |N_G(i)| + 1\}$ and $y \in \{2, 3, \ldots, |N_G(j)| + 1\}$.

- **Vertex $d_i$:** $\ell_{d_i}(a_i) = 1$.
- **Vertex $u^e_i$ for any $e = \{i, j\} \in E_G(i)$:** $\ell_{u^e_i}(u^e_f) = 1; \ell_{u^e_i}(b_i) = 2$.

This completes the description of the Edge Coverage gadget.

**Intuition.** The Edge Coverage gadget aims to encode (as we will see in Section 5) the selection of a vertex as follows. In every popular matching $M$, either $\{a_i, b_i\} \in M$ or both $\{a_i, d_i\} \in M$ and $\{b_i, c_i\} \in M$. The special choice of the preferences ensure this, where the first choice indicates that $i$ is present in the vertex cover encoded by $M$, while the second choice indicates that $i$ is not present in this vertex cover. Intuitively, $a_i$ and $b_i$ prefer each other the most, but if we choose to match them, we “leave out” both $d_i$ and $c_i$, which gives rise to the two configurations as above. Then, the addition of $u^e_i$ and $u^e_f$, which prefer each other the most, and which are inserted in the “middle” of $b_i$’s and $b_j$’s lists, respectively, will ensure that every edge is indeed covered. To establish this last claim, it will also be important that $c_i$ prefers $a_i$ over $b_i$—this will allow us to “move” from the configuration of having $\{a_i, d_i\}, \{b_i, c_i\} \in M$ and $\{a_j, d_j\}, \{b_j, c_j\} \in M$ to one where $c_i$ and $c_j$ are matched to $a_i$ and $a_j$, respectively, when we try to exhibit a matching more popular than $M$.

While this gadget, already given by Kavitha [19], is very useful to us, but it cannot enforce popular matchings to favor the selection of $\{b_i, c_i\} \in M$ and $\{a_i, d_i\} \in M$ over $\{a_i, b_i\} \in M$. In other words, this gadget does not help us, in any way, to force the encoded vertex cover to be as small as possible. (We remark that Kavitha [19] considers a variant of Popular Matching where the matchings should be as large as possible, and hence the inherent difficulty of the problem is circumvented.) By considering Partitioned Vertex Cover rather than Vertex Cover, we do not need to deal with such “size optimization” constraint anymore. However, we now need to handle the constraints imposed by $\mathcal{P}$ and $\mathcal{T}$. Nevertheless, these two sets are very structured as explained in Section 3 (in sharp contrast to, say, an arbitrary instance of 3-SAT). In fact, every detail of the gadgets described next is carefully tailored to exploit the extra structural properties of Partitioned Vertex Cover as much as possible, as will be made clear in Section 5.

### 4.2 Pair Selector

For every pair $\{i, j\} \in \mathcal{P}$ with $i < j$, we add two new vertices (to $H$), denoted by $f_{ij}$ and $f_{ji}$, along with the edges $\{d_i, f_{ij}\}, \{f_{ij}, c_i\}, \{d_j, f_{ji}\}$ and $\{f_{ji}, c_j\}$ (see Figure 3). In addition, we insert the edges $\{c_i, d_j\}$ and $\{c_j, d_i\}$.
We update the preference lists of the vertices as follows (see Figure 3):

- **Vertex** $c_i$: $\ell_{c_i}(f_{ji}) = 3$; $\ell_{c_i}(d_j) = 4$.
- **Vertex** $c_j$: $\ell_{c_j}(f_{ij}) = 3$; $\ell_{c_j}(d_i) = 4$.
- **Vertex** $d_i$: $\ell_{d_i}(c_i) = 2$; $\ell_{d_i}(f_{ij}) = 3$.
- **Vertex** $d_j$: $\ell_{d_j}(c_j) = 2$; $\ell_{d_j}(f_{ji}) = 3$.
- **Vertex** $f_{ij}$: $\ell_{f_{ij}}(d_i) = 1$; $\ell_{f_{ij}}(c_i) = 2$.
- **Vertex** $f_{ji}$: $\ell_{f_{ji}}(d_j) = 1$; $\ell_{f_{ji}}(c_j) = 2$.

Note that the definition above is valid, since no vertex in $V(G)$ participates in more than one pair, and hence no integer is assigned by any function $\ell_{\diamond}$ more than once. This completes the description of the Pair Selector gadget.

**Intuition.** First, we point out that the Pair Selector gadget is symmetric in the sense that if we swap $i$ and $j$, then we obtain an isomorphic structure also with respect to preferences. Thus, the gadget is well (uniquely) defined even if we drop the requirement “with $i < j$” above. We will use this symmetry when we prove the correctness of our reduction.

To gain some deeper understanding of this gadget, let us recall that in Partitioned Vertex Cover, exactly one vertex among $\{i, j\}$ must be selected. We already know that the Edge Coverage gadget is meant to ensure that at least one vertex among $\{i, j\}$ is selected. Hence, we only need to ensure that not both $i$ and $j$ are selected. However, if both $i$ and $j$ are selected, then both $c_i$ and $d_j$ are left not matched. Then, the preferences on the triangle on $\{c_i, d_j, f_{ji}\}$ are chosen specifically to “cause trouble”—no matter which edge of this triangle will be picked by the matching, we can replace it by a different edge on this triangle to exhibit a more popular matching. For example, if we pick $\{c_i, f_{ji}\}$, then $d_j$ is left not matched, while $f_{ji}$ prefers $d_j$ over $c_i$. This means that by replacing $\{c_i, f_{ji}\}$ by $\{f_{ji}, d_j\}$, we make both $f_{ji}$ and $d_j$ more satisfied, while only $c_i$ becomes less satisfied (no other vertex in $H$ is affected by the swap).
In light of the swap above, it may appear as if it would have been sufficient to keep the triangle on \{c_i, d_j, f_{ij}\}, while removing the triangle on \{c_j, d_i, f_{ij}\} from the gadget. However, without the second triangle, the proof of the forward direction fails—the matching attempted to construct from a vertex cover will not be popular. In particular, by having the second triangle as well, we will always be able to match all of the vertices in \(H\), and hence avoid the need to consider the second condition in Proposition 2.1. Again, we stress that the second triangle is not meant to ease the proof, but that without it the forward direction of the proof fails. It is also worth to note here that only having these two triangles is not sufficient, but the exact “orientation” of their preferences is crucial. In particular, if we changed the orientation of only one of the triangles—for example, if we made \(c_i\) prefer \(d_j\) over \(f_{ji}\), \(d_j\) prefer \(f_{ji}\) over \(c_i\), and \(f_{ji}\) prefer \(c_i\) over \(d_j\)—then the gadget would have no longer been symmetric, and the proof of the forward direction would have failed. Roughly speaking, the two triangles on \{c_i, d_j, f_{ji}\} and \{c_j, d_i, f_{ij}\} “work together” to prevent the existence of alternating cycles that must not exist by Proposition 2.1. Deeper coordination is required in the next gadget, and we will elaborate on it more when we explain the intuition behind that gadget.

### 4.3 Triple Selector

For every triple \(\{i, j, k\} \in T\) with \(i < j < k\), we add six new edges (to \(H\)): \(\{d_i, d_j\}, \{d_j, d_k\}, \{d_k, d_i\}, \{c_i, c_j\}, \{c_j, c_k\}\) and \(\{c_k, c_i\}\) (see Figure 4).
We update the preference lists of the vertices as follows (see Figure 4):

- **Vertex** $c_i$: $\ell_{c_i}(c_k) = 3$; $\ell_{c_i}(c_j) = 4$.
- **Vertex** $c_j$: $\ell_{c_j}(c_i) = 3$; $\ell_{c_j}(c_k) = 4$.
- **Vertex** $c_k$: $\ell_{c_k}(c_j) = 3$; $\ell_{c_k}(c_i) = 4$.
- **Vertex** $d_i$: $\ell_{d_i}(d_j) = 2$; $\ell_{d_i}(d_k) = 3$.
- **Vertex** $d_j$: $\ell_{d_j}(d_k) = 2$; $\ell_{d_j}(d_i) = 3$.
- **Vertex** $d_k$: $\ell_{d_k}(d_i) = 2$; $\ell_{d_k}(d_j) = 3$.

Note that the definition above is valid, since no vertex in $V(G)$ participates in both a pair and a triple, or in more than one triple, and hence no integer is assigned by any function $\ell_\nu$ more than once. This completes the description of the Triple Selector gadget.

**Intuition.** First, we point out that the Triple Selector gadget is symmetric with respect to cyclic shifts. That is, if we replace $j$ by $i$, $k$ by $j$, and $i$ by $k$, then we obtain an isomorphic structure also with respect to preferences. We will use this symmetry when we prove the correctness of our reduction.

To gain some deeper understanding of this gadget, let us recall that in Partitioned Vertex Cover, exactly two vertices among $\{i, j, k\}$ must be selected. We already know that the Edge Coverage gadget will ensure that at least two vertices among $\{i, j, k\}$ are selected (since $\{i, j, k\}$ induces a triangle in $G$ and to cover the edges of a triangle at least two of its vertices must be selected). Hence, we only need to ensure that not all of the vertices $i, j$ and $k$ are chosen specifically to “cause trouble” in a manner similar to the Pair Selector gadget—again, no matter which edge of this triangle will be picked by the matching, we can replace it by a different edge on this triangle to exhibit a more popular matching. For example, if we pick $\{d_i, d_j\}$, then $d_k$ is left not matched, while $d_j$ prefers $d_k$ over $d_i$. This means that by replacing $\{d_i, d_j\}$ by $\{d_j, d_k\}$, we make both $d_j$ and $d_k$ more satisfied, while only $d_i$ becomes less satisfied (no other vertex in $H$ is affected by the swap).

As in the case of the Pair Selector gadget, the inner triangle (in Figure 4) on $\{d_i, d_j, d_k\}$ is not sufficient—the forward direction of the proof fails without the outer triangle on $\{c_i, c_j, c_k\}$. Here, to make the forward direction go through, an additional idea is required. Roughly speaking, we need to have coordination between the triangles (recall that in the previous gadget, some coordination was also noted as a requirement to ensure symmetry, but here deeper coordination is required). Let us elaborate (in a non-formal manner) on the meaning of this coordination here. Specifically, we “orient” the inner triangle and the outer triangle in different directions. (Note that symmetry would have been achieved even if we would have oriented them in the same direction.)

By this, we mean that while in the inner triangle, $d_i$ prefers $d_j$ over $d_k$, $d_j$ prefers $d_k$ over $d_i$, and $d_k$ prefers $d_i$ over $d_j$, the same does not hold when we rename $d$ to be $c$—here, the direction is reversed, as $c_i$ prefers $c_k$ over $c_j$, $c_j$ prefers $c_i$ over $c_k$, and $c_k$ prefers $c_j$ over $c_i$. This reversal will come in handy when we prove the forward direction, as it will “block up” alternating cycles that must not exist by Proposition 2.1. Intuitively, the main insight is that if we try to improve the matching we will construct in the proof of the forward direction in a “clockwise direction,” then we can make two $d$-type vertices more satisfied and only one $d$-type vertex less satisfied, but at the same time, more $c$-type vertices become unsatisfied, and hence we overall do not gain more popularity. However, if we try to improve the matching in a “counter-clockwise direction,” then we can make two $c$-type vertices more satisfied and only one $c$-type vertex less satisfied, but at the same time, more $d$-type vertices become unsatisfied, and hence again we overall do not gain more popularity.
5 CORRECTNESS

In this section, we prove the correctness of our reduction. For the sake of clarity, the proof is divided into two subsections, corresponding to the forward and reverse directions. Together with Lemma 3.1, this proof will conclude the correctness of Theorem 1.

5.1 Forward Direction

Here, we prove that if there exists a solution to the instance \((G, \mathcal{P}, \mathcal{T})\) of Partitioned Vertex Cover, then there exists a popular matching in reduction \((I) = (H, L = \{\ell_v : v \in V(H)\})\). For this purpose, let us suppose that \(U\) is a solution to \((G, \mathcal{P}, \mathcal{T})\). In what follows, we first construct a matching \(M\) in \(H\). Then, we will show that the graph \(H_M\) (see Definition 2.4) satisfies several conditions of Proposition 2.1, which will eventually lead us to conclude that \(M\) is popular.

Construction of \(M\). The matching \(M\) is the union of the following sets.

- \(M_U = \{\{u^e_i, u^e_j\} : \{i, j\} \in E(G)\}\).
- For every \(\{i, j\} \in \mathcal{P}\) with \(i < j\), let \(x \in \{i, j\}\) be the vertex not in \(U\), and \(y \in \{i, j\}\) be the vertex in \(U\), and insert the edges \(\{a_x, d_x\}, \{b_x, c_x\}, \{a_y, b_y\}, \{f_{xy}, c_y\}\) and \(\{f_{yx}, d_y\}\) into \(M_P\) (see Figure 5).
- For every \(\{i, j, k\} \in \mathcal{T}\) with \(i < j < k\), let \(x \in \{i, j, k\}\) be the vertex not in \(U\), and \(y, z \in \{i, j, k\}\) be the two vertices in \(U\) such that \(d_x\) prefers \(d_y\) over \(d_z\), and insert the edges \(\{a_x, d_x\}, \{b_x, c_x\}, \{a_y, b_y\}, \{a_z, b_z\}, \{c_y, c_z\}\) and \(\{d_y, d_z\}\) into \(M_T\) (see Figure 6).

Since the sets in \(\mathcal{P} \cup \mathcal{T}\) are pairwise disjoint, the sets above are well (uniquely) defined. We also remark that the figures do not only capture the case where \(i = x\) due to the symmetry of our gadgets (i.e., if \(j = x\) or \(k = x\) in the case of a triple, then we obtain precisely the same figures).

Properties of \(H_M\). Let us start by observing that, since all vertices in \(H\) are matched by \(M\) and edges labeled \(-2\) have been deleted from \(H_M\), the following statement immediately holds.
Fig. 6. Edges shown in bold are inserted into $M$ (in Section 5.1). Edges labeled $-2$ by $\text{label}_M$ are highlighted in red, and edges labeled $+2$ by $\text{label}_M$ are highlighted in green.

**Observation 5.1.** There is no alternating path in $H_M$ that starts from a vertex not matched by $M$ and contains one or more edges labeled $+2$ by $\text{label}_M$.

We proceed to identify which edges in $H_M$ are labeled $+2$ by $\text{label}_M$.

**Lemma 5.1.** The set of edges labeled $+2$ by $\text{label}_M$ is $\{(a_i, b_i) : i \notin U\} \cup \{(a_i, c_i) : i \notin U\}$.

**Proof.** First, for all $i \notin U$, we have that $\{a_i, d_i\} \in M$ and $\{b_i, c_i\} \in M$. Since $a_i$ prefers both $b_i$ and $c_i$ over $d_i$, and both $b_i$ and $c_i$ prefer $a_i$ over each other, we have that all the edges in $\{(a_i, b_i) : i \notin U\} \cup \{(a_i, c_i) : i \notin U\}$ are labeled $+2$ by $\text{label}_M$. Next, we show that all other edges in $H$ are not labeled $+2$ by $\text{label}_M$, which will complete the proof.

Observe that for all $i \in U$, we have that $\{a_i, b_i\} \in M$, and since $\ell_{a_i}(b_i) = \ell_{b_i}(a_i) = 1$, this means that no edge incident to $a_i$ or $b_i$ can be labeled $+2$ by $\text{label}_M$. Similarly, for all $\{i, j\} \in E(G)$, we have that $\{u_i, u_j\} \in M$, and since $\ell_{u_i}(u_j) = \ell_{u_j}(u_i) = 1$, this means that no edge incident to $u_i$ or $u_j$ can be labeled $+2$ by $\text{label}_M$. Thus, no edge that belongs to an Edge Coverage gadget, excluding the edges in $\{(a_i, b_i) : i \notin U\} \cup \{(a_i, c_i) : i \notin U\}$, is labeled $+2$ by $\text{label}_M$.

Now, consider some pair $\{i, j\} \in P$ with $i < j$, and let $x \in \{i, j\}$ be the vertex not in $U$, and $y \in \{i, j\}$ be the vertex in $U$. Then, the edges $\{a_x, d_x\}$, $\{b_x, c_x\}$, $\{a_y, b_y\}$, $\{f_{xy}, c_y\}$, and $\{f_{yx}, d_y\}$ belong to $M$. However, $c_x$ prefers $b_x$ over both $f_{yx}$ and $d_y$, and $d_x$ prefers $a_x$ over both $f_{xy}$ and $c_u$, which means that none of the edges $\{c_x, f_{yx}\}$, $\{c_x, d_y\}$, $\{d_x, f_{xy}\}$, and $\{d_x, c_y\}$ is labeled $+2$ by $\text{label}_M$. 

ACM Transactions on Computation Theory, Vol. 13, No. 2, Article 9. Publication date: March 2021.
Finally, consider some triple \( \{i, j, k\} \in \mathcal{T} \) with \( i < j < k \), and let \( x \in \{i, j, k\} \) be the vertex not in \( U \), and \( y, z \in \{i, j, k\} \) be the two vertices in \( U \) such that \( d_x \) prefers \( d_y \) over \( d_z \). Then, the edges \( \{a_x, d_x\}, \{b_x, c_x\}, \{a_y, b_y\}, \{a_z, b_z\}, \{c_y, c_z\} \), and \( \{d_y, d_z\} \) belong to \( M \). However, \( c_x \) prefers \( b_x \) over both \( c_y \) and \( c_z \), and \( d_x \) prefers \( a_x \) over both \( d_y \) and \( d_z \), which means that none of the edges \( \{c_x, c_y\}, \{c_x, c_z\}, \{d_x, d_y\} \), and \( \{d_x, d_z\} \) is labeled +2 by \( \text{label}_M \). \( \square \)

Now, Lemma 5.1 directly implies the correctness of the following lemma.

**Lemma 5.2.** For any \( P \in \mathcal{P} \), the only edges labeled +2 by \( \text{label}_M \) in the Pair Selector gadget associated with \( P \) are \( \{a_x, b_x\} \) and \( \{a_x, c_x\} \) for the unique vertex \( x \in P \) that is not in \( U \). Similarly, for any \( T \in \mathcal{T} \), the only edges labeled +2 by \( \text{label}_M \) in the Triple Selector gadget associated with \( T \) are \( \{a_x, b_x\} \) and \( \{a_x, c_x\} \) for the unique vertex \( x \in T \) that is not in \( U \).

Having Lemma 5.2 at hand, we are ready to rule out the possibility of having a “bad” alternating path that is completely contained inside a Pair Selector gadget or a Triple Selector gadget.

**Lemma 5.3.** For any \( P \in \mathcal{P} \), there is no alternating path in \( H_M \) that contains two or more edges labeled +2 by \( \text{label}_M \) and that consists only of edges from the Pair Selector gadget associated with \( P \). Similarly, for any \( T \in \mathcal{T} \), there is no alternating path in \( H_M \) that contains two or more edges labeled +2 by \( \text{label}_M \) and that consists only of edges from the Triple Selector gadget associated with \( T \).

**Proof.** First, consider some pair \( P \in \mathcal{P} \). By Lemma 5.2, the only edges labeled +2 by \( M \) in the Pair Selector gadget associated with \( P \) are \( \{a_x, b_x\} \) and \( \{a_x, c_x\} \) for the unique vertex \( x \in P \) that is not in \( U \). However, these two edges are part of a triangle in \( H \), and therefore no alternating path can contain both of them together.

Second, consider some triple \( T \in \mathcal{T} \). By Lemma 5.2, the only edges labeled +2 by \( M \) in the Triple Selector gadget associated with \( T \) are \( \{a_x, b_x\} \) and \( \{a_x, c_x\} \) for the unique vertex \( x \in T \) that is not in \( U \). However, these two edges are again part of a triangle in \( H \), and therefore no alternating path can contain both of them together. \( \square \)

In the following two lemmas, we also rule out the possibly of having a “bad” alternating cycle that is completely contained inside a Pair Selector gadget or a Triple Selector gadget.

**Lemma 5.4.** For any \( P \in \mathcal{P} \), there is no alternating cycle in \( H_M \) that contains one or more edge labeled +2 by \( \text{label}_M \) and that consists only of edges from the Pair Selector gadget associated with \( P \).

**Proof.** Consider some pair \( P = \{i, j\} \in \mathcal{P} \) with \( i < j \), and let \( x \in \{i, j\} \) be the vertex not in \( U \), and \( y \in \{i, j\} \) be the vertex in \( U \). Suppose, by way of contradiction, that there exists an alternating cycle \( C \) in \( H_M \) that contains one or more edges labeled +2 by \( \text{label}_M \) and that consists only of edges from the Pair Selector gadget associated with \( P \).

First, suppose that \( \{a_x, c_x\} \in E(C) \). Then, since \( \{c_x, b_x\} \in M \), we have that \( \{c_x, b_x\} \in E(C) \). Since the only neighbor in the gadget of \( b_x \) apart from \( c_x \) is \( a_x \), we have that \( \{b_x, a_x\} \in E(C) \). However, we have thus “closed” a triangle, which contradicts the choice of \( C \) as an alternating cycle.

By Lemma 5.2 and since \( C \) contains one or more edges labeled +2 by \( \text{label}_M \), it must hold that \( \{a_x, b_x\} \in E(C) \). Then, since \( \{c_x, b_x\} \in M \), we have that \( \{c_x, b_x\} = \{b_x, a_x\} \) is a subpath of \( C \). Now, note that \( c_x \) prefers \( b_x \) over its two other neighbors in the gadget, and \( f_{yx} \) prefers \( d_y \) over \( c_x \). Therefore, \( \{c_x, f_{yx}\} \) is labeled −2 by \( \text{label}_M \), and hence it does not exist in \( H_M \). Thus, we also have that \( \{d_y, c_x\} \in E(C) \), and since \( \{d_y, f_{yx}\} \in M \), we have that \( \{f_{yx}, d_y\} = \{d_y, c_x\} = \{c_x, b_x\} = \{b_x, a_x\} \) is a subpath of \( C \) (see Figure 7). However, \( f_{yx} \) has no neighbor in \( H_M \) apart from \( d_y \), and therefore we have reached a contradiction to the choice of \( C \) as an alternating cycle. \( \square \)
LEMMA 5.5. For any $T \in \mathcal{T}$, there is no alternating cycle in $H_M$ that contains one or more edges labeled +2 by $\text{label}_M$ and that consists only of edges from the Triple Selector gadget associated with $T$.

PROOF. Consider some triple $T = \{i, j, k\} \in \mathcal{T}$ with $i < j < k$, and let $x \in \{i, j, k\}$ be the vertex not in $U$, and $y, z \in \{i, j, k\}$ be the two vertices in $U$ such that $d_x$ prefers $d_y$ over $d_z$. Suppose, by way of contradiction, that there exists an alternating cycle $C$ in $H_M$ that contains one or more edges labeled +2 by $\text{label}_M$ and that consists only of edges from the Triple Selector gadget associated with $T$.

First, suppose that $\{a_x, c_x\} \in E(C)$. Then, since $\{c_x, b_x\} \in M$, we have that $\{c_x, b_x\} \in E(C)$. Since the only neighbor in the gadget of $b_x$ apart from $c_x$ is $a_x$, we have that $\{b_x, a_x\} \in E(C)$. However, we have thus “closed” a triangle, which contradicts the choice of $C$ as an alternating cycle.

By Lemma 5.2 and since $C$ contains one or more edges labeled +2 by $\text{label}_M$, it must hold that $\{a_x, b_x\} \in E(C)$. Then, since $\{c_x, b_x\} \in M$ and $\{a_x, d_x\} \in M$, we have that $\{c_x, b_x\} - \{b_x, a_x\} - \{a_x, d_x\}$ is a subpath of $C$. Observe that $c_x$ prefers $b_x$ over $c_z$, and $c_z$ prefers $c_y$ over $c_x$. Moreover, $d_x$ prefers $a_x$ over $d_y$, and $d_y$ prefers $d_z$ over $d_x$. Therefore, both $\{c_x, c_z\}$ and $\{d_x, d_y\}$ are labeled $-2$ by $\text{label}_M$, which means that these two edges do not exist in $H_M$. Since the only neighbor of $c_x$ in the gadget except for $a_x$, $b_x$ and $c_z$ is $c_y$, and since the only neighbor of $d_x$ in the gadget except for $a_x$ and $d_y$ is $d_z$, we have that $\{c_y, c_x\}, \{d_x, d_z\} \in E(C)$. Since $\{c_z, c_y\}, \{d_z, d_y\} \in M$, this means that $\{c_z, c_y\} - \{c_y, c_x\} - \{c_x, b_x\} - \{b_x, a_x\} - \{a_x, d_x\} - \{d_x, d_z\} = \{d_x, d_y\}$ is a subpath of $C$. Now, since the only neighbor of $d_y$ in the gadget except for $d_x$ and $d_z$ is $a_y$, we have that $\{d_y, a_y\} \in E(C)$. Because $\{a_y, b_y\} \in M$, and since the only neighbor of $b_y$ in this gadget except for $a_y$ is $c_y$, this means that $\{c_z, c_y\} - \{c_y, c_x\} - \{c_x, b_x\} - \{b_x, a_x\} - \{a_x, d_x\} - \{d_x, d_z\} - \{d_x, d_y\} - \{d_y, a_y\} - \{a_y, b_y\} - \{b_y, c_y\}$ is a subpath of $C$ (see Figure 8). However, $c_y$ has three different neighbors on this path, which contradicts the choice of $C$ as an alternating cycle.

Fig. 7. The path constructed in the proof of Lemma 5.4, highlighted in yellow.

ACM Transactions on Computation Theory, Vol. 13, No. 2, Article 9. Publication date: March 2021.
In the following lemma, we consider a minimal counterexample to $M$ being a popular matching in $H$. Specifically, we consider a shortest “bad” alternating path and show that it cannot contain any vertex of the form $u^e_i$.

**Lemma 5.6.** Let $S$ be a shortest alternating path in $H_M$ that contains two or more edges labeled $+2$ by $\text{label}_M$. Then, $S$ does not contain the vertex $u^e_i$ for any $e \in E(G)$ and $i \in e$.

**Proof.** Suppose, by way of contradiction, that $S$ contains the vertex $u^e_i$ for some $e \in E(G)$ and $i \in e$. Then, since $\{u^e_i, u^e_j\} \in M$ for the vertex $j \in V(G)$ such that $e = \{i, j\}$, we have that $\{u^e_i, u^e_j\} \in E(S)$. Since $U$ is a vertex cover in $G$, at least one vertex in $\{i, j\}$ belongs to $U$, and let us suppose without loss of generality that this vertex is $i$. Then, $\{a_i, b_i\} \in M$, which means that $\text{label}_M(\{b_i, u^e_i\}) = -2$. Thus, $\{b_i, u^e_i\} \notin E(H_M)$. Since the only neighbors of $u^e_i$ in $H$ are $b_i$ and $u^e_j$, the only neighbors of $u^e_j$ in $H$ are $b_j$ and $u^e_i$, this implies that $u^e_i$ is an endpoint of $S$ and $\{u^e_i, b_j\} \in E(S)$. Since $u^e_j$ prefers $u^e_i$ over $b_j$, we have that $\text{label}_M(\{b_j, u^e_j\}) \neq +2$. Thus, by removing $\{u^e_i, u^e_j\}$ and $\{b_j, u^e_j\}$ from $S$, we obtain yet another alternating path in $H_M$ that contains two or more edges labeled $+2$ by $\text{label}_M$. This contradicts the choice of $S$ as the shortest alternating path in $H_M$ with this property. \qed
In the following lemma we prove that a “bad” alternating cycle cannot contain any vertex of the form $u_i^e$. The proof is essentially a simplified version of the proof of the previous one, but we present the complete details for the sake of clarity.

**Lemma 5.7.** There is no alternating cycle in $H_M$ that contains the vertex $u_i^e$ for any $e \in E(G)$ and $i \in e$.

**Proof.** Suppose, by way of contradiction, that there exists an alternating cycle $C$ in $H_M$ that contains the vertex $u_i^e$ for some $e \in E(G)$ and $i \in e$. Then, since $\{u_i^e, u_j^e\} \in E$ for the vertex $j \in V(G)$ such that $e = \{i, j\}$, we have that $\{u_i^e, u_j^e\} \in E(C)$. Since $U$ is a vertex cover in $G$, at least one vertex in $\{i, j\}$ belongs to $U$, and let us suppose without loss of generality that this vertex is $i$. Then, $\{a_i, b_i\} \in M$, which means that $\text{label}_M(\{b_i, u_j^e\}) = -2$. Thus, $\{b_i, u_j^e\} \not\in E(H_M)$. Since the only neighbors of $u_j^e$ in $H$ are $b_i$ and $u_j^e$, this implies that $C$ cannot be a cycle, and hence we have reached a contradiction. \hfill $\square$

**Conclusion of the forward direction.** First, by Observation 5.1, there is no alternating path in $H_M$ that starts from a vertex not matched by $M$ and contains one or more edge labeled +2 by $\text{label}_M$. Now, by Lemmas 5.6 and 5.7, if there exists an alternating cycle in $H_M$ that contains one or more edge labeled +2 by $\text{label}_M$, or an alternating path in $H_M$ that contains two or more edges labeled +2 by $\text{label}_M$, then there also exists such a cycle or path that does not contain any vertex in $\{u_i^e : e \in E(G), i \in e\}$. However, if we remove the vertices in $\{u_i^e : e \in E(G), i \in e\}$ from $H$, then the remaining connected components are precisely the Pair Selector and Triple Selector gadgets. By Lemmas 5.3 and 5.4, there exists no alternating cycle in $H_M$ that contains one or more edges labeled +2 by $\text{label}_M$, as well as no alternating path in $H_M$ that contains two or more edges labeled +2 by $\text{label}_M$, which consists only of edges of a Pair Selector gadget. Moreover, by Lemmas 5.3 and 5.5, the same claim holds also with respect to a Triple Selector gadget. Thus, by Proposition 2.1, we conclude that $M$ is popular.

### 5.2 Reverse Direction

Here, we prove that if there exists a popular matching in reduction$(I) = (H, L = \{\ell_v : v \in V(H)\})$, then there exists a solution to the instance $(G, P, T)$ of Partitioned Vertex Cover. For this purpose, let us suppose that $M$ is a popular matching in $(H, L = \{\ell_v : v \in V(H)\})$. In what follows, we first construct a subset $U \subseteq V(G)$. Then, we will show that $U$ is a vertex cover of $G$. Afterwards, we will show that for every $P \in P$, it holds that $|U \cap P| = 1$. Last, we will show that for every $T \in T$, it holds that $|U \cap T| = 2$, which will conclude the proof.

Before implementing this plan, let us give a folklore observation (that is true for any graph and preference lists) that will be used in all proofs ahead.

**Observation 5.2.** Let $J$ be an instance of Popular Matching. Every popular matching in $J$ is a maximal matching.

**Proof.** Suppose, by way of contradiction, that there exists a popular matching $\tilde{M}$ in $J$ that is not maximal. Then, there exists an edge $\{x, y\}$ that is present in the graph in $J$, and with both endpoints not matched by $\tilde{M}$. However, by adding $\{x, y\}$ to $\tilde{M}$, we obtain a matching more popular than $\tilde{M}$, and thus reach a contradiction. \hfill $\square$

**Construction of $U$.** We simply define $U := \{i \in V(G) : \{a_i, b_i\} \in M\}$.

**Proof that $U$ is a vertex cover.** The proof the $U$ is a vertex cover is the same as a proof given by Kavitha [19]. However, for the sake of completeness, and also to verify that although our construction has other components, that same proof still goes through.
Lemma 5.8. The set $U$ is a vertex cover of $G$.

Proof. Toward the proof, we first note that the second condition of Proposition 2.1 implies that both the endpoints of an edge labeled $+2$ by $\text{labeled}_M$ must be matched in $M$.

Thus, for each $i \in V(G)$, the vertex $a_i$ in $V(H)$ must be matched in $M$, since otherwise edge $\{a_i, b_i\}$ will be labeled $+2$ by $\text{labeled}_M$, and $a_i$ being unmatched will contradict the second condition of Proposition 2.1. What remains to be shown is that either $\{a_i, b_i\} \in M$ or $\{a_i, d_i\} \in M$. In fact, we will prove the following claim.

Claim 5.8.1. For every $i \in V(G)$, we have that

- $\{a_i, b_i\} \in M$, or
- both $\{a_i, d_i\} \in M$ and $\{b_i, c_i\} \in M$.

Proof. We begin by ruling out the possibility that $a_i$ is matched to $c_i$ by $M$. If $a_i$ is matched to $c_i$, then $b_i$ must be matched to $u_i^e$ for some edge $e$ incident to $i$, as otherwise $\{a_i, b_i\}$ is an edge labeled $+2$ by $\text{labeled}_M$ and $b_i$ is unmatched. As noted above, this is a contradiction to the second condition of Proposition 2.1. Let $e = \{i, j\}$. Then, the $M$-alternating path $\{c_i, a_i\} - \{a_i b_i\} - \{b_i, u_i^e\} - \{u_i^e, u_i^f\}$ contains two edges labeled $+2$ by $\text{labeled}_M$, namely $\{a_i b_i\}$ and $\{u_i^e, u_i^f\}$, a contradiction to the third condition of Proposition 2.1. Thus, $a_i$ can only be matched to $a_j$ or $d_i$.

It remains to show that if $a_i$ is matched to $d_i$, then $b_i$ is matched to $c_i$. To this end, suppose that $a_i$ is matched to $d_i$. If $b_i$ is unmatched, then by removing $\{a_i, d_i\}$ and adding $\{b_i, a_i\}$, we obtain a more popular matching. Thus, if $b_i$ is not matched to $c_i$, then it must be matched to $u_i^e$ for some edge $e = \{i, j\}$ incident to $i$. In this case, where $b_i$ is matched to $u_i^e$, by removing $\{a_i, d_i\}$, $\{b_i, u_i^e\}$ and $\{u_i^e, b_j\}$ (if $\{u_i^e, b_j\} \in M$), and adding $\{a_i, b_i\}$ and $\{u_i^e, u_i^f\}$, we get four votes in favor of the change (from $a_i, b_i, u_i^e$ and $u_i^f$) and at most two votes against it (from $d_i$ and possibly $b_j$), which contradicts the popularity of $M$.

We now proceed with the proof of the lemma. To this end, let $\{i, j\} \in E(G)$ be an arbitrarily chosen edge. To prove that this edge is covered by $U$, we need to show that at least one edge among $\{a_i, b_i\}$ and $\{a_j, b_j\}$ is in $M$. Suppose, by way of contradiction, that this statement is false. Then, by Claim 5.8.1, it holds that $\{a_i, d_i\}, \{b_i, c_i\}, \{a_j, d_j\}, \{b_j, c_j\} \in M$. In this case, $\{u_i^e, u_i^f\}$ must be in $M$ (else they are not matched, which contradicts Observation 5.2). Then, we remove $\{a_i, d_i\}, \{b_i, c_i\}, \{u_i^e, u_i^f\}, \{a_j, d_j\}$, and $\{b_j, c_j\}$ from $M$, and add $\{a_i, c_i\}, \{b_i, u_i^e\}, \{a_j, c_j\}$, and $\{b_j, u_i^f\}$ to $M$. Then, we gain six votes in favor of the replacement (from $a_i, a_j, b_i, b_j, c_i$ and $c_j$) and only four votes against it (from $d_i, d_j, u_i^e$, and $u_i^f$), which contradicts the popularity of $M$. This completes the proof of the lemma.

Proof that $U$ is a solution. Since we have already established that $U$ is a vertex cover, the proof that $U$ is a solution will follow from the correctness of the two following lemmas.

Lemma 5.9. For every $P \in \mathcal{P}$, it holds that $|U \cap P| = 1$.

Proof. Let us consider some arbitrary pair $P = \{i, j\} \in \mathcal{P}$. By Lemma 5.8, and because a pair is also an edge in $G$, we have that $|U \cap P| \geq 1$. Thus, to prove the lemma, it suffices to show that it is not possible to have $|U \cap P| = 2$. To this end, suppose by way of contradiction that $|U \cap P| = 2$. By the definition of $U$, both $\{a_i, b_i\} \in M$ and $\{a_j, b_j\} \in M$. Note that the only neighbors of $c_i$ besides $a_i$ and $b_i$ are $f_{ji}$ and $d_i$, the only neighbors of $f_{ji}$ are $c_i$ and $d_i$, and the only neighbors of $d_i$ besides $a_j$ are $c_i$ and $f_{ji}$. Thus, by Observation 5.2, $M$ must contain exactly one of the edges $\{c_i, d_i\}$, $\{d_i, f_{ji}\}$, and $\{f_{ji}, c_i\}$. If $\{c_i, d_i\} \in M$, then by replacing this edge by $\{f_{ji}, c_i\}$, we obtain a more popular matching (both $c_i$ and $f_{ji}$ vote in favor of the replacement, while only $d_i$ votes against it). If $\{d_i, f_{ji}\} \in M$, then by replacing this edge by $\{c_i, d_i\}$, we obtain a more popular matching (both
$c_i$ and $d_j$ vote in favor of the replacement, while only $f_{ji}$ votes against it). If $\{f_{ji}, c_i\} \in M$, then by replacing this edge by $\{d_j, f_{ji}\}$, we obtain a more popular matching (both $d_j$ and $f_{ji}$ vote in favor of the replacement, while only $c_i$ votes against it). Since every case led to a contradiction, the proof is complete. \hfill \Box

Lemma 5.10. For every $T \in \mathcal{T}$, it holds that $|U \cap T| = 2$.

**Proof.** Let us consider some arbitrary triple $T = \{i, j, k\} \in \mathcal{T}$. By Lemma 5.8, and because a triple is also a triangle in $G$, we have that $|U \cap T| \geq 2$. Thus, to prove the lemma, it suffices to show that it is not possible to have $|U \cap T| = 3$. To this end, suppose by way of contradiction that $|U \cap T| = 3$. By the definition of $U$, all the three edges $\{a_i, b_j\}$, $\{a_j, b_k\}$, and $\{d_k, b_h\}$ belong to $M$. Note that the only neighbors of $d_i$ besides $a_i$ are $d_j$ and $d_k$, the only neighbors of $d_j$ besides $a_j$ are $d_i$ and $d_k$, and the only neighbors of $d_k$ besides $a_k$ are $d_i$ and $d_j$. Thus, by Observation 5.2, $M$ must contain exactly one of the edges $\{d_i, d_j\}$, $\{d_j, d_k\}$, and $\{d_k, d_i\}$. If $\{d_i, d_j\} \in M$, then by replacing this edge by $\{d_j, d_k\}$, we obtain a more popular matching (both $d_j$ and $d_k$ vote in favor of the replacement, while only $d_i$ votes against it). If $\{d_j, d_k\} \in M$, then by replacing this edge by $\{d_i, d_j\}$, we obtain a more popular matching (both $d_i$ and $d_j$ vote in favor of the replacement, while only $d_k$ votes against it). If $\{d_k, d_i\} \in M$, then by replacing this edge by $\{d_i, d_j\}$, we obtain a more popular matching (both $d_i$ and $d_j$ vote in favor of the replacement, while only $d_k$ votes against it). Since every case led to a contradiction, the proof is complete. \hfill \Box

REFERENCES

[1] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. 2005. Popular matchings. In Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA’05). ACM-SIAM, 424–432. https://dl.acm.org/doi/10.5555/1070432.1070491.

[2] David J. Abraham, Robert W. Irving, Telikepalli Kavitha, and Kurt Mehlhorn. 2007. Popular matchings. SIAM J. Comput. 37, 4 (2007), 1030–1045. DOI: https://doi.org/10.1137/06067328X

[3] Péter Biró, Robert W. Irving, and David F. Manlove. 2010. Popular matchings in the marriage and roommates problems. In Algorithms and Complexity, Tiziana Calamoneri and Josep Diaz (Eds.). Springer, Berlin, 97–108.

[4] K. S. Chung. 2000. On the existence of stable roommate matchings. Games Econ. Behav. 33, 2 (2000), 206–230.

[5] Agnes Cseh. 2015. Popular matchings. (2015). Combinatorial Optimization, Hausdorff Trimester Program. Retrieved from http://www.him.uni-bonn.de/combinatorial-optimization-2015/.

[6] Agnes Cseh. 2017. Popular matchings. Trends Comput. Soc. Choice 105, 3 (2017), 105–122.

[7] Agnes Cseh, Chien-Chung Huang, and Telikepalli Kavitha. 2017. Popular matchings with two-sided preferences and one-sided ties. SIAM J. Discr. Math. 31, 4 (2017), 2348–2377.

[8] Agnes Cseh and Telikepalli Kavitha. 2016. Popular edges and dominant matchings. In Proceedings of the 18th International Conference Integer Programming and Combinatorial Optimization (IPCO’16). Springer, Berlin, 138–151.

[9] M.-J.-A.-N. de C. (Marquis de) Condorcet. 1785. Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix. De l’imprimerie royale, M. DCCCLXXV.

[10] Yuri Faenza, Telikepalli Kavitha, Vladlena Powers, and Xingyu Zhang. 2019. Popular matchings and limits to tractability. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’19). ACM-SIAM, 2790–2809. DOI: https://doi.org/10.1137/1.9781611975482.173

[11] D. Gale and L. Shapley. 1962. College admissions and the stability of marriage. Am. Math. Monthly 69, 1 (1962), 9–15.

[12] P. Gärdenfors. 1975. Match making: Assignments based on bilateral preferences. Behav. Sci. 20, 3 (1975), 166–173.

[13] Sushmita Gupta, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. 2019. Popular matching in roommates setting is NP-hard. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’19). ACM-SIAM, USA, 2810–2822. DOI: https://doi.org/10.1137/1.9781611975482.174

[14] Chien-Chung Huang and Telikepalli Kavitha. 2013. Near-popular matchings in the roommates problem. SIAM J. Discr. Math. 27, 1 (2013), 43–62.

[15] Chien-Chung Huang and Telikepalli Kavitha. 2013. Popular matchings in the stable marriage problem. Inf. Comput. 222 (2013), 180–194.

[16] Chien-Chung Huang, Telikepalli Kavitha, Dimitrios Michail, and Meghana Nasre. 2008. Bounded unpopularity matchings. In Proceedings of the 11th Scandinavian Workshop on Algorithm Theory (SWAT’08). Springer, Berlin, 127–137.
[17] Chien-Chung Huang and Telikepalli Kavitha. 2017. Popularity, mixed matchings, and self-duality. In Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms. ACM-SIAM, 2294–2310. DOI: https://doi.org/10.1137/1.9781611974782.151

[18] Telikepalli Kavitha. 2014. A size-popularity tradeoff in the stable marriage problem. SIAM J. Comput. 43, 1 (2014), 52–71.

[19] Telikepalli Kavitha. 2018. Max-size popular matchings and extensions. arXiv:1802.07440v1.

[20] Telikepalli Kavitha. 2018. The popular roommates problem. arXiv:1804.00141v2.

[21] Telikepalli Kavitha, Julián Mestre, and Meghana Nasre. 2011. Popular mixed matchings. Theor. Comput. Sci. 412, 24 (2011), 2679–2690.

[22] Telikepalli Kavitha and Meghana Nasre. 2009. Optimal popular matchings. Discr. Appl. Math. 157, 14 (2009), 3181–3186.

[23] Telikepalli Kavitha and Meghana Nasre. 2009. Popular matchings with variable job capacities. In Proceedings of the 20th International Symposium Algorithms and Computation (ISAAC’09). Springer, Berlin, 423–433.

[24] Tamás Király and Zsuzsa Mészáros-Karkus. 2020. Finding strongly popular b-matchings in bipartite graphs. Eur. J. Combin. 88 (2020), 103105.

[25] David Manlove. 2013. The House Allocation problem (with applications to reviewer assignment). (2013). Summer School on Matching Problems, Markets, and Mechanisms, Budapest 2013. Retrieved from http://econ.core.hu/english/res/MatchingSchool.html.

[26] David Manlove and Colin T. S. Sng. 2006. Popular matchings in the capacitated house allocation problem. In Proceedings of the 14th Annual European Symposium (ESA’06). Springer, Berlin, 492–503.

[27] David F. Manlove. 2013. Algorithmics of Matching Under Preferences. Theoretical Computer Science, Vol. 2. World Scientific.

[28] Richard Matthew McCutchen. 2008. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In Proceedings of the Latin American Theoretical Informatics Symposium (LATIN’08). Springer, Berlin, 593–604.

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

[30] Michael Sipser. 2006. Introduction to the Theory of Computation. Thomson Course Technology.

[31] Egres Open Wiki. 2019. Deciding the existence of popular matchings. Retrieved from http://lemon.cs.elte.hu/egres/open/Deciding_the_existence_of_popular_matchings.

[32] Y. Xiao, Z. Han, C. Yuen, and L. A. DaSilva. 2016. Carrier aggregation between operators in next generation cellular networks: A stable roommate market. IEEE Trans. Wireless Commun. 15, 1 (2016), 633–650. DOI: https://doi.org/10.1109/TWC.2015.2477077

Received December 2019; revised November 2020; accepted January 2021