# Arborescences, Colorful Forests, and Popularity*

Telikepalli Kavitha † Kazuhisa Makino ‡ Ildikó Schlotter § Yu Yokoi ¶

**Abstract**

Our input is a directed, rooted graph $G = (V \cup \{r\}, E)$ where each vertex in $V$ has a partial order preference over its incoming edges. The preferences of a vertex extend naturally to preferences over arborescences rooted at $r$. We seek a *popular* arborescence in $G$, i.e., one for which there is no "more popular" arborescence. Popular arborescences have applications in liquid democracy or collective decision making; however, they need not exist in every input instance. The popular arborescence problem is to decide if a given input instance admits a popular arborescence or not. We show a polynomial-time algorithm for this problem, whose computational complexity was not known previously.

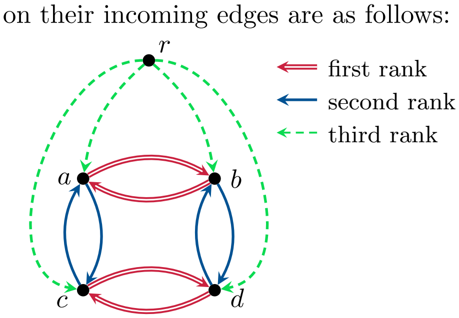
Our algorithm is combinatorial, and can be regarded as a primal-dual algorithm. It searches for an arborescence along with its dual certificate, a chain of subsets of $E$, witnessing its popularity. In fact, our algorithm solves the more general popular common base problem in the intersection of two matroids, where one matroid is the partition matroid defined by any partition $E = \dot\bigcup_{v \in V} \delta(v)$ and the other is an arbitrary matroid $M = (E, \mathcal{I})$ of rank $|V|$, with each $v \in V$ having a partial order over elements in $\delta(v)$. We extend our algorithm to the case with forced or forbidden edges.

We also study the related popular colorful forest (or more generally, the popular common independent set) problem where edges are partitioned into color classes, and the task is to find a colorful forest that is popular within the set of all colorful forests. For the case with weak rankings, we formulate the popular colorful forest polytope, and thus show that a minimum-cost popular colorful forest can be computed efficiently. By contrast, we prove that it is NP-hard to compute a minimum-cost popular arborescence, even when rankings are strict.

## 1 Introduction

Let $G = (V \cup \{r\}, E)$ be a directed graph where the vertex $r$ (called the root) has no incoming edge. Every vertex $v \in V$ has a partial ordering $\succ_v$ (i.e., a preference relation that is irreflexive, antisymmetric and transitive) over its incoming edges, as in this example from [19] where preference orders are strict rankings. Here $V = \{a, b, c, d\}$ and the preference orders of these four vertices on their incoming edges are as follows:

$$(b,a) \succ_a (c,a) \succ_a (r,a)$$
$$(a,b) \succ_b (d,b) \succ_b (r,b)$$
$$(d,c) \succ_c (a,c) \succ_c (r,c)$$
$$(c,d) \succ_d (b,d) \succ_d (r,d).$$




We are interested in computing an *optimal arborescence* rooted at $r$, where an arborescence is an acyclic subgraph of $G$ in which each vertex $v \in V$ has a unique incoming edge. Our notion of optimality is a function of the preferences $(\succ_v)_{v \in V}$ of vertices for their incoming edges.

Given any pair of arborescences $A$ and $A'$ in $G$, we say that $v \in V$ prefers $A$ to $A'$ if $v$ prefers its incoming edge in $A$ to its incoming edge in $A'$, i.e., $v$ prefers $A$ to $A'$ if $A(v) \succ_v A'(v)$ where $A(v)$ (resp., $A'(v)$) is $v$'s incoming edge in $A$ (resp., $A'$). Let $\phi(A, A')$ be the number of vertices that prefer $A$ to $A'$. We say that $A$ is *more popular than* $A'$ if $\phi(A, A') > \phi(A', A)$.

*The full version of the paper can be accessed at https://arxiv.org/abs/2310.19455

†Tata Institute of Fundamental Research, Mumbai, India; kavitha@tifr.res.in

‡Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan; makino@kurims.kyoto-u.ac.jp

§Centre for Economic and Regional Studies, Budapest, Hungary; also at Budapest University of Technology and Economics, Budapest, Hungary; schlotter.ildiko@krtk.hun-ren.hu

¶Tokyo Institute of Technology, Tokyo, Japan; yokoi@c.titech.ac.jp

Definition 1.1. *An arborescence $A$ is popular if $\phi(A, A') \geq \phi(A', A)$ for all arborescences $A'$.*

Our notion of optimality is popularity, in other words, we seek a popular arborescence $A$ in $G$. So there is *no* arborescence more popular than $A$, thus $A$ is maximal under the "more popular than" relation. The "more popular than" relation is not transitive and popular arborescences need not always exist.

Consider the example from [19] illustrated above. The arborescence $A = \{(r, a), (a, b), (a, c), (c, d)\}$ is not popular, since the arborescence $A' = \{(r, d), (d, c), (c, a), (a, b)\}$ is more popular. This is because $a$ and $c$ prefer $A'$ to $A$, while $d$ prefers $A$ to $A'$, and $b$ is indifferent between $A$ and $A'$. We can similarly obtain an arborescence $A'' = \{(r, b), (b, a), (b, d), (d, c)\}$ more popular than $A'$. It is easy to check that for any arborescence here, there is a more popular arborescence. Therefore this instance has no popular arborescence.

Consider the above instance without the edge $(r, d)$. Vertex preferences are the same as in the earlier instance, except that vertex $d$ has no third-choice edge. It can be shown that this instance has two popular arborescences: $A = \{(r, a), (a, b), (a, c), (c, d)\}$ and $A''' = \{(r, b), (b, a), (a, c), (c, d)\}$ (Appendix A has more details).

**The popular arborescence problem.** Given a directed graph $G$ as described above, the popular arborescence problem is to determine if $G$ admits a popular arborescence or not, and to find one, if so. The computational complexity of the popular arborescence problem was posed as an open problem at the Emléktábla workshop [22] in 2019 and the problem has remained open till now. Thus it is an intriguing open problem—aside from its mathematical interest and curiosity, it has applications in *liquid democracy*, which is a voting scheme that allows a voter to delegate its vote to another voter.[1]

**Popular branchings.** A special case of the popular arborescence problem is the popular branching problem. A branching is a directed forest in a digraph $G = (V, E)$ where each vertex has at most one incoming edge. Any branching in $G$ can be viewed as an arborescence in an auxiliary graph obtained by augmenting $G$ with a new vertex $r$ as the root and adding the edge $(r, v)$ for each $v \in V$ as the least-preferred incoming edge of $v$. So the problem of deciding whether the given instance $G$ admits a popular branching or not reduces to the problem of deciding whether this auxiliary instance admits a popular arborescence or not. An efficient algorithm for this special case of the popular arborescence problem (where the root $r$ is an in-neighbor of every $v \in V$) was given in [19].

The applications of popular branchings in liquid democracy were discussed in [19]—as mentioned above, each voter can delegate its vote to another voter; however delegation cycles are forbidden. A popular branching $B$ represents a cycle-free delegation process that is stable, and every root in $B$ casts a weighted vote on behalf of all its descendants. As mentioned in [19], liquid democracy has been used for internal decision making at Google [16] and political parties such as the German *Pirate Party* or the Swedish party *Demoex*. We refer to [28] for more details.

However, in many real-world applications, not all agents would be willing to be representatives, i.e., to be roots in a branching. Thus it cannot be assumed that *every* vertex is an out-neighbor of $r$, so it is only agents who are willing to be representatives that are out-neighbors of $r$ in our instance. Thus the popular arborescence problem has to be solved in a general digraph $G = (V \cup \{r\}, E)$ rather than in one where every vertex is an out-neighbor of $r$. As mentioned earlier, the computational complexity of the popular arborescence problem was open till now. We show the following result.

Theorem 1.1. *Let $G = (V \cup \{r\}, E)$ be a directed graph where each $v \in V$ has a partial order over its incoming edges. There is a polynomial-time algorithm to solve the* popular arborescence *problem in $G$.*

**Popular matchings and assignments.** The notion of popularity has been extensively studied in the domain of bipartite matchings where vertices on one side of the graph have weak rankings (i.e., linear preference order with possible ties) over their neighbors. The popular matching problem is to decide if such a bipartite graph admits a *popular matching*, i.e., a matching $M$ such that there is no matching more popular than $M$.

An efficient algorithm for the popular matching problem was given almost 20 years ago [1]. Very recently (in 2022), the popular assignment problem was considered [18]. What is sought in this problem is a perfect matching that is popular within the set of perfect matchings—so the cardinality of the matching is more important than

[1] A vertex $v$ delegating its vote to $u$ should be represented as the edge $(v, u)$; however as said in [19], it will be more convenient to denote this delegation by $(u, v)$ so as to be consistent with downward edges in an arborescence.

popularity here. It is easy to see that the popular assignment problem is a generalization of the popular matching problem (a simple reduction from the popular matching problem to the popular assignment problem can be shown by adding some dummy vertices). An efficient algorithm for the popular assignment problem was given in [18].

**Popular common base problem.** Observe that the popular arborescence and popular assignment problems are special cases of the popular common base problem in the intersection of two matroids, where one matroid is the partition matroid defined by any partition $E = \dot\bigcup_{v\in V} \delta(v)$ and the other is an arbitrary matroid $M = (E, \mathcal{I})$ of rank $|V|$, and each $v \in V$ has a partial order $\succ_v$ over elements in $\delta(v)$.

- For any pair of common bases (i.e., common maximal independent sets) $I$ and $I'$ in the matroid intersection, we say that $v \in V$ prefers $I$ to $I'$ if $v$ prefers the element in $I \cap \delta(v)$ to the element in $I' \cap \delta(v)$, i.e., $e \succ_v f$ where $I \cap \delta(v) = \{e\}$ and $I' \cap \delta(v) = \{f\}$. Let $\phi(I, I')$ be the number of vertices in $V$ that prefer $I$ to $I'$. The set $I$ is popular within the set of common bases if $\phi(I, I') \geq \phi(I', I)$ for all common bases $I'$.

Arborescences are the common bases in the intersection of a partition matroid with a graphic matroid (for any edge set $I \subseteq E$, $I \in \mathcal{I}$ if and only if $I$ has no cycle in the underlying undirected graph) while assignments are common bases in the intersection of two partition matroids. In fact, our algorithm and the proof of correctness for Theorem 1.1 work for the general popular common base problem.

THEOREM 1.2. *A popular common base in the intersection of a partition matroid on $E = \dot\bigcup_{v\in V} \delta(v)$ with any matroid $M = (E, \mathcal{I})$ of rank $|V|$ can be computed in polynomial time.*

Interestingly, the popular common independent set problem which asks for a common independent set that is popular in the set of all common independent sets (of all sizes) in the matroid intersection can be reduced to the popular common base problem (see Section 4). Therefore, the following fact is obtained as a corollary to Theorem 1.2.

COROLLARY 1.1. *A popular common independent set in the intersection of a partition matroid on $E = \dot\bigcup_{v\in V} \delta(v)$ with any matroid $M = (E, \mathcal{I})$ can be computed in polynomial time.*

All of the following problems fall in the framework of a popular common base (or common independent set) in the intersection of a partition matroid with another matroid:

1. Popular matchings [1].
2. Popular assignments [18].
3. Popular branchings [19].
4. Popular matchings with matroid constraints[2] [17].

Since Corollary 1.1 holds for partial order preferences, it generalizes the tractability result in [17] which assumes that preferences are weak rankings (note that the results in [17] are based on the paper [1], which in turn strongly relies on weak rankings). There are other interesting problems, e.g., the popular colorful forest problem and the popular colorful spanning tree problem, that fall in our framework. The popular colorful forest problem and popular colorful spanning tree problem are new problems introduced in our paper and they are natural generalizations of the popular branching problem and popular arborescence problem, respectively.

**Popular colorful forests and popular colorful spanning trees.** The input here is an undirected graph $G$ where each edge has a color in $\{1, \dots, n\}$. A forest $F$ is *colorful* if each edge in $F$ has a distinct color. Colorful forests are the common independent sets of the partition matroid defined by color classes and the graphic matroid of $G$. For each $i \in \{1, \dots, n\}$, we assume there is an agent $i$ with a partial order $\succ_i$ over color $i$ edges. Agent $i$ prefers forest $F$ to forest $F'$ if either (i) $F$ contains an edge colored $i$ while $F'$ has no edge colored $i$ or (ii) both $F$ and $F'$ contain color $i$ edges and $i$ prefers the color $i$ edge in $F$ to the color $i$ edge in $F'$.

A colorful forest $F$ is popular if $\phi(F, F') \geq \phi(F', F)$ for all colorful forests $F'$, where $\phi(F, F')$ is the number of agents that prefer $F$ to $F'$. The popular colorful forest problem is to decide if a given graph $G$ admits a popular

[2]This problem asks for a popular many-to-one matching in a bipartite graph $G = (A \cup B, E)$ where vertices in $A$ have weak rankings and the vertices that get matched to each $b \in B$ must form an independent set in a matroid $M_b$.

colorful forest or not, and to find one, if so. The motivation here is to find an optimal *independent* network (cycles are forbidden) with diversity, i.e., there is at most one edge from each color class—as before, our definition of optimality is popularity. The popular branching problem is a special case of the popular colorful forest problem where all edges entering vertex $i$ are colored $i$.

A colorful spanning tree is a colorful forest with exactly one component. In the popular colorful spanning tree problem, *connectivity* is more important than popularity, and we seek popularity within the set of colorful spanning trees rather than popularity within the set of all colorful forests.

**Implications of Theorem 1.2.** Along with the popular arborescence problem, our algorithm also solves the problems considered in [1, 17–19]; furthermore, it also solves the popular colorful forest and popular colorful spanning tree problems. The algorithms given in [17–19] for solving their respective problems are quite different from each other. Thus our algorithm provides a unified framework for all these problems and shows that there is one polynomial-time algorithm that solves all of them.

In general, the matroid intersection need not admit common bases, and in such a case, an alternative is a largest common independent set that is popular among all largest common independent sets. This problem can be easily reduced to the popular common base problem (see the full version). Furthermore, along with some simple reductions, we can use our popular common base algorithm to find a popular solution under certain constraints.

For example, we can find a common independent set that is popular subject to a size constraint (if a solution exists). We can further solve the problem under a category-wise size constraint: consider a setting where the set $V$ of voters is partitioned into categories, and for each category, there are lower and upper bounds on the number of voters who (roughly speaking) have an element in the chosen independent set belonging to them (see the full version). In the liquid democracy application mentioned earlier, this translates to setting lower and upper bounds on the number of representatives taken from each category so as to ensure that there is diversity among representatives.

**Popular common independent set polytope.** If preferences are weak rankings, then we also give a formulation of an extension of the *popular common independent set polytope*, i.e., the convex hull of incidence vectors of popular common independent sets in our matroid intersection.

THEOREM 1.3. *If preferences are weak rankings, the popular common independent set polytope is a projection of a face of the matroid intersection polytope.*

There are an exponential number of constraints in this formulation, however it admits an efficient separation oracle. As a consequence, when there is a function $\mathsf{cost} : E \to \mathbb{R}$, a min-cost popular common independent set can be computed in polynomial time by optimizing over this polytope, assuming that preferences are weak rankings. Unfortunately, such a result does not hold for the min-cost popular arborescence problem.

THEOREM 1.4. *Given an instance $G = (V \cup \{r\}, E)$ of the popular arborescence problem where each vertex has a strict ranking over its incoming edges along with a function $\mathsf{cost} : E \to \{0, 1, \infty\}$, it is NP-hard to compute a min-cost popular arborescence in $G$.*

Nevertheless, finding a popular arborescence with forced/forbidden edges in an input instance with partial order preferences is polynomial-time solvable. This result allows us to recognize in polynomial time all those edges that are present in every popular arborescence and all those edges that are present in *no* popular arborescence.

THEOREM 1.5. *For any instance $G = (V \cup \{r\}, E)$ of the popular arborescence problem with a set $E^+ \subseteq E$ of forced edges and a set $E^- \subseteq E$ of forbidden edges, there is a polynomial-time algorithm to decide if there is a popular arborescence $A$ with $E^+ \subseteq A$ and $E^- \cap A = \emptyset$ and to find one, if so.*

In instances where a popular arborescence does not exist, we could relax popularity to *near-popularity* or "low unpopularity". A standard measure of unpopularity is the *unpopularity margin* [24], defined for any arborescence $A$ as $\mu(A) = \max_{A'} \phi(A', A) - \phi(A, A')$ where the maximum is taken over all arborescences $A'$. An arborescence $A$ is popular if and only if $\mu(A) = 0$. Unfortunately, finding an arborescence with minimum unpopularity margin is NP-hard.

THEOREM 1.6. *Given an instance $G = (V \cup \{r\}, E)$ of the popular arborescence problem where each vertex has a strict ranking over its incoming edges, together with an integer $k$, it is NP-complete to decide whether $G$ contains an arborescence with unpopularity margin at most $k$.*

**1.1 Background.** The notion of popularity was introduced by Gärdenfors [14] in 1975 in bipartite graphs with two-sided strict preferences. In this model every stable matching [13] is popular, thus popular matchings always exist in this setting. When preferences are *one-sided*, popular matchings need not always exist. This is not very surprising given that popular solutions correspond to (weak) Condorcet winners [5, 25] and it is well-known in social choice theory that such a winner need not exist.

For the case when preferences are weak rankings, a combinatorial characterization of popular matchings was given in [1] and this yielded an efficient algorithm to solve the popular matching problem in this case. Note that the characterization in [1] does not generalize to partial order preferences, as argued in [20]. Several extensions of the popular matching problem have been considered such as random popular matchings [23], weighted voters [26], capacitated objects [30], popular mixed matchings [21], and popularity with matroid constraints [17]. We refer to [6] for a survey on results in popular matchings.

Popular spanning trees were studied in [7–9] where the incentive was to find a "socially best" spanning tree. However, in contrast to the popular colorful spanning tree problem, edges have no colors in their model and voters have rankings over the entire edge set. Many different ways to compare a pair of trees were studied here, and most of these led to hardness results. Popular branchings, i.e., popular directed forests, in a directed graph (where each vertex has preferences as a partial order over its incoming edges) were studied in [19] where a polynomial-time algorithm was given for the popular branching problem. When preferences are weak rankings, polynomial-time algorithms for the min-cost popular branching problem and the $k$-unpopularity margin branching problem were shown in [19]; however these problems were shown to be NP-hard for partial order preferences. The popular branching problem where each vertex (i.e., voter) has a weight was considered in [27].

The popular assignment algorithm from [18] solves the popular maximum matching problem in a bipartite graph, and works for partial order preferences. It was also shown in [18] that the min-cost popular assignment problem is NP-hard, even for strict rankings.

Many combinatorial optimization problems can be expressed as (largest) common independent sets in the intersection of two matroids. Interestingly, constraining one of the two matroids in the matroid intersection to be a partition matroid is not really a restriction, because any matroid intersection can be reduced to the case where one matroid is a partition matroid (see [11, Claims 104–106]). We refer to [15, 29] for notes on matroid intersection and for the formulation of the matroid intersection polytope.

**1.2 An overview of our algorithm.** For an arborescence $A$, we can naturally define a weight function $\mathsf{wt}_A : E \to \{-1, 0, 1\}$ such that for any arborescence $A'$ we have $\mathsf{wt}_A(A') = \phi(A', A) - \phi(A, A')$. Then a popular arborescence $A$ is a max-weight arborescence in $G = (V \cup \{r\}, E)$ with this function $\mathsf{wt}_A$. Therefore, the popular arborescence problem is the problem of finding $A \in \mathcal{A}_G$ such that $\max_{A' \in \mathcal{A}_G} \mathsf{wt}_A(A') = \mathsf{wt}_A(A) = 0$ where $\mathcal{A}_G$ is the set of all arborescences in $G$. Thus a popular arborescence $A$ is an optimal solution to the max-weight arborescence LP with edge weights given by $\mathsf{wt}_A$.

**Dual certificates.** We show that every popular arborescence $A$ has a dual certificate with a special structure; this corresponds to a *chain* $\mathcal{C} = \{C_1, \ldots, C_p\}$ of subsets of $E$ with $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$ and $\mathrm{span}(A \cap C_i) = C_i$ for all $i$.[3] Our algorithm to compute a popular arborescence is a search for such a chain $\mathcal{C}$ and arborescence $A$. At a high level, this method is similar to the approach used in [18] for popular assignment, however our dual certificates are more complex than those in [18], and hence the steps in our algorithm (and its proof of correctness) become much more challenging.

Given a chain $\mathcal{C}$ of subsets of $E$, there is a polynomial-time algorithm to check if $\mathcal{C}$ corresponds to a dual certificate for some popular arborescence. It follows from dual feasibility and complementary slackness that $\mathcal{C}$ is a dual certificate if and only if a certain subgraph $G_{\mathcal{C}} = (V \cup \{r\}, E(\mathcal{C}))$ admits an arborescence $A$ such that $\mathrm{span}(A \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$. If such an arborescence $A$ exists in $G_{\mathcal{C}}$, then it is easy to show that $A$ is a popular arborescence in $G$ with $\mathcal{C}$ as its dual certificate.

If $G_{\mathcal{C}}$ does not admit such an arborescence, then we need to update $\mathcal{C}$. Since updating $\mathcal{C}$ changes $E(\mathcal{C})$, we now seek an arborescence $A$ in the new graph $G_{\mathcal{C}}$ such that $\mathrm{span}(A \cap C_i) = C_i$ for all $i$. If such an $A$ does not exist, then $\mathcal{C}$ is updated again. Note that updating $\mathcal{C}$ may increase $|\mathcal{C}|$. When $|\mathcal{C}|$ becomes larger than $|V|$, we claim that $G$ has *no* popular arborescence. Among other ideas, our technical novelty lies in the proof of this claim that is based on the strong exchange property of matroids.

[3] In the arborescence case, the set $\mathrm{span}(A \cap C_i)$ is defined as $(A \cap C_i) \cup \{\, e \in E : (A \cap C_i) + e \text{ contains a cycle} \,\}$.

**Matroid Intersection.** Our algorithm holds in the generality of matroid intersection (where one of the matroids is a partition matroid); dual certificates for popular common bases are exactly the same, i.e., chains that are described above. We also show that a popular common independent set has a dual certificate $\mathcal{C} = \{C, E\}$ of length at most 2. This leads to the polyhedral result given in Theorem 1.3.

Our algorithm is quite different from the popular branching algorithm [19] that (loosely speaking) first finds a maximum branching on *best* edges and then augments this branching with *second best* edges entering certain vertices. Indeed, as seen in Theorem 1.3, popular branchings or popular common independent sets have a significantly simpler structure than popular common bases—the latter seem far tougher to characterize and analyze. Pleasingly, as we show here, there is a clean and compact algorithm to solve the popular common base problem (see Algorithm 1).

For the sake of readability, we will describe our results for the popular common base problem in terms of the popular arborescence problem and our results for the popular common independent set problem in terms of the popular colorful forest problem.

**Organization of the paper.** The rest of the paper is organized as follows. Section 2 describes dual certificates for popular arborescences. Section 3 presents the popular arborescence algorithm and its proof of correctness. In Section 4, we discuss popular colorful forests and their polytope. Section 6 provides the algorithm for the popular arborescence problem with forced/forbidden edges.

Section 5 shows the NP-hardness of the min-cost popular arborescence problem and we refer to the full version of our paper for the proof of Theorem 1.6. In Appendix A, we present various examples and explain how our algorithm works on them.

## 2 Dual Certificates

In this section we show that every popular arborescence has a special dual certificate—this will be crucial in designing our algorithm in Section 3. Our input is a directed graph $G = (V \cup \{r\}, E)$ where the root vertex $r$ has no incoming edge, and every vertex $v \in V$ has a partial order $\succ_v$ over its set of incoming edges, denoted by $\delta(v)$. For edges $e, f \in \delta(v)$, we write $e \sim_v f$ to denote that $v$ is indifferent between $e$ and $f$, i.e., $e \not\succ_v f$ and $f \not\succ_v e$.

Given an arborescence $A$, there is a simple method (as shown in [19]) to check if $A$ is popular or not. We need to check that $\phi(A, A') \geq \phi(A', A)$ for all arborescences $A'$ in $G$. For this, we will use the following function $\mathsf{wt}_A : E \to \{-1, 0, 1\}$.

For any $v \in V$, let $A(v)$ be the unique edge in $A \cap \delta(v)$. For any $v \in V$ and $e \in \delta(v)$, let

$$\mathsf{wt}_A(e) = \begin{cases} 1 & \text{if } e \succ_v A(v) \quad (v \text{ prefers } e \text{ to } A(v)); \\ 0 & \text{if } e \sim_v A(v) \quad (v \text{ is indifferent between } e \text{ and } A(v)); \\ -1 & \text{if } e \prec_v A(v) \quad (v \text{ prefers } A(v) \text{ to } e). \end{cases}$$

It immediately follows from the definition of $\mathsf{wt}_A$ that we have $\mathsf{wt}_A(A') = \phi(A', A) - \phi(A, A')$ for any arborescence $A'$ in $G$. Thus $A$ is popular if and only if every arborescence in $G$ has weight at most 0, where edge weights are given by $\mathsf{wt}_A$.

Consider the linear program problem LP1 below. The constraints of LP1 describe the face of the matroid intersection polytope corresponding to common bases. Recall that this is the intersection of the partition matroid on $E = \biguplus_{v \in V} \delta(v)$ with the graphic matroid $M = (E, \mathcal{I})$ of $G$, whose rank is $|V|$. Here, rank : $2^E \to \mathbb{Z}_+$ is the rank function of $(E, \mathcal{I})$, i.e, for any $S \subseteq E$, the value of rank$(S)$ is the maximum size of an acyclic subset of $S$ in the graph $G$.

$$\text{(LP1)} \quad \max \sum_{e \in E} \mathsf{wt}_A(e) \cdot x_e$$
$$\text{s.t.} \quad \sum_{e \in \delta(v)} x_e = 1 \quad \forall\, v \in V$$
$$\sum_{e \in S} x_e \leq \text{rank}(S) \quad \forall\, S \subseteq E$$
$$x_e \geq 0 \quad \forall\, e \in E.$$

$$\text{(LP2)} \quad \min \sum_{S \subseteq E} \text{rank}(S) \cdot y_S + \sum_{v \in V} \alpha_v$$
$$\text{s.t.} \quad \sum_{S : e \in S} y_S + \alpha_v \geq \mathsf{wt}_A(e) \quad \forall\, e \in \delta(v), \forall v \in V$$
$$y_S \geq 0 \quad \forall\, S \subseteq E.$$

The feasible region of LP1 is the arborescence polytope of $G$. Hence LP1 is the max-weight arborescence LP in $G$ with edge weights given by $\mathsf{wt}_A$. The linear program LP2 is the dual LP in variables $y_S$ and $\alpha_v$ where $S \subseteq E$ and $v \in V$.

The arborescence $A$ is popular if and only if the optimal value of LP1 is at most 0, more precisely, if the optimal value is exactly 0, since $\mathsf{wt}_A(A) = 0$. Equivalently, $A$ is popular if and only if the optimal value of LP2 is 0. We will now show that LP2 has an optimal solution with some special properties. For a popular arborescence $A$, a dual optimal solution that satisfies all these special properties (see Lemma 2.1) will be called a *dual certificate* for $A$.

The function $\mathrm{span} : 2^E \to 2^E$ of a matroid $(E, \mathcal{I})$ is defined as follows:

$$\mathrm{span}(S) = \{\, e \in E : \mathrm{rank}(S + e) = \mathrm{rank}(S) \,\} \quad \text{where } S \subseteq E.$$

In particular, if $S \in \mathcal{I}$, then $\mathrm{span}(S) = S \cup \{\, e \in E : S + e \notin \mathcal{I} \,\}$.

A *chain* $\mathcal{C}$ of length $p$ is a collection of $p$ distinct subsets of $E$ such that for each two distinct sets $C, C' \in \mathcal{C}$, we have either $C \subsetneq C'$ or $C' \subsetneq C$. That is, a chain has the form $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$ where $C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_p$.

Lemma 2.1 shows that LP2 always admits an optimal solution in the following special form. The proof is based on basic facts on matroid intersection and linear programming, and we postpone it to the end of Section 3.

LEMMA 2.1. *An arborescence $A$ is popular if and only if there exists a feasible solution $(\vec{y}, \vec{\alpha})$ to LP2 such that $\sum_{S \subseteq E} \mathrm{rank}(S) \cdot y_S + \sum_{v \in V} \alpha_v = 0$ and properties 1–4 are satisfied:*

1. *$\vec{y}$ is integral and its support $\mathcal{C} \coloneqq \{\, S \subseteq E : y_S > 0 \,\}$ is a chain.*
2. *Each $C \in \mathcal{C}$ satisfies $\mathrm{span}(A \cap C) = C$.*
3. *Every element in $\mathcal{C}$ is nonempty, and the maximal element in $\mathcal{C}$ is $E$.*
4. *For each $C \in \mathcal{C}$, we have $y_C = 1$. For each $v \in V$, we have $\alpha_v = -|\,\{\, C \in \mathcal{C} : A(v) \in C \,\}\,|$.*

For any chain $\mathcal{C}$, we will now define a subset $E(\mathcal{C})$ of $E$ that will be used in our algorithm. The construction of $E(\mathcal{C})$ is inspired by the construction of an analogous edge subset in the popular assignment algorithm [18].

For a chain $\mathcal{C} = \{C_1, C_2, \cdots, C_p\}$ with $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, define

$$\begin{aligned} \mathsf{lev}_{\mathcal{C}}(e) &= \text{the index } i \text{ such that } e \in C_i \setminus C_{i-1} && \text{for any } e \in E,\\ \mathsf{lev}^*_{\mathcal{C}}(v) &= \max \{\, \mathsf{lev}_{\mathcal{C}}(e) : e \in \delta(v) \,\} && \text{for any } v \in V, \end{aligned}$$

where we let $C_0 = \emptyset$. Thus every element $e \in E$ has a *level* in $\{1, \ldots, p\}$ associated with it, which is the minimum subscript $i$ such that $e \in C_i$ (where $C_i \in \mathcal{C}$). Furthermore, each $v \in V$ has a $\mathsf{lev}^*_{\mathcal{C}}$-value which is the highest level of any element in $\delta(v)$.

Define $E(\mathcal{C}) \subseteq E$ as follows. For each $v \in V$, an element $e \in \delta(v)$ belongs to $E(\mathcal{C})$ if one of the following two conditions holds:

- $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v)$ and there is no element $e' \in \delta(v)$ such that $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}^*_{\mathcal{C}}(v)$ and $e' \succ_v e$;
- $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v) - 1$ and there is no element $e' \in \delta(v)$ such that $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}^*_{\mathcal{C}}(v) - 1$ and $e' \succ_v e$, and moreover, $e \succ_v f$ for every $f \in \delta(v)$ with $\mathsf{lev}_{\mathcal{C}}(f) = \mathsf{lev}^*_{\mathcal{C}}(v)$.

In other words, $e \in \delta(v)$ belongs to $E(\mathcal{C})$ if either (i) $e$ is a maximal element in $\delta(v)$ with respect to $\succ_v$ among those in $\mathsf{lev}^*_{\mathcal{C}}(v)$ or (ii) $e$ is a maximal element in $\delta(v)$ among those in $\mathsf{lev}^*_{\mathcal{C}}(v) - 1$ and $v$ strictly prefers $e$ to all elements in level $\mathsf{lev}^*_{\mathcal{C}}(v)$. From Lemma 2.1, we obtain the following useful characterization of popular arborescences.

LEMMA 2.2. *An arborescence $A$ is popular if and only if there exists a chain $\mathcal{C} = \{C_1, \ldots, C_p\}$ such that $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, $A \subseteq E(\mathcal{C})$, and $\mathrm{span}(A \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$.*

The proof is given below. Recall that for a popular arborescence $A$, we defined its *dual certificate* as a dual optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 that satisfies properties 1–4 in Lemma 2.1. As shown in the proof of Lemma 2.2, we can obtain such a solution $(\vec{y}, \vec{\alpha})$ from a chain satisfying the properties in Lemma 2.2. We therefore will also use the term *dual certificate* to refer to a chain as described in Lemma 2.2.

***Proof of Lemma 2.2.*** We first show the existence of a desired chain $\mathcal{C}$ for a popular arborescence $A$. Since $A$ is popular, we know from Lemma 2.1 that there exists an optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 such that properties 1–4 hold, where $\mathcal{C}$ is the support of $\vec{y}$. Since the properties $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$ and $\mathrm{span}(A \cap C_i) = C_i$ $(\forall C_i \in \mathcal{C})$ directly follow from properties 3 and 2, respectively, it remains to show that $A \subseteq E(\mathcal{C})$.

Since $(\vec{y}, \vec{\alpha})$ is a feasible solution of LP2, we have $\sum_{S:e\in S} y_S + \alpha_v \geq \mathsf{wt}_A(e)$ for every $e \in \delta(v)$ with $v \in V$. By property 4, the left hand side can be expressed as

$$|\{ C_i \in \mathcal{C} : e \in C_i \}| - |\{ C_i \in \mathcal{C} : A(v) \in C_i \}| = (p - \mathsf{lev}_{\mathcal{C}}(e) + 1) - (p - \mathsf{lev}_{\mathcal{C}}(A(v)) + 1) = \mathsf{lev}_{\mathcal{C}}(A(v)) - \mathsf{lev}_{\mathcal{C}}(e).$$

Thus it is equivalent to the condition that for every $e \in \delta(v)$:

$$\mathsf{lev}_{\mathcal{C}}(A(v)) - \mathsf{lev}_{\mathcal{C}}(e) \geq \mathsf{wt}_A(e) = \begin{cases} 1 & \text{if } e \succ_v A(v); \\ 0 & \text{if } e \sim_v A(v); \\ -1 & \text{if } e \prec_v A(v). \end{cases} \tag{2.1}$$

In particular, this holds for an edge $e'$ with $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}^*_{\mathcal{C}}(v)$, and hence we have $\mathsf{lev}_{\mathcal{C}}(A(v)) \geq \mathsf{lev}^*_{\mathcal{C}}(v) - 1$. Since $\mathsf{lev}_{\mathcal{C}}(A(v)) \leq \mathsf{lev}^*_{\mathcal{C}}(v)$ by $A(v) \in \delta(v)$, $\mathsf{lev}_{\mathcal{C}}(A(v))$ is either $\mathsf{lev}^*_{\mathcal{C}}(v)$ or $\mathsf{lev}^*_{\mathcal{C}}(v) - 1$.

- If $\mathsf{lev}_{\mathcal{C}}(A(v)) = \mathsf{lev}^*_{\mathcal{C}}(v)$, then for any $e \in \delta(v)$ with $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v)$, the left hand side of (2.1) is 0, and hence it must be the case that either $A(v) \succ_v e$ or $A(v) \sim_v e$. Hence $A(v)$ is a maximal element in $\{ e \in \delta(v) : \mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v) \}$ with respect to $\succ_v$.
- If $\mathsf{lev}_{\mathcal{C}}(A(v)) = \mathsf{lev}^*_{\mathcal{C}}(v) - 1$, then we can similarly show that $A(v)$ is a maximal element in the set $\{ e \in \delta(v) : \mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v) - 1 \}$ with respect to $\succ_v$. Furthermore, in this case, for any $e \in \delta(v)$ with $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v)$, the left hand side of (2.1) is $-1$, and hence $A(v) \succ_v e$ must hold.

Therefore, in either case, we have $A(v) \in E(\mathcal{C})$, which implies that $A \subseteq E(\mathcal{C})$.

For the converse, suppose that $\mathcal{C} = \{C_1, \ldots, C_p\}$ is a chain such that $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, $A \subseteq E(\mathcal{C})$, and $\mathrm{span}(A \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$. Define $\vec{y}$ by $y_{C_i} = 1$ for every $C_i \in \mathcal{C}$ and $y_S = 0$ for all $S \in 2^S \setminus \mathcal{C}$. We also define $\vec{\alpha}$ by $\alpha_v = -|\{ C \in \mathcal{C} : A(v) \in C \}|$ for any $v \in V$. Then $(\vec{y}, \vec{\alpha})$ satisfies properties 1-4 given in Lemma 2.1, which also implies that the objective value is 0. Thus it is enough to show that $(\vec{y}, \vec{\alpha})$ is a feasible solution to LP2, because it implies that $A$ is a popular arborescence by Lemma 2.1. Observe that constraint (2.1) is satisfied for every $v \in V$ and $e \in \delta(v)$, which follows from $A \subseteq E(\mathcal{C})$. Since it is equivalent to the constraint in LP2 for $v \in V$ and $e \in \delta(v)$, the proof is completed. □

## 3 Our Algorithm

We now present our main result. The popular arborescence algorithm seeks to construct an arborescence $A$ along with its dual certificate $\mathcal{C} = \{C_1, \ldots, C_p\}$, which is a chain satisfying (i) $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, (ii) $A \subseteq E(\mathcal{C})$, and (iii) $\mathrm{span}(A \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$.

- The existence of such a chain $\mathcal{C}$ means that $A$ is popular by Lemma 2.2.
- Since a popular arborescence need not always exist, the algorithm also needs to detect when a solution does not exist.

The algorithm starts with the chain $\mathcal{C} = \{E\}$ and repeatedly updates it. It always maintains $\mathcal{C}$ as a multichain, where a collection $\mathcal{C} = \{C_1, \cdots, C_p\}$ of indexed subsets of $E$ is called a *multichain* if $C_1 \subseteq \cdots \subseteq C_p$. Note that it is a chain if all the inclusions are strict. We will use the notations $\mathsf{lev}_{\mathcal{C}}$, $\mathsf{lev}^*_{\mathcal{C}}$, and $E(\mathcal{C})$ also for multichains, which are defined in the same manner as for chains.

During the algorithm, $\mathcal{C} = \{C_1, \ldots, C_p\}$ is always a multichain with $C_p = E$ and $\mathrm{span}(C_i) = C_i$ for all $C_i \in \mathcal{C}$. Note that when $\mathrm{span}(C_i) = C_i$ holds, the condition (iii) for some $A$ above is equivalent to $|A \cap C_i| = \mathrm{rank}(C_i)$. Furthermore, as explained later, any multichain can be modified to a chain that satisfies (i) preserving the remaining conditions (ii) and (iii). Therefore, we can obtain a desired chain if $|A \cap C_i| = \mathrm{rank}(C_i)$ is attained for all $C_i \in \mathcal{C}$ for some arborescence $A \subseteq E(\mathcal{C})$ in the algorithm.

**Lex-maximal branching.** In order to determine the existence of an arborescence $A \subseteq E(\mathcal{C})$ that satisfies $|A \cap C_i| = \text{rank}(C_i)$ for all $C_i \in \mathcal{C}$, the algorithm computes a *lex-maximal* branching $I$ in $E(\mathcal{C})$. That is, $I$ is a branching whose $p$-tuple $(|I \cap C_1|, \ldots, |I \cap C_p|)$ is lexicographically maximum among all branchings in $E(\mathcal{C})$. If $(|I \cap C_1|, \ldots, |I \cap C_p|) = (\text{rank}(C_1), \ldots, \text{rank}(C_p))$, then we can show that $I$ is a popular arborescence[4]; otherwise the multichain $\mathcal{C}$ is updated. We describe the algorithm as Algorithm 1; recall that $\text{rank}(E) = |V| = n$.

**Algorithm 1** The popular arborescence algorithm

1: Initialize $p = 1$ and $C_1 = E$. ▷ Initially we set $\mathcal{C} = \{E\}$.
2: **while** $p \leq n$ **do**
3: Compute the edge set $E(\mathcal{C})$ from the current multichain $\mathcal{C}$.
4: Find a branching $I \subseteq E(\mathcal{C})$ that lexicographically maximizes $(|I \cap C_1|, \ldots, |I \cap C_p|)$.
5: **if** $|I \cap C_i| = \text{rank}(C_i)$ for every $i = 1, \ldots, p$ **then** return $I$.
6: Let $k$ be the minimum index such that $|I \cap C_k| < \text{rank}(C_k)$.
7: Update $C_k \leftarrow \text{span}(I \cap C_k)$.
8: **if** $k = p$ **then** $p \leftarrow p + 1$, $C_p \leftarrow E$, and $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_p\}$.
9: Return "$G$ has no popular arborescence".

We include some examples in Appendix A to illustrate the working of Algorithm 1 on different input instances. The following observation is important.

OBSERVATION 3.1. *During Algorithm 1, $\mathcal{C}$ is always a multichain and* $\text{span}(C_i) = C_i$ *for all* $C_i \in \mathcal{C}$.

*Proof.* When $C_k$ is updated, it becomes smaller but the inclusion $C_{k-1} \subseteq C_k$ is preserved. Indeed, since $|I \cap C_{k-1}| = \text{rank}(C_{k-1})$ by the choice of $k$, we have $C_{k-1} \subseteq \text{span}(I \cap C_{k-1}) \subseteq \text{span}(I \cap C_k)$, for the set $C_k$ before the update. Hence the updated value for $C_k$, i.e., $\text{span}(I \cap C_k)$, is still a superset of $C_{k-1}$, and thus $\mathcal{C}$ remains a multichain.

Since any $C_i \in \mathcal{C}$ is defined in the form $\text{span}(X)$ for some $X \subseteq E$ (note that $E = \text{span}(E)$) and $\text{span}(\text{span}(X)) = \text{span}(X)$ holds in general, we have $\text{span}(C_i) = C_i$. ∎

Line 4 can be implemented in polynomial time by a max-weight branching algorithm [2, 4, 10] and, in the more general case of the intersection of two matroids, by the weighted matroid intersection algorithm [12]. Hence Algorithm 1 can be implemented in polynomial time.

**Correctness of the algorithm.** Suppose that a branching $I$ is returned by the algorithm. Then $I$ is an arborescence (see Footnote 4) with $I \subseteq E(\mathcal{C})$, where $\mathcal{C}$ is the current multichain. Since $I$ was returned by the algorithm, we have $|I \cap C_i| = \text{rank}(C_i)$ for all $C_i \in \mathcal{C}$ and this implies $\text{span}(I \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$ by Observation 3.1.

In order to prove that $I$ is a popular arborescence, let us first prune the multichain $\mathcal{C}$ to a chain $\mathcal{C}'$, i.e., $\mathcal{C}'$ contains a single occurrence of each $C_i \in \mathcal{C}$; we will also remove any occurrence of $\emptyset$ from $\mathcal{C}'$. Observe that $E(\mathcal{C}) \subseteq E(\mathcal{C}')$: indeed, if $C_i = C_{i+1}$ in $\mathcal{C}$, then no element $e \in E$ can have $\mathsf{lev}_{\mathcal{C}}(e) = i + 1$, and hence no element gets deleted from $E(\mathcal{C})$ by pruning $C_{i+1}$ from $\mathcal{C}$. Thus $I \subseteq E(\mathcal{C}) \subseteq E(\mathcal{C}')$. This implies that $\mathcal{C}' = \{C'_1, \ldots, C'_{p'}\}$ satisfies $\emptyset \subsetneq C'_1 \subsetneq \cdots \subsetneq C'_{p'} = E$, $I \subseteq E(\mathcal{C}')$, and $\text{span}(I \cap C'_i) = C'_i$ for all $C'_i \in \mathcal{C}'$.[5] Hence $I$ is a popular arborescence by Lemma 2.2.

We will now show that the algorithm always returns a popular arborescence, if $G$ admits one. Let $A$ be any popular arborescence in $G$ and let $\mathcal{D} = \{D_1, \ldots, D_q\}$ be a dual certificate for $A$.

CLAIM 3.1. *We have $q \leq n$ where $|\mathcal{D}| = q$.*

*Proof.* From the definition of dual certificate, we have $\emptyset \subsetneq D_1 \subsetneq \cdots \subsetneq D_q = E$ and $\text{span}(D_i) = D_i$ for each $D_i$. This implies $0 < \text{rank}(D_1) < \cdots < \text{rank}(D_q)$. Since $\text{rank}(D_q) = \text{rank}(E) = |V|$, we obtain $q \leq |V| = n$. ∎

[4] Observe that the branching $I$ will be an *arborescence* since $|I \cap E| = |I \cap C_p| = \text{rank}(C_p) = \text{rank}(E) = |V|$.

[5] In fact, it will turn out that $\mathcal{C} = \mathcal{C}'$, i.e., the final $\mathcal{C}$ obtained by the algorithm itself is a dual certificate of $I$ if the algorithm returns an arborescence $I$. This fact follows from Lemma 3.1 (with $\mathcal{C}'$ substituted for $\mathcal{D}$).

The following crucial lemma shows an invariant of the algorithm that holds for the multichain $\mathcal{C} = \{C_1, \ldots, C_p\}$ constructed in the algorithm and a dual certificate $\mathcal{D} = \{D_1, \ldots, D_q\}$ of any popular arborescence $A$. The proof will be given in this section.

LEMMA 3.1. At any moment of Algorithm 1, $p \le q$ and $D_i \subseteq C_i$ holds for $i = 1, \ldots, p$.

If $p = n+1$ occurs in Algorithm 1, then Lemma 3.1 implies $q \ge n+1$. This contradicts Claim 3.1. Hence it has to be the case that $G$ has no popular arborescence when $p = n+1$. Thus assuming Lemma 3.1, the correctness of Algorithm 1 follows.

Before we prove Lemma 3.1, we need the following claim on $E(\mathcal{C})$ and $E(\mathcal{D})$.

CLAIM 3.2. *Assume $p \le q$ and $D_i \subseteq C_i$ for $i = 1, \ldots, p$. For each $e \in E$, if $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}_{\mathcal{D}}(e)$ and $e \in E(\mathcal{D})$, then $e \in E(\mathcal{C})$.*

*Proof.* Suppose for the sake of contradiction that $e$ fulfills the conditions of the claim, but $e \notin E(\mathcal{C})$. Let $e \in \delta(v)$. It follows from the definition of $E(\mathcal{C})$ that there exists an element $e' \in \delta(v)$ such that one of the following three conditions holds: (a) $\mathsf{lev}_{\mathcal{C}}(e') \ge \mathsf{lev}_{\mathcal{C}}(e) + 2$, (b) $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}_{\mathcal{C}}(e) + 1$ and $e \not\succ_v e'$, or (c) $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}_{\mathcal{C}}(e)$ and $e' \succ_v e$.

Because $D_i \subseteq C_i$ for each $i \in \{1, \ldots, p\}$, we have $\mathsf{lev}_{\mathcal{D}}(e') \ge \mathsf{lev}_{\mathcal{C}}(e')$. Since $\mathsf{lev}_{\mathcal{D}}(e) = \mathsf{lev}_{\mathcal{C}}(e)$, the existence of such an $e' \in \delta(v)$ implies $e \notin E(\mathcal{D})$, a contradiction. Thus we have $e \in E(\mathcal{C})$. ∎

The proof of Lemma 3.1 will use the following fact, known as the strong exchange property, that is satisfied by any matroid.[6]

FACT 3.1. (BRUALDI [3]) *For any $X, Y \in \mathcal{I}$ and $e \in X \setminus Y$, if $Y + e \notin \mathcal{I}$, then there exists an element $f \in Y \setminus X$ such that $X - e + f$ and $Y + e - f$ are in $\mathcal{I}$.*

Now we provide the proof of Lemma 3.1. As mentioned above, this completes the proof of the correctness of our algorithm, and hence we can conclude Theorem 1.1. Furthermore, we can conclude Theorem 1.2 since Algorithm 1 and its correctness proof hold in the generality of a common base in the intersection of the partition matroid on the set $E = \biguplus_{v \in V} \delta(v)$ with any matroid $M = (E, \mathcal{I})$ of rank $|V|$.

***Proof of Lemma 3.1.*** Algorithm 1 starts with $\mathcal{C} = \{E\}$. Then the conditions in Lemma 3.1 hold at the beginning. We show by induction that they are preserved through the algorithm.

It is easy to see that the condition $p \le q$ is preserved. Indeed, whenever Algorithm 1 is going to increase $p$ (in line 8), it is the case that $p + 1 \le q$ because $D_p \subseteq C_p \subsetneq E = D_q$ by the induction hypothesis. Thus $p \le q$ is maintained in the algorithm.

We now show that $D_i \subseteq C_i$ $(i = 1, \ldots, p)$ is maintained. Note that $\mathcal{C}$ is updated in lines 7 or 8. The update in line 8 (adding $C_p = E$) clearly preserves the condition. We complete the proof by showing that the update in line 7 also preserves the condition, i.e., we show the following statement.

- Let $\mathcal{C} = \{C_1, \ldots, C_p\}$ be a multichain with $C_p = E$ such that $p \le q$ and $D_i \subseteq C_i$ for $i = 1, \ldots, p$. Suppose the following two conditions hold.

  1. $I$ is a lex-maximal common independent set subject to $I \subseteq E(\mathcal{C})$.
  2. $\mathrm{span}(I \cap C_i) = C_i$ for $i = 1, \ldots, k-1$, and $\mathrm{span}(I \cap C_k) \subsetneq C_k$.

  Then $D_k \subseteq \mathrm{span}(I \cap C_k)$.

To show this statement, assume for contradiction that $D_k \not\subseteq \mathrm{span}(I \cap C_k)$.

We will first show the existence of distinct elements $e_1$ and $f_1$ such that $e_1, f_1 \in \delta(v_1)$ for some $v_1 \in V$ and $f_1 \in A \setminus I$ while $e_1 \in I \setminus A$. Then we will use the pair $e_1, f_1$ to show the existence of another pair $e_2, f_2$ such that $e_2, f_2 \in \delta(v_2)$ where $f_2 \neq f_1$ and $f_2 \in A \setminus I$ while $e_2 \in I \setminus A$. In this manner, for any $t \in \mathbb{Z}_+$ we will be able to show *distinct* elements $f_1, f_2, \ldots, f_t$ that belong to $A$. However $A$ has only $n$ elements, a contradiction. Then we can conclude that our assumption $D_k \not\subseteq \mathrm{span}(I \cap C_k)$ is wrong. The following is our starting claim.

[6]The original statement in [3] claims this property only for pairs of bases (maximal independent sets), but it is equivalent to Fact 3.1. Indeed, if we consider the rank$(E)$-truncation of the direct sum of $(E, \mathcal{I})$ and a free matroid whose rank is rank$(E)$, then the axiom in [3] applied to this new matroid implies Fact 3.1 for $(E, \mathcal{I})$.

Claim 3.3. *There exists $v_1 \in V$ such that there are $e_1, f_1 \in \delta(v_1)$ satisfying the following properties:*

1. $f_1 \in A \setminus I$, $I_1 \coloneqq (I \cap C_k) + f_1 \in \mathcal{I}$, $I_1 \subseteq E(\mathcal{C})$, *and* $\mathsf{lev}_{\mathcal{C}}(f_1) = k$,
2. $e_1 \in I_1 \setminus A$ *and* $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1) \le k$.

*Proof.* Since $\mathcal{D}$ is a dual certificate of $A$, we have $\mathrm{span}(A \cap D_k) = D_k$. So $D_k \not\subseteq \mathrm{span}(I \cap C_k)$ implies that $\mathrm{span}(A \cap D_k) \not\subseteq \mathrm{span}(I \cap C_k)$. Hence $A \cap D_k \not\subseteq \mathrm{span}(I \cap C_k)$. So there exists $f_1$ such that $f_1 \in A \cap D_k$ and $f_1 \notin \mathrm{span}(I \cap C_k)$.

Since $D_k \subseteq C_k$, we have $f_1 \in D_k \subseteq C_k$. We also have $D_{k-1} \subseteq C_{k-1} = \mathrm{span}(I \cap C_{k-1}) \subseteq \mathrm{span}(I \cap C_k) \not\ni f_1$. Hence $f_1 \in C_k \setminus C_{k-1}$ and $f_1 \in D_k \setminus D_{k-1}$, i.e., $\mathsf{lev}_{\mathcal{C}}(f_1) = \mathsf{lev}_{\mathcal{D}}(f_1) = k$.

Since $f_1 \in A \subseteq E(\mathcal{D})$ and $\mathsf{lev}_{\mathcal{C}}(f_1) = \mathsf{lev}_{\mathcal{D}}(f_1)$, we have $f_1 \in E(\mathcal{C})$ by Claim 3.2. As $I \subseteq E(\mathcal{C})$, we then have $I_1 \coloneqq (I \cap C_k) + f_1 \subseteq E(\mathcal{C})$. Also, $I_1 \in \mathcal{I}$ by $f_1 \notin \mathrm{span}(I \cap C_k)$. Since $\mathsf{lev}_{\mathcal{C}}(f_1) = k$, the set $I_1 = (I \cap C_k) + f_1$ is lexicographically better than $I$. Then, the lex-maximality of $I$ implies that $I_1$ must violate the partition matroid constraint, i.e., there exists $e_1 \in I_1$ such that $e_1 \neq f_1$ and $e_1, f_1 \in \delta(v_1)$ for some $v_1 \in V$.

We have $\mathsf{lev}_{\mathcal{C}}(e_1) \le k$ as $e_1 \in I_1 \setminus \{f_1\} = I \cap C_k$. Since $f_1 \in \delta(v_1) \cap A$ and $|\delta(v_1) \cap A| \le 1$, we have $e_1 \notin A$. Note that $f_1 \in E(\mathcal{D})$ implies $\mathsf{lev}_{\mathcal{D}}(f_1) \ge \mathsf{lev}_{\mathcal{D}}(e_1) - 1$ and $e_1 \in E(\mathcal{C})$ implies $\mathsf{lev}_{\mathcal{C}}(e_1) \ge \mathsf{lev}_{\mathcal{C}}(f_1) - 1$. Note also that, for any element $e \in E$, we have $\mathsf{lev}_{\mathcal{D}}(e) \ge \mathsf{lev}_{\mathcal{C}}(e)$ because $D_i \subseteq C_i$ for all $i$.

- If $f_1 \succ_{v_1} e_1$, then $\mathsf{lev}_{\mathcal{C}}(e_1) > \mathsf{lev}_{\mathcal{C}}(f_1)$ by $e_1 \in E(\mathcal{C})$,[7] and hence $\mathsf{lev}_{\mathcal{D}}(f_1) \ge \mathsf{lev}_{\mathcal{D}}(e_1) - 1 \ge \mathsf{lev}_{\mathcal{C}}(e_1) - 1 \ge \mathsf{lev}_{\mathcal{C}}(f_1)$. As we have $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_{\mathcal{C}}(f_1)$, all the equalities hold.
- If $e_1 \succ_{v_1} f_1$, then $\mathsf{lev}_{\mathcal{D}}(f_1) > \mathsf{lev}_{\mathcal{D}}(e_1)$ by $f_1 \in E(\mathcal{D})$, and hence $\mathsf{lev}_{\mathcal{D}}(f_1) \ge \mathsf{lev}_{\mathcal{D}}(e_1) + 1 \ge \mathsf{lev}_{\mathcal{C}}(e_1) + 1 \ge \mathsf{lev}_{\mathcal{C}}(f_1)$. As we have $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_{\mathcal{C}}(f_1)$, all the equalities hold.
- If $f_1 \sim_{v_1} e_1$, then $\mathsf{lev}_{\mathcal{C}}(e_1) \ge \mathsf{lev}_{\mathcal{C}}(f_1)$ by $e_1 \in E(\mathcal{C})$; also $\mathsf{lev}_{\mathcal{D}}(f_1) \ge \mathsf{lev}_{\mathcal{D}}(e_1)$ by $f_1 \in E(\mathcal{D})$. Hence, we have $\mathsf{lev}_{\mathcal{D}}(f_1) \ge \mathsf{lev}_{\mathcal{D}}(e_1) \ge \mathsf{lev}_{\mathcal{C}}(e_1) \ge \mathsf{lev}_{\mathcal{C}}(f_1)$. Since $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_{\mathcal{C}}(f_1)$, all the equalities hold.

Thus in all the cases, we have $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1) \le k$ and $e_1 \in I_1 \setminus A$. ∎

Our next claim is the following. Recall that $I_1 \coloneqq (I \cap C_k) + f_1 \in \mathcal{I}$.

Claim 3.4. *There exists $v_2 \in V$ such that there are $e_2, f_2 \in \delta(v_2)$ satisfying the following properties:*

1. $f_2 \in A \setminus I_1$, $I_2 \coloneqq I_1 - e_1 + f_2 \in \mathcal{I}$, $I_2 \subseteq E(\mathcal{C})$, *and* $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{C}}(f_2)$,
2. $e_2 \in I_2 \setminus A$ *and* $\mathsf{lev}_{\mathcal{C}}(e_2) = \mathsf{lev}_{\mathcal{D}}(e_2) \le k$.

*Proof.* We know from Claim 3.3 that $I_1 = (I \cap C_k) + f_1 \in \mathcal{I}$. The set $I_1$ satisfies $\mathrm{span}(I_1 \cap C_i) = \mathrm{span}(I \cap C_i) = C_i$ for each $1 \le i \le k-1$; this is because $I_1 \cap C_i = I \cap C_i$ for each $i \le k-1$. Let us apply the exchange axiom in Fact 3.1 to $I_1, A \in \mathcal{I}$ and $e_1 \in I_1 \setminus A$. Since $A$ is maximal in $\mathcal{I}$, we have $A + e_1 \notin \mathcal{I}$, and hence there exists $f_2 \in A \setminus I_1$ such that $I_1 - e_1 + f_2$ and $A + e_1 - f_2$ are in $\mathcal{I}$.

Using that $\mathrm{span}(A \cap D_i) = D_i$ for $1 \le i \le q$, from $e_1 \notin \mathrm{span}(A - f_2)$ we obtain $\mathsf{lev}_{\mathcal{D}}(f_2) \le \mathsf{lev}_{\mathcal{D}}(e_1)$: indeed, assuming $\mathsf{lev}_{\mathcal{D}}(f_2) = \ell \ge 2$ we get $D_{\ell-1} = \mathrm{span}(A \cap D_{\ell-1}) \subseteq \mathrm{span}(A - f_2)$, which implies $e_1 \notin D_{\ell-1}$ and hence also $\mathsf{lev}_{\mathcal{D}}(e_1) \ge \ell = \mathsf{lev}_{\mathcal{D}}(f_2)$. Similarly, from $f_2 \notin \mathrm{span}(I_1 - e_1)$, $\mathsf{lev}_{\mathcal{C}}(e_1) \le k$, and $\mathrm{span}(I_1 \cap C_i) = C_i$ for $1 \le i \le k-1$, we obtain $\mathsf{lev}_{\mathcal{C}}(e_1) \le \mathsf{lev}_{\mathcal{C}}(f_2)$. Thus we have $\mathsf{lev}_{\mathcal{C}}(e_1) \le \mathsf{lev}_{\mathcal{C}}(f_2) \le \mathsf{lev}_{\mathcal{D}}(f_2) \le \mathsf{lev}_{\mathcal{D}}(e_1) = \mathsf{lev}_{\mathcal{C}}(e_1)$, implying all the equalities. Hence we have

$$f_2 \in A \setminus I_1, \quad \mathsf{lev}_{\mathcal{C}}(f_2) = \mathsf{lev}_{\mathcal{D}}(f_2), \quad \mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{C}}(f_2).$$

As $f_2 \in A \subseteq E(\mathcal{D})$, Claim 3.2 implies $f_2 \in E(\mathcal{C})$.

Observe that $I_2 \coloneqq I_1 - e_1 + f_2 = (I \cap C_k) + f_1 - e_1 + f_2 \subseteq E(\mathcal{C})$, and recall $I_2 \in \mathcal{I}$. Since $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{C}}(f_2)$ and $\mathsf{lev}_{\mathcal{C}}(f_1) = k$, $I_2$ is lexicographically better than $I$. This implies that $I_2$ must violate the partition matroid constraint. By the same argument as used in Claim 3.3 to show $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1)$, we see that there exists $e_2$ such that $e_2, f_2 \in \delta(v_2)$ for some $v_2 \in V$, satisfying

$$e_2 \in I_2 \setminus A, \quad \mathsf{lev}_{\mathcal{C}}(e_2) = \mathsf{lev}_{\mathcal{D}}(e_2) \le k.$$

This completes the proof of this claim. ∎

[7] Actually, the case $f_1 \succ_{v_1} e_1$ is impossible because $\mathsf{lev}_{\mathcal{C}}(e_1) > \mathsf{lev}_{\mathcal{C}}(f_1)$ contradicts $\mathsf{lev}_{\mathcal{C}}(e_1) \le k = \mathsf{lev}_{\mathcal{C}}(f_1)$. We write the proof in this form because the proofs of Claims 3.4 and 3.5 refer to the argument here to apply it to $e_j, f_j$, where $\mathsf{lev}_{\mathcal{C}}(f_j) = k$ is not assumed.

Note that $f_2 \neq f_1$ since $f_1 \in I_1$ and $f_2 \in A \setminus I_1$. Let $t \in \mathbb{Z}_+$. As shown in Claim 3.4 for $t = 3$, suppose we have constructed for $2 \leq j \leq t-1$:

1. $f_j \in A \setminus I_{j-1},\ I_j \coloneqq I_{j-1} - e_{j-1} + f_j \in \mathcal{I},\ I_j \subseteq E(\mathcal{C}),$ and $\mathsf{lev}_{\mathcal{C}}(e_{j-1}) = \mathsf{lev}_{\mathcal{C}}(f_j)$,
2. $e_j \in I_j \setminus A$ and $\mathsf{lev}_{\mathcal{C}}(e_j) = \mathsf{lev}_{\mathcal{D}}(e_j) \leq k$.

For each $j$ with $2 \leq j \leq t-1$, note that $I_j$ satisfies $\mathrm{span}(I_j \cap C_i) = \mathrm{span}(I \cap C_i) = C_i$ for each $1 \leq i \leq k-1$. Indeed, since $\mathsf{lev}_{\mathcal{C}}(e_{j-1}) = \mathsf{lev}_{\mathcal{C}}(f_j)$, we have $|I_j \cap C_i| = |I \cap C_i| = \mathrm{rank}(C_i)$ for each $i \leq k-1$. This implies $\mathrm{span}(I_j \cap C_i) = C_i$. Claim 3.5 generalizes Claim 3.4 for any $t \geq 3$.

Claim 3.5. *There exists $v_t \in V$ such that there are $e_t, f_t \in \delta(v_t)$ satisfying the following properties:*

1. $f_t \in A \setminus I_{t-1},\ I_t \coloneqq I_{t-1} - e_{t-1} + f_t \in \mathcal{I},\ I_t \subseteq E(\mathcal{C}),$ *and* $\mathsf{lev}_{\mathcal{C}}(e_{t-1}) = \mathsf{lev}_{\mathcal{C}}(f_t)$,
2. $e_t \in I_t \setminus A$ *and* $\mathsf{lev}_{\mathcal{C}}(e_t) = \mathsf{lev}_{\mathcal{D}}(e_t) \leq k$.

*Proof.* Let us apply the exchange axiom in Fact 3.1 to $I_{t-1}, A \in \mathcal{I}$ and $e_{t-1} \in I_{t-1} \setminus A$. Since $A + e_{t-1} \notin \mathcal{I}$, there exists $f_t \in A \setminus I_{t-1}$ such that $I_{t-1} - e_{t-1} + f_t$ and $A + e_{t-1} - f_t$ are in $\mathcal{I}$.

By the conditions $\mathrm{span}(A \cap D_i) = D_i$ for $1 \leq i \leq q$ we have $\mathsf{lev}_{\mathcal{D}}(f_t) \leq \mathsf{lev}_{\mathcal{D}}(e_{t-1})$, and by $\mathrm{span}(I_{t-1} \cap C_i) = C_i$ for $1 \leq i \leq k-1$ and $\mathsf{lev}_{\mathcal{C}}(e_{t-1}) \leq k$ we have $\mathsf{lev}_{\mathcal{C}}(e_{t-1}) \leq \mathsf{lev}_{\mathcal{C}}(f_t)$. Then $\mathsf{lev}_{\mathcal{C}}(e_{t-1}) \leq \mathsf{lev}_{\mathcal{C}}(f_t) \leq \mathsf{lev}_{\mathcal{D}}(f_t) \leq \mathsf{lev}_{\mathcal{D}}(e_{t-1}) = \mathsf{lev}_{\mathcal{C}}(e_{t-1})$, and hence all the equalities hold.

So we have $f_t \in A \setminus I_{t-1}, \mathsf{lev}_{\mathcal{C}}(f_t) = \mathsf{lev}_{\mathcal{D}}(f_t)$, and $\mathsf{lev}_{\mathcal{C}}(e_{t-1}) = \mathsf{lev}_{\mathcal{C}}(f_t)$. As $f_t \in A \subseteq E(\mathcal{D})$, Claim 3.2 implies $f_t \in E(\mathcal{C})$.

Observe that $I_t \coloneqq I_{t-1} - e_{t-1} + f_t = (I \cap C_k) + f_1 - e_1 + \ldots + f_{t-1} - e_{t-1} + f_t \subseteq E(\mathcal{C})$, and recall $I_t \in \mathcal{I}$. Since $\mathsf{lev}_{\mathcal{C}}(e_{j-1}) = \mathsf{lev}_{\mathcal{C}}(f_j)$ for $2 \leq j \leq t$ and $\mathsf{lev}_{\mathcal{C}}(f_1) = k$, the set $I_t$ is lexicographically better than $I$. This implies that $I_t$ must violate the partition matroid constraint. By the same argument as used in Claim 3.3 to show $\mathsf{lev}_{\mathcal{C}}(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1)$, we see that there exists $e_t$ such that $e_t, f_t \in \delta(v_t)$ for some $v_t$, satisfying also $e_t \in I_t \setminus A$ and $\mathsf{lev}_{\mathcal{C}}(e_t) = \mathsf{lev}_{\mathcal{D}}(e_t) \leq k$. This completes the proof of this claim. ◻

Observe that $f_t$ is distinct from $f_1, \ldots, f_{t-1}$ since $\{f_1, \ldots, f_{t-1}\} \subseteq I_{t-1}$ while $f_t \in A \setminus I_{t-1}$. Thus, for each $t \in \mathbb{Z}_+$, we have shown distinct elements $f_1, \ldots, f_t$ in $A$, contradicting that $|A| \leq n$. Therefore, it has to be the case that $D_k \subseteq \mathrm{span}(I \cap C_k)$.

This completes the proof of Lemma 3.1. ◻

We conclude this section with the proof of Lemma 2.1, which was postponed in Section 2.

***Proof of Lemma 2.1.*** The optimal value of LP1 is at least 0 since $\mathsf{wt}_A(A) = 0$. Thus if there exists a feasible solution $(\vec{y}, \vec{\alpha})$ to LP2 whose objective value is 0, then $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2. Since the optimal value of LP2 is 0, $A$ is a popular arborescence in $G$.

If $A$ is a popular arborescence, then the optimal value of LP2 is 0. We will now show there always exists an optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 that satisfies properties 1-4.

1. It is a well-known fact on matroid intersection (see [29, Theorem 41.12] or [15, Lecture 12, Claim 2]) that there exists an integral optimal solution to LP2 such that the support of the dual variables corresponding to the matroid $M$ is a chain. Thus property 1 follows.

2. Among all the optimal solutions to LP2 that satisfy property 1, let $(\vec{y}, \vec{\alpha})$ be the one that minimizes $\sum_{C \in \mathcal{C}} |\mathrm{span}(C) \setminus C|$, where $\mathcal{C}$ is the support of $\vec{y}$. We claim that $\mathrm{span}(A \cap C) = C$ holds for all $C \in \mathcal{C}$. Observe that each $C \in \mathcal{C}$ satisfies $y_C > 0$, and hence complementary slackness implies that the characteristic vector $\vec{x}$ of $A$ satisfies $\sum_{e \in C} x_e = \mathrm{rank}(C)$, i.e., $|A \cap C| = \mathrm{rank}(C)$. Therefore, to obtain $\mathrm{span}(A \cap C) = C$ for all $C \in \mathcal{C}$, it suffices to show $\mathrm{span}(C) = C$ for all $C \in \mathcal{C}$. Suppose to the contrary that it does not hold. Then there exists at least one $C \in \mathcal{C}$ with $\mathrm{span}(C) \neq C$. Among all such $C$, let $C^* \in \mathcal{C}$ be the maximal one.

Define $\vec{z}$ as follows: (i) $z_{\mathrm{span}(C^*)} = y_{\mathrm{span}(C^*)} + y_{C^*}$, (ii) $z_{C^*} = 0$, and (iii) $z_S = y_S$ for all other $S \subseteq E$. Then $\mathcal{C}' = (\mathcal{C} \setminus \{C^*\}) \cup \{\mathrm{span}(C^*)\}$ is the support of $\vec{z}$. Note that $\mathcal{C}'$ is again a chain because any $C \in \mathcal{C}$ with $C^* \subsetneq C$ satisfies $\mathrm{span}(C) = C$ by the choice of $C^*$, hence $\mathrm{span}(C^*) \subseteq \mathrm{span}(C) = C$.

Observe that $(\vec{z}, \vec{\alpha})$ is a feasible solution to LP2. Moreover, since $\mathrm{rank}(C^*) = \mathrm{rank}(\mathrm{span}(C^*))$, it does not change the objective value. Thus $(\vec{z}, \vec{\alpha})$ is an optimal solution to LP2 that satisfies property 1 and $\sum_{C \in \mathcal{C}'} |\mathrm{span}(C) \setminus C| < \sum_{C \in \mathcal{C}} |\mathrm{span}(C) \setminus C|$. This contradicts the choice of $(\vec{y}, \vec{\alpha})$.

3. Suppose $(\vec{y}, \vec{\alpha})$ satisfies properties 1–2 but not property 3. If $\emptyset \in \mathcal{C}$, then remove $\emptyset$ from $\mathcal{C}$ and modify $\vec{y}$ by setting $y_\emptyset = 0$. This does not change the objective value and does not violate feasibility constraints.

If $E \notin \mathcal{C}$, then add $E$ to $\mathcal{C}$ and modify $(\vec{y}, \vec{\alpha})$ by (i) setting $y_E = 1$ and (ii) decreasing every $\alpha_v$ value by 1. Since $\operatorname{rank}(E) = |V|$, the objective value does not change. Also, all constraints in LP2 are preserved. Hence the new solution satisfies properties 1–3.

4. Among all the optimal solutions to LP2 that satisfy properties 1–3, let $(\vec{y}, \vec{\alpha})$ be the one that minimizes $\sum_{S \subseteq E} y_S$ and let $\mathcal{C}$ be the support of $\vec{y}$. Note that $\alpha_v = -\sum_{C \in \mathcal{C}: A(v) \in C} y_C$ holds for any $v \in V$ by complementary slackness (observe that $x_{A(v)} > 0$ for $A$'s characteristic vector $\vec{x}$).

Suppose $y_{C^*} \geq 2$ for some $C^* \in \mathcal{C}$. Define $(\vec{z}, \vec{\beta})$ as follows: $z_{C*} = y_{C^*} - 1$ and $z_S = y_S$ for every other $S \subseteq E$. For any $v \in V$, let $\beta_v = -\sum_{C \in \mathcal{C}: A(v) \in C} z_C$. We will show below that $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2. Let us first see what is the objective value attained by $(\vec{z}, \vec{\beta})$.

This value is $\sum_{C \in \mathcal{C}} \operatorname{rank}(C) \cdot z_C + \sum_{v \in V} \beta_v$. When compared to $\sum_{C \in \mathcal{C}} \operatorname{rank}(C) \cdot y_C + \sum_{v \in V} \alpha_v$, the first term has decreased by $\operatorname{rank}(C^*)$ and the second term has increased by $|\{ v \in V : A(v) \in C^* \}| = |A \cap C^*| \leq \operatorname{rank}(C^*)$. Thus the objective value does not increase.

We will now show that $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2, that is, $\sum_{C \in \mathcal{C}: e \in C} z_C + \beta_v \geq \mathsf{wt}_A(e)$ for each $e \in \delta(v)$, $v \in V$. Since $(\vec{y}, \vec{\alpha})$ is feasible and the first term $\sum_{C \in \mathcal{C}: e \in C} z_C$ decreases by at most 1 and the second term $\beta_v = -\sum_{C \in \mathcal{C}: A(v) \in C} z_C$ never decreases, the only case we need to worry about is when the first term decreases and the second term does not increase. This implies that $e \in C^*$ and $A(v) \notin C^*$; hence $\sum_{C \in \mathcal{C}: e \in C} z_C + \beta_v = \sum_{C \in \mathcal{C}: e \in C} z_C - \sum_{C \in \mathcal{C}: A(v) \in C} z_C \geq z_{C*} \geq 1 \geq \mathsf{wt}_A(e)$. Thus $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2; furthermore, it is an optimal solution to LP2. Since $\sum_{S \subseteq E} z_S < \sum_{S \subseteq E} y_S$, this contradicts the choice of $(\vec{y}, \vec{\alpha})$.

Thus, we have shown that $(\vec{y}, \vec{\alpha})$ satisfies properties 1–3 and $y_C = 1$ for all $C \in \mathcal{C}$. Since we have $\alpha_v = -\sum_{C \in \mathcal{C}: A(v) \in C} y_C$, it follows that $\alpha_v = -|\{ C \in \mathcal{C} : A(v) \in C \}|$ for each $v \in V$. $\square$

## 4 Popular Colorful Forests

This section proves Corollary 1.1 and Theorem 1.3 (in terms of the popular colorful forest problem). Let $H = (U_H, E_H)$ be an undirected graph where $E_H = E_1 \uplus \cdots \uplus E_n$, i.e., $E_H$ is partitioned into $n$ color classes. Equivalently, there are $n$ agents $1, \ldots, n$ where agent $i$ owns the elements in $E_i$. For each $i$, there is a partial order $\succ_i$ over elements in $E_i$.

Recall that $S \subseteq E_H$ is a *colorful forest* if (i) $S$ is a forest in $H$ and (ii) $|S \cap E_i| \leq 1$ for every $i \in \{1, \ldots, n\}$. We refer to Section 1 on how every agent compares any pair of colorful forests; for any pair of colorful forests $F$ and $F'$, let $\phi(F, F')$ be the number of agents that prefer $F$ to $F'$.

Definition 4.1. *A colorful forest $F$ is popular if $\phi(F, F') \geq \phi(F', F)$ for any colorful forest $F'$.*

The popular colorful forest problem is to decide if a given instance $H$ admits a popular colorful forest or not. We will now show that Algorithm 1 solves the popular colorful forest problem.

Observe that a popular colorful forest is a popular common independent set in the intersection of the partition matroid defined by $E_H = E_1 \uplus \cdots \uplus E_n$ and the graphic matroid of $H$. In order to use the popular common *base* algorithm to solve this problem, we will augment the ground set $E_H$.

**An auxiliary instance $G$.** For each $i \in \{1, \ldots, n\}$, add a dummy edge $e_i = (u_i, v_i)$ with endpoints $u_i, v_i$, where $u_i$ and $v_i$ are new vertices that we introduce; call the resulting graph $G$. The vertex and edge sets of $G = (U, E)$ are given by $U = U_H \cup \bigcup_{i=1}^n \{u_i, v_i\}$ and $E = E_H \cup \bigcup_{i=1}^n \{e_i\}$. Furthermore, for each $i$, the edge $e_i$ will be the *worst* element in $i$'s preference order $\succ_i$, i.e., every $f \in E_i$ satisfies $f \succ_i e_i$.

In the setting of general matroids, $n$ dummy elements $e_1, \ldots, e_n$ are being introduced into the ground set $E$ as *free* elements, i.e., for any $i$, no set $S \subseteq E$ such that $e_i \notin S$ can span $e_i$. The partitions in the constructed matroid are $E_i \cup \{e_i\}$ for all $i \in \{1, \ldots, n\}$.

Observe that there exists a one-to-one correspondence between colorful forests in $H$ and colorful forests of size $n$ in $G$. Suppose $F_H$ is a colorful forest in $H$ and let $C \subseteq \{1, \ldots, n\}$ be the set of colors missing in $F_H$, i.e., $F_H \cap E_i = \emptyset$ exactly if $i \in C$. Let $F_G = F_H \cup \bigcup_{i \in C} \{e_i\}$. Then $F_G$ is a colorful forest of size $n$ in $G$. Conversely, given a colorful forest $F_G$ of size $n$ in $G$, we can obtain a colorful forest $F_H$ in $H$ by deleting the dummy elements.

**Colorful forests in $G$.** Let $F_H$ and $F'_H$ be colorful forests in $H$ and let $F_G$ and $F'_G$ be the corresponding forests (of size $n$) in $G$. Observe that $\phi(F_H, F'_H) = \phi(F_G, F'_G)$. Thus popular colorful forests in $H$ correspond to popular colorful forests of size $n$ in $G$ and vice-versa. We want popular colorful forests of size $n$ to be popular common *bases* in the intersection of the partition matroid and the graphic matroid of $G$.

Hence we will consider the $n$-truncation of the graphic matroid of $G$, i.e., all sets of size larger than $n$ will be deleted from the graphic matroid of $G$. The function $\text{rank}(\cdot)$ now denotes the rank function of the truncation and we have $\text{rank}(E) = n$. Thus solving the popular common base problem in the intersection of the partition matroid defined by the color classes on $E$ and the truncated graphic matroid of $G$ solves the popular colorful forest problem in $H$. Observe that such a reduction holds for the popular common independent set problem; hence Corollary 1.1 follows.

**The popular colorful forest polytope.** We will henceforth refer to a colorful forest of size $n$ in the auxiliary instance $G$ as a *colorful base* in $G$. Every popular colorful base $F$ in $G$ has a dual certificate as given in Lemma 2.1[8] and Lemma 2.2. We will now show these dual certificates are even more special than what is given in Lemma 2.2—along with the properties described there, the following property is also satisfied.

LEMMA 4.1. *Let $F$ be a popular colorful base in the auxiliary instance $G$ and let $\mathcal{C} = \{C_1, \dots, C_p\}$ be a dual certificate for $F$. Then $p \leq 2$.*

*Proof.* Suppose not, i.e., $p \geq 3$. From the definition of a dual certificate $\mathcal{C}$, we have $\emptyset \subsetneq C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_p = E$ (see Lemma 2.2). We will now show that $F \cap C_1 = \emptyset$. Since $\text{span}(F \cap C_1) = C_1$, this means $C_1 = \emptyset$; however this contradicts $C_1 \neq \emptyset$. This will give us the desired contradiction, proving $p \leq 2$.

In order to show that $F \cap C_1 = \emptyset$, it suffices to prove that for each $i \in \{1, \dots, n\}$, the unique element in $F \cap (E_i \cup \{e_i\})$, denoted by $F(i)$, is not contained in $C_1$.

- If $F(i) \neq e_i$, then the dummy edge $e_i$ is not in $F$. Since $e_i$ is not spanned by any set $S \subseteq E$ with $e_i \notin S$ and $\text{rank}(S) < n$, the condition $\text{span}(F \cap C_j) = C_j$, yielding also $|F \cap C_j| = \text{rank}(C_j)$, for all $j = 1, 2, \dots, p$ implies that $e_i \notin C_j$ for any $j < p$. Hence $\mathsf{lev}_{\mathcal{C}}(e_i) = p$, which implies that every edge in $E(\mathcal{C}) \cap E_i$ has level either $p$ or $p-1$. Because $p \geq 3$, this means that no edge of $C_1$ is present in $E(\mathcal{C}) \cap E_i$. Thus we have $F(i) \notin C_1$.
- If $F(i) = e_i$, then $e_i \in E(\mathcal{C})$. This implies $\mathsf{lev}_{\mathcal{C}}(e_i) > 1$ because $e_i$ is the worst element in $E_i \cup \{e_i\}$. Hence $F(i)$ is not in $C_1$.

In both cases, $F(i) \notin C_1$ for any $i \in \{1, \dots, n\}$. Thus we have $F \cap C_1 = \emptyset$, as desired. ◻

Lemma 4.1 shows that any dual certificate $\mathcal{C}$ for a popular colorful base $F$ in $G$ has length at most 2, i.e., $F$ has a dual certificate either of the form $\mathcal{C} = \{E\}$ or of the form $\mathcal{C} = \{C, E\}$. Let $F$ be the popular colorful base computed by Algorithm 1 in $G$ and let $\mathcal{C}$ be a dual certificate for $F$. The following lemma shows that if preferences are weak rankings, then $\mathcal{C}$ is a dual certificate for all popular colorful bases. Note that this proof crucially uses the fact that preferences are weak rankings—recall that we use this assumption in Theorem 1.3 as well. Indeed, assuming weak rankings is indispensable there, since the min-cost popular colorful forest problem for partial order preferences is NP-hard, due to the NP-hardness of its special case, the min-cost popular branching problem with partial order preferences [19].

LEMMA 4.2. *Assume that preferences are weak rankings and suppose that $F$ is the popular colorful base computed by Algorithm 1 in the auxiliary instance $G$, and $\mathcal{C}$ is a dual certificate for $F$. Then for any arbitrary popular colorful base $F'$ in $G$, we have (i) $F' \subseteq E(\mathcal{C})$ and (ii) if $\mathcal{C} = \{C, E\}$, then $|F' \cap C| = \text{rank}(C)$.*

*Proof.* Let $(\vec{y}, \vec{\alpha})$ be the dual variables defined from $\mathcal{C}$ as given in Lemma 2.1. That is, $y_{\hat{C}} = 1$ for each $\hat{C} \in \mathcal{C}$ and $y_S = 0$ for any other $S \subseteq E$, and $\alpha_i = -|\{\hat{C} \in \mathcal{C} : F(i) \in \hat{C}\}|$ for every $i \in \{1, \dots, n\}$. Note that the length of $\mathcal{C}$ is at most two by Lemma 4.1.

Consider LP1 and LP2 defined with respect to $F$. Since both $F$ and $F'$ are popular, their characteristic vectors are both optimal solutions to LP1. Since $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2, if $\mathcal{C} = \{C, E\}$ then we have

[8]In LP1 and LP2 defined with respect to $F$, the set $\delta(v)$ for $v \in V$ will be replaced by $E_i \cup \{e_i\}$ for $i \in \{1, \dots, n\}$, and in the definition of $\mathsf{wt}_F$, the edge $A(v)$ will be replaced by the unique element in $F \cap (E_i \cup \{e_i\})$, denoted by $F(i)$.

$|F' \cap C| = \mathrm{rank}(C)$ by complementary slackness. Then, what is left is to show $F' \subseteq E(\mathcal{C})$. We consider the cases where the length of $\mathcal{C}$ is one and two.

1. Suppose $\mathcal{C} = \{E\}$. Let $\mathcal{D}$ be a dual certificate of $F'$ as described in Lemma 2.2. Then $F' \subseteq E(\mathcal{D})$. Assume that $\mathcal{D} = \{D, E\}$ (otherwise $\mathcal{D} = \{E\} = \mathcal{C}$).

   Take any $i \in \{1, \ldots, n\}$. We now show $F'(i) \in E(\mathcal{C})$. If $F'(i) \in D$ then $\mathsf{lev}_{\mathcal{D}}(F'(i)) = 1 = \mathsf{lev}_{\mathcal{C}}(F'(i))$; along with $F'(i) \in E(\mathcal{D})$, this implies $F'(i) \in E(\mathcal{C})$ by Claim 3.2. We thus assume that $F'(i) \notin D$.

   Since the characteristic vector $\vec{x}$ of $F$ and $\vec{x}'$ of $F'$ are optimal solutions to LP1 (defined with respect to $F$) and $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2 (its dual LP), we will use complementary slackness. Because $x_{F(i)} = 1$, we have $\sum_{\hat{C} \in \mathcal{C}: F(i) \in \hat{C}} y_{\hat{C}} + \alpha_i = \mathsf{wt}_F(F(i)) \,(= 0)$. Similarly, because $x'_{F'(i)} = 1$, we have $\sum_{\hat{C} \in \mathcal{C}: F'(i) \in \hat{C}} y_{\hat{C}} + \alpha_i = \mathsf{wt}_F(F'(i))$. By subtracting the former from the latter, we obtain

   $$\sum_{\hat{C} \in \mathcal{C}: F'(i) \in \hat{C}} y_{\hat{C}} \;-\; \sum_{\hat{C} \in \mathcal{C}: F(i) \in \hat{C}} y_{\hat{C}} \;=\; \mathsf{wt}_F(F'(i)). \tag{4.2}$$

   Since $\mathcal{C} = \{E\}$, the left hand side is $1 - 1 = 0$. By this $\mathsf{wt}_F(F'(i)) = 0$, which implies $F(i) \sim_i F'(i)$. The fact $F(i) \in E(\mathcal{C})$ implies that $F(i)$ is maximal with respect to $\succ_i$ in $E_i \cup \{e_i\}$. Because $\succ_i$ is a weak ranking, $F(i) \sim_i F'(i)$ means that $F'(i)$ is also maximal, and hence $F'(i) \in E(\mathcal{C})$ follows.

2. Suppose $\mathcal{C} = \{C, E\}$. Let $\mathcal{D}$ be a dual certificate of $F'$. Then we have $\mathcal{D} = \{D, E\}$ and $D \subseteq C$ (by Lemma 3.1). Take any $i \in \{1, \ldots, n\}$. We now show $F'(i) \in E(\mathcal{C})$. If $F'(i) \notin C$ (resp., if $F'(i) \in D$), then $F'(i) \notin D$ (resp., $F'(i) \in C$); hence $\mathsf{lev}_{\mathcal{C}}(F'(i)) = \mathsf{lev}_{\mathcal{D}}(F'(i))$. This fact along with $F'(i) \in E(\mathcal{D})$ implies that $F'(i) \in E(\mathcal{C})$, by Claim 3.2. Therefore, let us assume that $F'(i) \in C \setminus D$.

   By the same analysis as given in Case 1, Equation (4.2) holds. Let us also consider LP1 and LP2 defined with respect to $F'$ (instead of $F$). Let $(\vec{z}, \vec{\beta})$ be the optimal solution of LP2 corresponding to $\mathcal{D}$. As before, the characteristic vectors of $F$ and $F'$ are optimal solutions to LP1. By the same argument (with $F'$, $F$ and $\mathcal{D}$ taking the places of $F$, $F'$, and $\mathcal{C}$, resp.), we have:

   $$\sum_{\hat{D} \in \mathcal{D}: F(i) \in \hat{D}} z_{\hat{D}} \;-\; \sum_{\hat{D} \in \mathcal{D}: F'(i) \in \hat{D}} z_{\hat{D}} \;=\; \mathsf{wt}_{F'}(F(i)). \tag{4.3}$$

   Since $F'(i) \in C$, the left hand side of (4.2) is 1 or 0, and so is $\mathsf{wt}_F(F'(i))$, which implies that we have $F'(i) \succ_i F(i)$ or $F'(i) \sim_i F(i)$. Furthermore, since $F'(i) \notin D$, the left hand side of (4.3) is 1 or 0, and so is $\mathsf{wt}_{F'}(F(i))$, which implies that $F(i) \succ_i F'(i)$ or $F(i) \sim_i F'(i)$. Therefore we must have $F'(i) \sim_i F(i)$. Hence $F(i) \in C$ follows from (4.2).

   We have shown that $F'(i) \sim_i F(i)$ and $F(i) \in C$. We also have $F'(i) \in C$. Since $F(i) \in E(\mathcal{C})$, we see that $F(i)$ is maximal in $C \cap (E_i \cup \{e_i\})$ and dominates all elements in $(E_i \cup \{e_i\}) \setminus C$ with respect to $\succ_i$. Since $\succ_i$ is a weak ranking and $F'(i) \sim_i F(i)$, the element $F'(i) \in C$ also satisfies these conditions, and hence $F'(i) \in E(\mathcal{C})$.

Thus we have $F'(i) \in E(\mathcal{C})$ for every $i \in \{1, \ldots, n\}$. Hence $F' \subseteq E(\mathcal{C})$. ∎

By Lemma 4.2, any popular colorful base $F'$ in $G$ satisfies $F' \subseteq E(\mathcal{C})$ and $|F' \cap C| = \mathrm{rank}(C)$ if $\mathcal{C} = \{C, E\}$. Conversely, any popular colorful base $F'$ in $G$ that satisfies these conditions is popular by Lemma 2.2. Therefore the set of all popular colorful bases in $G$ can be described as a face of the matroid intersection polytope. Since a popular colorful forest in the given instance $H$ is obtained by deleting the dummy elements from popular colorful bases in $G$, Theorem 1.3 follows.

We also state this result explicitly in Theorem 4.1 in the setting of popular colorful forests. Let $\mathcal{C} = \{C, E\}$ be a dual certificate for the popular colorful base $F$ in $G$ computed by Algorithm 1.

Theorem 4.1. *If preferences are weak rankings, an extension of the popular colorful forest polytope of the given instance $H$ is defined by the constraints $\sum_{e \in C} x_e = \mathrm{rank}(C)$ and $x_e = 0$ for all $e \in E \setminus E(\mathcal{C})$ along with all the constraints of LP1.*

## 5 Min-Cost Popular Arborescence

We prove Theorem 1.4 in this section. We present a reduction from the Vertex Cover problem, whose input is an undirected graph $H$ and an integer $k$, and asks whether $H$ admits a set of $k$ vertices that is a vertex cover, that is, contains an endpoint from each edge in $H$.

Our reduction is strongly based on the reduction used in [19, Theorem 6.3] which showed the NP-hardness of the min-cost popular branching problem when vertices have partial order preferences. Recall that the min-cost popular branching problem is polynomial-time solvable when vertices have weak rankings [19] (also implied by Theorem 1.3). Note also that neither the hardness of min-cost popular branching for partial order preferences [19], nor the hardness of min-cost popular assignment for strict preferences [18] implies Theorem 1.4, since the min-cost popular arborescence problem with strict rankings does not contain either of these problems.

To show the NP-hardness of the min-cost popular arborescence problem when vertices have strict rankings, we construct a directed graph $G = (V \cup \{r\}, E = E_1 \cup E_2 \cup E_3)$ as follows; see Figure 1 for an illustration. We set

$$\begin{aligned} V &= \{w\} \cup \{v_0, v_1 : v \in V(H)\} \cup \{e_u, e_v : e = uv \in E(H)\}, \\ E_1 &= \{(e_u, e_v), (e_v, e_u), (e_u, w), (e_v, w) : e = uv \in E(H)\} \\ &\quad \cup \{(v_0, v_1), (v_1, v_0) : v \in V(H)\}, \\ E_2 &= \{(r, w)\} \cup \{(w, x) : x \in V(G) \setminus \{r, w\}\}, \\ E_3 &= \{(r, v_1) : v \in V(H)\} \cup \{(u_0, e_u), (v_0, e_v) : e = uv \in E(H)\}. \end{aligned}$$

To define the preferences of each vertex in $G$, we let all vertices prefer edges of $E_1$ to edges of $E_2$, which in turn are preferred to edges of $E_3$. Whenever some vertex has more than one incoming edge in some $E_i$, $i \in \{1, 2, 3\}$, then it orders them in some arbitrarily fixed strict order. We set the cost of each edge in $E_3$, as well as the cost of all edges entering $w$ except for $(r, w)$ as $\infty$. We set the cost of $(w, v_1)$ as 1 for each $v \in V(H)$, and we set the cost of all remaining edges as 0. We define our budget to be $k$, finishing the construction of our instance of min-cost popular arborescence.

We are going to show that $H$ admits a vertex cover of size at most $k$ if and only if $G$ has a popular arborescence of cost at most $k$.

Suppose first that $A$ is a popular arborescence in $G$ with cost at most $k$. We prove that the set $S = \{v \in V(H) : (w, v_1) \in A\}$ is a vertex cover in $H$. Since each edge $(w, v_1)$ has cost 1, our budget implies $|S| \leq k$.

For a vertex $v \in V(H)$ and an edge $e = uv \in E(H)$, let $A_v = A \cap (\delta(v_0) \cup \delta(v_1))$ and $A_e = A \cap (\delta(e_u) \cup \delta(e_v))$, respectively. We note that any $v \in V(H)$ satisfies that $A_v$ is either $\{(w, v_0), (v_0, v_1)\}$ or $\{(w, v_1), (v_1, v_0)\}$. Indeed, if it is not the case, we have $A_v = \{(w, v_0), (w, v_1)\}$, since $A$ is an arborescence with finite cost. However, this contradicts the popularity of $A$, since $A \setminus \{(w, v_1)\} \cup \{(v_0, v_1)\}$ is more popular than $A$. We can similarly show that each $e = uv \in E(H)$ satisfies that $A_e$ is either $\{(w, e_u), (e_u, e_v)\}$ or $\{(w, e_v), (e_v, e_u)\}$. Note also that $(r, w) \in A$, as all other edges entering $w$ have infinite cost.

Assume for the sake of contradiction that $S$ is not a vertex cover of $H$, i.e., there exists an edge $e = uv \in E(H)$ such that neither $(w, u_1)$ nor $(w, v_1)$ is contained in $A$. Then we have $A_u = \{(w, u_0), (u_0, u_1)\}$ and $A_v = (w, v_0), (v_0, v_1)\}$. By symmetry, we assume without loss of generality that $A_e = \{(w, e_u), (e_u, e_v)\}$. Define an edge set $A'$ by

$$A' = (A \setminus (A_e \cup A_v \cup \{(r, w)\})) \cup \{(r, v_1), (v_1, v_0), (v_0, e_v), (e_v, e_u), (e_u, w)\}.$$

We can see that $A'$ is an arborescence and is more popular than $A$, since three vertices, $v_0$, $e_u$, and $w$, prefer $A'$ to $A$, while two vertices, $v_1$ and $e_v$, prefer $A$ to $A'$, and all others are indifferent between them. This proves that $S$ is a vertex cover of $H$.

For the other direction, assume that $S$ is a vertex cover in $H$. We construct a popular arborescence $A$ of cost $|S|$ in $G$. For each $e \in E(H)$ we fix an endpoint $\sigma(e)$ of $e$ that is contained in $S$, and we denote by $\bar{\sigma}(e)$ the other endpoint of $e$ (which may or may not be in $S$). Let

$$\begin{aligned} A = \{(r, w)\} &\cup \{(w, v_1), (v_1, v_0) : v \in S\} \\ &\cup \{(w, v_0), (v_0, v_1) : v \in V(H) \setminus S\} \\ &\cup \{(w, e_{\bar{\sigma}(e)}), (e_{\bar{\sigma}(e)}, e_{\sigma(e)}) : e \in E(H)\}. \end{aligned}$$

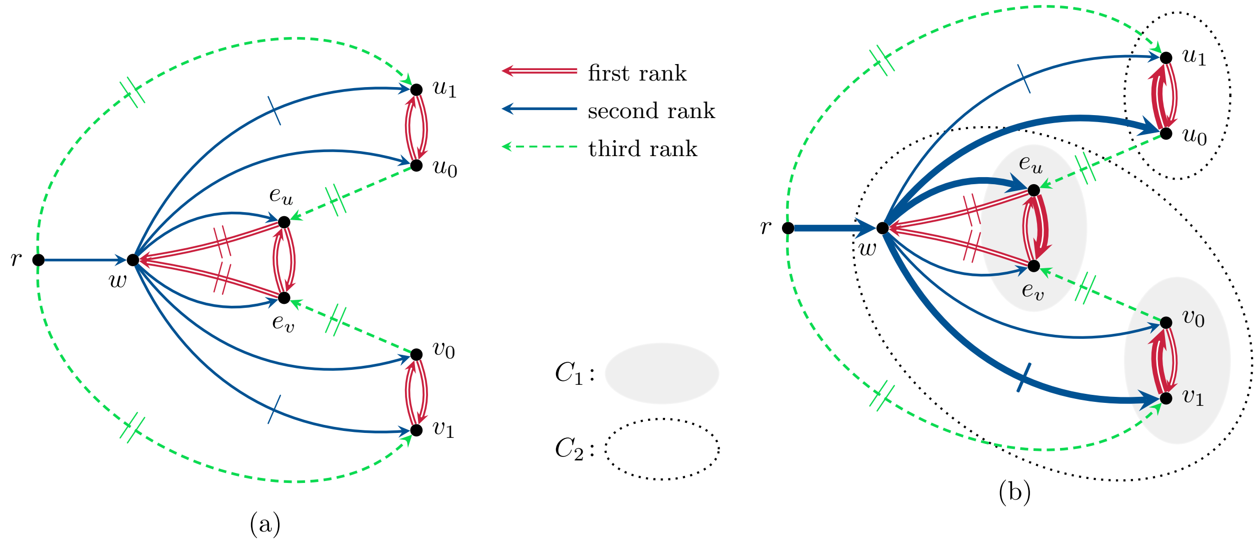



Figure 1: Illustration of the reduction in the proof of Theorem 1.4. Figure (a) illustrates the construction showing a subgraph of $G$, assuming that the input graph $H$ contains an edge $e = uv$. Edges in $E_1$, $E_2$, and $E_3$ are depicted with double red, single blue, and dashed green lines, respectively. Edges marked with two, one, and zero crossbars have cost $\infty$, 1, and 0, respectively. Figure (b) illustrates the popular arborescence $A$ in bold, assuming $v \in S$ and $u \notin S$. The chain $C_1 \subsetneq C_2 \subsetneq C_3 = E$ certifying the popularity of $A$ is shown using grey and dotted ellipses for edges in $C_1$ and $C_2$, respectively.

It is straightforward to verify that $A$ is an arborescence and its cost is exactly $|S|$. Hence it remains to prove its popularity, which is done by showing a dual certificate $\mathcal{C}$ for $A$.

To define $\mathcal{C}$, let us first define a set $X = \{w\} \cup \{e_u, e_v : e = uv \in E(H)\} \cup \{v_0, v_1 : v \in S\}$ of vertices in $G$. Then we set $\mathcal{C} = \{C_1, C_2, C_3\}$ where

$$\begin{aligned} C_1 &= \{(e_u, e_v), (e_v, e_u) : e = uv \in E(H)\} \cup \{(v_0, v_1), (v_1, v_0) : v \in S\}, \\ C_2 &= \{f \in E(H) : f \text{ has two endpoints in } X\} \cup \{(v_0, v_1), (v_1, v_0) : v \in V(H) \setminus S\}, \\ C_3 &= E. \end{aligned}$$

Let us first check that $\mathrm{rank}(C_i) = |A \cap C_i|$ for each $C_i \in \mathcal{C}$. Clearly, $C_1$ consists of mutually vertex-disjoint 2-cycles, and $A$ contains an edge from each of them. Thus $\mathrm{rank}(C_1) = |A \cap C_1|$ follows. The edge set $C_2$ consists of all edges induced by the vertices of $X$, together with another set of mutually vertex-disjoint 2-cycles that share no vertex with $X$. It is easy to verify that $A \cap C_2$ contains an edge from each of the 2-cycles in question, as well as a directed tree containing all vertices of $X$. Thus, $\mathrm{rank}(C_2) = |A \cap C_2|$ holds. Since $A$ is an arborescence, $\mathrm{rank}(C_3) = \mathrm{rank}(E) = |V| = |A \cap C_3|$ is obvious. Observe that for each $i \in \{1, 2, 3\}$ we have $\mathrm{span}(C_i) = C_i$, and hence $\mathrm{rank}(C_i) = |A \cap C_i|$ implies $\mathrm{span}(A \cap C_i) = C_i$.

It remains to see that $A \subseteq E(\mathcal{C})$. First, $A(w) = (r, w)$ is the unique incoming edge of $w$ with $\mathcal{C}$-level 3. For some $v \in S$, $\mathsf{lev}^*_{\mathcal{C}}(v_0) = 2$ while $\mathsf{lev}^*_{\mathcal{C}}(v_1) = 3$, and by their preferences both $A(v_0) = (v_1, v_0)$ and $A(v_1) = (w, v_1)$ are in $E(\mathcal{C})$. For some $v \in V(H) \setminus S$, $\mathsf{lev}^*_{\mathcal{C}}(v_0) = \mathsf{lev}^*_{\mathcal{C}}(v_1) = 3$, and hence both $A(v_0) = (w, v_0)$ and $A(v_1) = (v_0, v_1)$ are in $E(\mathcal{C})$. Finally, consider an edge $e = uv \in E(H)$ with $\sigma(e) = v \in S$. As $\mathsf{lev}^*_{\mathcal{C}}(e_u) \leq 3$, and since $e_u$ prefers $(w, e_u)$ to $(u_0, e_u)$, we know that the edge $A(e_u) = (w, e_u) \in C_2$ is contained in $E(\mathcal{C})$. By contrast, since $v \in S$ implies $v_0 \in X$, we obtain $\mathsf{lev}^*_{\mathcal{C}}(e_v) = 2$, and therefore the edge $A(e_v) = (e_u, e_v) \in C_1$ is contained in $E(\mathcal{C})$. By Lemma 2.2, this proves that $A$ is indeed a popular arborescence.

## 6 Popular Arborescences with Forced/Forbidden Edges

We prove Theorem 1.5 in this section. Observe that the problem of deciding if there exists a popular arborescence $A$ such that $A \supseteq E^+$ for a given set $E^+ \subseteq E$ of *forced* edges can be reduced to the problem of deciding if there exists a popular arborescence $A$ such that certain edges are *forbidden* for $A$.

Let $V' \subseteq V$ be the set of those vertices $v$ such that $\delta(v) \cap E^+ \neq \emptyset$; clearly, we may assume $|\delta(v) \cap E^+| = 1$ for each $v \in V'$. Let $E' = \bigcup_{v \in V'} (\delta(v) \setminus E^+)$. Since $A \supseteq E^+$ if and only if $A \cap E' = \emptyset$, it follows that the problem of deciding if there exists a popular arborescence $A$ such that $E^+ \subseteq A$ and $E^- \cap A = \emptyset$ reduces to the problem of deciding if there exists a popular arborescence $A$ such that $A \cap E_0 = \emptyset$ for a set $E_0 \subseteq E$ of forbidden edges.

**Forbidden edges.** We present our algorithm that decides if $G$ admits a popular arborescence that avoids $E_0$ for a given subset $E_0$ of $E$ as Algorithm 2. The only difference from the original popular arborescence algorithm (Algorithm 1) is in line 4: the new algorithm finds a lexicographically maximal branching in the set $E(\mathcal{C}) \setminus E_0$ instead of $E(\mathcal{C})$. Recall that $\text{rank}(E) = |V| = n$.

**Algorithm 2** The popular arborescence algorithm with the forbidden edge set $E_0$

1: Initialize $p = 1$ and $C_1 = E$. ▷ Initially we set $\mathcal{C} = \{E\}$.
2: **while** $p \leq n$ **do**
3: Compute the edge set $E(\mathcal{C})$ from the current multichain $\mathcal{C}$.
4: Find a branching $I \subseteq E(\mathcal{C}) \setminus E_0$ that lexicographically maximizes $(|I \cap C_1|, \dots, |I \cap C_p|)$.
5: **if** $|I \cap C_i| = \text{rank}(C_i)$ for every $i = 1, \dots, p$ **then** return $I$.
6: Let $k$ be the minimum index such that $|I \cap C_k| < \text{rank}(C_k)$.
7: Update $C_k \leftarrow \text{span}(I \cap C_k)$.
8: **if** $k = p$ **then** $p \leftarrow p + 1$, $C_p \leftarrow E$, and $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_p\}$.
9: Return "$G$ has no popular arborescence that avoids $E_0$".

Theorem 6.1. *Let $E_0 \subseteq E$. The instance $G = (V \cup \{r\}, E)$ admits a popular arborescence $A$ such that $A \cap E_0 = \emptyset$ if and only if Algorithm 2 returns a popular arborescence with no edge of $E_0$.*

*Proof.* The easy side is to show that if Algorithm 2 returns an arborescence $I$, then (i) $I$ is popular and (ii) $I \cap E_0 = \emptyset$. As done in Section 3, let us prune the multichain $\mathcal{C}$ into a chain $\mathcal{C}'$. Because $I \subseteq E(\mathcal{C}) \setminus E_0$ and $E(\mathcal{C}) \subseteq E(\mathcal{C}')$, we have $I \subseteq E(\mathcal{C}') \setminus E_0$. Since $I \subseteq E(\mathcal{C}')$ and $|I \cap C'_i| = \text{rank}(C'_i)$ (and hence $\text{span}(I \cap C'_i) = C'_i$) for every $C'_i \in \mathcal{C}'$, it follows from Lemma 2.2 that $I$ is a popular arborescence.

We now show the converse. Suppose that $G$ admits a popular arborescence $A$ with $A \cap E_0 = \emptyset$. Let $\mathcal{D} = \{D_1, \dots, D_q\}$ be a dual certificate for $A$. Then we have $A \subseteq E(\mathcal{D}) \setminus E_0$. It suffices to show that Algorithm 2 maintains the following invariant: the multichain $\mathcal{C} = \{C_1, \dots, C_p\}$ maintained in the algorithm satisfies $p \leq q$ and $D_i \subseteq C_i$ for any $i = 1, 2, \dots, p$.

We can show a variant of Lemma 3.1, i.e., we can show that when $C_k$ is updated in the algorithm, $D_k \subseteq \text{span}(I \cap C_k)$ holds where $I$ is a lexicographically maximal branching in $E(\mathcal{C}) \setminus E_0$. The proof of Lemma 3.1 works almost as it is. Recall that we sequentially find elements $f_1, e_1, f_2, e_2, \dots$ in the proof of Lemma 3.1. For each $j = 1, 2, \dots,$ in addition to the condition $f_j \in E(\mathcal{C})$, we have $f_j \notin E_0$ since $f_j \in A \subseteq E \setminus E_0$. By this, $I_j = (I \cap C_k) + f_1 - e_1 + f_2 \cdots - e_{j-1} + f_j$ satisfies $I_j \subseteq E(C) \setminus E_0$ for each $j$. Hence the proof of Lemma 3.1 works with "lex-maximality subject to $I \subseteq E(\mathcal{C}) \setminus E_0$" replacing "lex-maximality subject to $I \subseteq E(\mathcal{C})$". ◻

## 7 Conclusions

We considered the popular arborescence problem, which asks to determine whether a given directed rooted graph, in which vertices have preferences over incoming edges, admits a popular arborescence or not and to find one if so. We provided a polynomial-time algorithm to solve this problem, which affirmatively answers an open problem posed in 2019 [22]. Our algorithm and its correctness proof work in the generality of matroid intersection (where one of the matroids is a partition matroid), which means that we also solved the popular common base problem. Furthermore, we observed that the popular common independent set problem, which includes the popular colorful forest problem as a special case, can be reduced to the popular common base problem, and hence can be solved by our algorithm. Utilizing structural observations, we also proved that the min-cost popular common independent set problem is tractable if preferences are weak rankings.

On the intractability side, we proved that the min-cost popular arborescence problem and the $k$-unpopularity margin arborescence problem are both NP-hard even for strict preferences. Note that the min-cost problem is NP-hard for *popular common bases* (a fact implied by the NP-hardness of the popular assignment problem shown in [18], as well as by Theorem 1.4), while it is tractable for *popular common independent sets* if preferences are weak rankings by Theorem 1.3. By analogy, one may expect the problem of finding a common independent set with unpopularity margin at most $k$ to be polynomial-time solvable. However, this is not the case (unless $\mathsf{P} = \mathsf{NP}$), since the $k$-unpopularity matching problem is NP-hard even for strict rankings [24]. Note that the $k$-unpopularity margin branching problem is polynomial-time solvable when preferences are weak rankings, as shown in [19], but this does not contradict the above fact: branchings and matchings are both special cases of common independent sets (where one matroid is a partition matroid), but neither of them includes the other.

**Acknowledgments**

We are grateful for inspiring discussions on the popular arborescence problem to Chien-Chung Huang, Satoru Iwata, Tamás Király, Jannik Matuschke, and Ulrike Schmidt-Kraepelin. We thank the anonymous reviewers for their valuable comments. Telikepalli Kavitha is supported by the Department of Atomic Energy, Government of India, under project no. RTI4001. Kazuhisa Makino is partially supported by JSPS KAKENHI Grant Numbers JP20H05967, JP19K22841, and JP20H00609. Ildikó Schlotter is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and its János Bolyai Research Scholarship, and by the Hungarian Scientific Research Fund (OTKA grant K124171). Yu Yokoi is supported by JST PRESTO Grant Number JPMJPR212B. This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society."

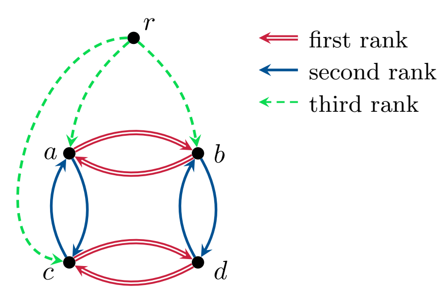
## Appendix A Examples of Algorithm Execution

We illustrate how Algorithm 1 works using some examples. We provide three instances of the popular arborescence problem. In all of these instances, a digraph is given as $G = (V \cup \{r\}, E)$ with $V = \{a, b, c, d\}$, and each node $v \in V$ has a strict preference on $\delta(v)$. For better readability, for a multichain $\mathcal{C} = \{C_1, \ldots, C_p\}$ with $C_1 \subseteq \cdots \subseteq C_p$ we will also use the notation $\langle C_1, \ldots, C_p \rangle$.

**A.1 Example 1.** This instance is similar to the one illustrated in Section 1; the only difference is that now the edge $(r, d)$ is deleted. In contrast to the case where $(r, d)$ exists, this instance admits a popular arborescence, which is found by Algorithm 1 as follows.

The preference orders for the four vertices are as follows:

$$(b,a) \succ_a (c,a) \succ_a (r,a)$$
$$(a,b) \succ_b (d,b) \succ_b (r,b)$$
$$(d,c) \succ_c (a,c) \succ_c (r,c)$$
$$(c,d) \succ_d (b,d).$$




For convenience, we denote by $E_1$, $E_2$, and $E_3$ the sets of the first, second and third choice edges, respectively. That is, $E_1 = \{(b,a), (a,b), (d,c), (c,d)\}$, $E_2 = \{(c,a), (d,b), (a,c), (b,d)\}$, and $E_3 = \{(r,a), (r,b), (r,c)\}$.

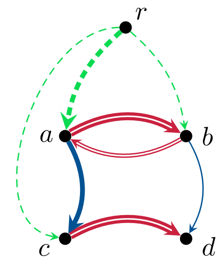
**Algorithm Execution.** Below we describe the steps in our algorithm.

1. $p = 1$ and $C_1 = E$. Then $E(\mathcal{C}) = E_1$ and $I = \{(a,b), (c,d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 2 < 4 = \text{rank}(C_1)$, the set $C_1$ is updated to $\text{span}(I \cap C_1) = E_1$. Since $C_1 = C_p$ is updated, $p$ is incremented and $E$ is added to $\mathcal{C}$ as $C_2$.

2. $p = 2$ and $\langle C_1, C_2 \rangle = \langle E_1, E \rangle$. Then $E(\mathcal{C}) = E_1 \cup E_2$ and $I = \{(a,b), (c,d), (a,c)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 2 = \text{rank}(C_1)$ and $|I \cap C_2| = 3 < 4 = \text{rank}(C_2)$, the set $C_2$ is updated to $\text{span}(I \cap C_2) = E_1 \cup E_2$. Since $C_2 = C_p$ is updated, $p$ is incremented and $E$ is added to $\mathcal{C}$ as $C_3$.

3. $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle E_1,\ E_1 \cup E_2,\ E\rangle$. Then $E(\mathcal{C}) = \{(c, d)\} \cup E_2 \cup E_3$ and $I = \{(c, d), (c, a), (d, b), (r, c)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 1 < 2 = \mathrm{rank}(C_1)$, the set $C_1$ is updated to $\mathrm{span}(I \cap C_1) = \{(c, d), (d, c)\}$.

4. $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle \{(c, d), (d, c)\},\ E_1 \cup E_2,\ E\rangle$. Then we have $E(\mathcal{C}) = \{(r, a), (b, a), (r, b), (a, b), (r, c), (a, c), (c, d), (b, d)\}$ (all edges on the figure to the right) and $I = \{(r, a), (a, b), (a, c), (c, d)\}$ (thick edges on the figure to the right) is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_i| = \mathrm{rank}(C_i)$ holds for $i = 1, 2, 3$, the algorithm returns $I$.

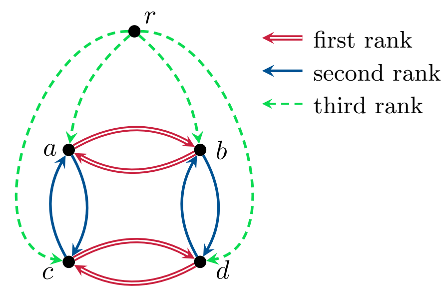



Note that $I' = \{(r, b), (b, a), (a, c), (c, d)\}$ is also a possible output of the algorithm. Indeed, both $I$ and $I'$ are popular arborescences.

**A.2 Example 2.** We next demonstrate how the algorithm works for an instance that admits no popular arborescences.

Consider the instance illustrated in the introduction. For the reader's convenience, we include the same figure again. As observed there, this instance has no popular arborescence.

We denote by $E_1$, $E_2$, and $E_3$ the sets of the first, second and third rank edges, respectively. Note that, unlike in Example 1, here $E_3$ contains $(r, d)$.




**Algorithm Execution**

1. The first step is the same as Step 1 in Example 1. That is, $p = 1$, $C_1 = E$, $E(\mathcal{C}) = E_1$, and $I = \{(a, b), (c, d)\}$ is found as a lex-maximal branching in $E(\mathcal{C})$. Then, $C_1$ is updated to $\mathrm{span}(I \cap C_1) = E_1$, $p$ is incremented, and $E$ is added to $\mathcal{C}$ as $C_2$.

2. The second step is also the same as Step 2 in Example 1. That is, $p = 2$, $\langle C_1, C_2\rangle = \langle E_1, E\rangle$, $E(\mathcal{C}) = E_1 \cup E_2$, and $I = \{(a, b), (c, d), (a, c)\}$ is found as a lex-maximal branching in $E(\mathcal{C})$. Then, $C_2$ is updated to $\mathrm{span}(I \cap C_2) = E_1 \cup E_2$, $p$ is incremented, and $E$ is added to $\mathcal{C}$ as $C_3$.

3. $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle E_1, E_1 \cup E_2, E\rangle$. Then $E(\mathcal{C}) = E_2 \cup E_3$ (compared to Example 1, here $(r, d)$ is included while $(c, d)$ is excluded) and $I = \{(a, c), (b, d), (r, a), (r, b)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 0 < 2 = \mathrm{rank}(C_1)$, the set $C_1$ is updated to $\mathrm{span}(I \cap C_1) = \emptyset$.

4. $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle \emptyset, E_1 \cup E_2, E\rangle$. Then $E(\mathcal{C}) = E_1 \cup E_3$ and $I = \{(a, b), (c, d), (r, a), (r, c)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = \mathrm{rank}(C_1)$ and $|I \cap C_2| = 2 < 3 = \mathrm{rank}(C_2)$, the set $C_2$ is updated to $\mathrm{span}(I \cap C_2) = E_1$.

5. $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle \emptyset, E_1, E\rangle$. Then $E(\mathcal{C}) = E_1 \cup E_2$ and $I = \{(a, b), (c, d), (a, c)\}$ is a lex-maximal branching in $E(\mathcal{C})$. (Observe that these $E(\mathcal{C})$ and $I$ are the same as Step 2.) Since $|I \cap C_i| = \mathrm{rank}(C_i)$ for $i = 1, 2$ and $|I \cap C_3| = 3 < 4 = \mathrm{rank}(C_3)$, the set $C_3$ is updated to $\mathrm{span}(I \cap C_3) = E_1 \cup E_2$, $p$ is incremented, and $E$ is added to $\mathcal{C}$ as $C_4$.

6. $p = 4$ and $\langle C_1, C_2, C_3, C_4\rangle = \langle \emptyset, E_1, E_1 \cup E_2, E\rangle$. Then, as in Step 3, $E(\mathcal{C}) = E_2 \cup E_3$ and $I = \{(a, c), (b, d), (r, a), (r, b)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = \mathrm{rank}(C_1)$ and $|I \cap C_2| = 0 < 2 = \mathrm{rank}(C_2)$, the set $C_2$ is updated to $\mathrm{span}(I \cap C_2) = \emptyset$.

7. $p = 4$ and $\langle C_1, C_2, C_3, C_4\rangle = \langle \emptyset, \emptyset, E_1 \cup E_2, E\rangle$. By the same argument as in Step 4, the set $C_3$ is updated to $E_1$.

8. $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle \emptyset, \emptyset, E_1, E \rangle$. By the same argument as in Step 5, the set $C_4$ is updated to $E_1 \cup E_2$, $p$ is incremented, and $E$ is added to $\mathcal{C}$ as $C_5$.

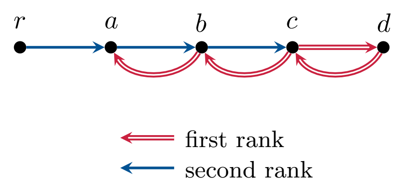
9. $p = 5$ and $\langle C_1, C_2, C_3, C_4, C_5 \rangle = \langle \emptyset, \emptyset, E_1, E_1 \cup E_2, E \rangle$. Since $p = 5 > 4 = n = |V|$, the algorithm halts with returning "$G$ has no popular arborescence."

The reader might observe that whenever $C_1$ becomes empty in the algorithm, then by Lemma 3.1 we can conclude that the instance admits no popular arborescence, since the dual certificate contains only non-empty sets (Lemma 2.2) and hence $D_1 \subseteq C_1 = \emptyset$ is not possible. Therefore, we could in fact stop the algorithm already in Step 3 when $C_1$ gets updated to $\emptyset$. Nevertheless, the algorithm will reach a correct answer even without using this observation, as illustrated by the above example.

**A.3 Example 3.** We next provide an example that shows the importance of considering multichains. During the algorithm's execution on this instance, $\mathcal{C}$ does become a multichain that is not a chain.

The preferences of the four vertices are as follows:

$$(b, a) \succ_a (r, a)$$
$$(c, b) \succ_b (a, b)$$
$$(d, c) \succ_c (b, c)$$
$$(c, d)$$

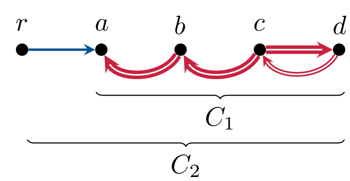



where $(c, d)$ is the unique incoming edge of $d$. For convenience, we denote by $E_{abcd}$, $E_{bcd}$, and $E_{cd}$ the edge sets of the induced subgraphs for the vertex sets $\{a, b, c, d\}$, $\{b, c, d\}$, and $\{c, d\}$, respectively. That is, $E_{abcd} = E \setminus \{(r, a)\}$, $E_{bcd} = \{(b, c), (c, b), (c, d), (d, c)\}$, and $E_{cd} = \{(c, d), (d, c)\}$. Note that $\{(r, a), (a, b), (b, c), (c, d)\}$ is the unique arborescence in this instance, and hence it is a popular arborescence.

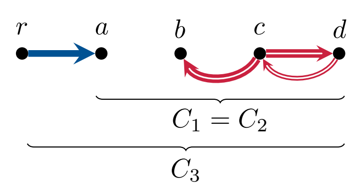
**Algorithm Execution**

1. $p = 1$ and $C_1 = E$. Then $E(\mathcal{C}) = \{(b, a), (c, b), (d, c), (c, d)\}$ and $I = \{(b, a), (c, b), (c, d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 3 < 4 = \mathrm{rank}(C_1)$, the set $C_1$ is updated to $\mathrm{span}(I \cap C_1) = E_{abcd}$. Since $C_1 = C_p$ is updated, $p$ is incremented and $E$ is added to $\mathcal{C}$ as $C_2$.

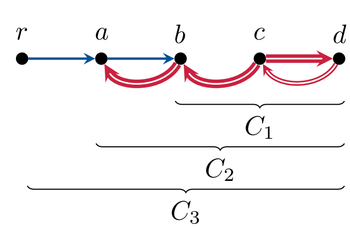
2. $p = 2$ and $\langle C_1, C_2 \rangle = \langle E_{abcd}, E \rangle$ (shown by braces on the right). Then $E(\mathcal{C}) = \{(r, a), (b, a), (c, b), (d, c), (c, d)\}$ (all edges on the right) and $I = \{(b, a), (c, b), (c, d)\}$ (thick edges on the right) is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = \mathrm{rank}(C_1)$ and $|I \cap C_2| = 3 < 4 = \mathrm{rank}(C_2)$, $C_2$ is updated to $\mathrm{span}(I \cap C_2) = E_{abcd}$. Since $C_2 = C_p$ is updated, $p$ is incremented and $E$ is added to $\mathcal{C}$ as $C_3$.



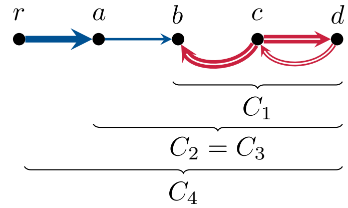

3. $p = 3$ and $\langle C_1, C_2, C_3 \rangle = \langle E_{abcd}, E_{abcd}, E \rangle$ (so $C_1 = C_2$). Then $E(\mathcal{C}) = \{(r, a), (c, b), (d, c), (c, d)\}$. Note that $(b, a)$ is not in $E(\mathcal{C})$ as $\mathsf{lev}_{\mathcal{C}}((b, a)) = 1$ while $\mathsf{lev}_{\mathcal{C}}((r, a)) = 3$. $I = \{(r, a), (c, b), (c, d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 2 < 3 = \mathrm{rank}(C_1)$, the set $C_1$ is updated to $\mathrm{span}(I \cap C_1) = E_{bcd}$.



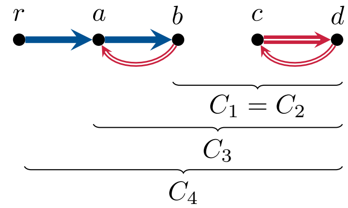

4. $p = 3$ and $\langle C_1, C_2, C_3 \rangle = \langle E_{bcd}, E_{abcd}, E \rangle$. Then, $E(\mathcal{C}) = E \setminus \{(b, c)\}$ and $I = \{(b, a), (c, b), (c, d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_i| = \mathrm{rank}(C_i)$ for $i = 1, 2$ and $|I \cap C_3| = 3 < 4 = \mathrm{rank}(C_3)$, the set $C_3$ is updated to $\mathrm{span}(I \cap C_3) = E_{abcd}$. Since $C_3 = C_p$ is updated, $p$ is incremented and $E$ is added to $\mathcal{C}$ as $C_4$.



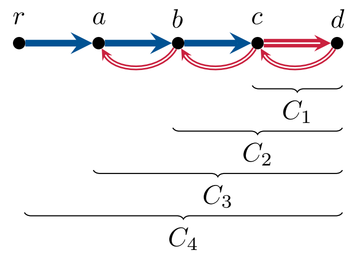
5. $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{bcd}, E_{abcd}, E_{abcd}, E \rangle$. $E(\mathcal{C}) = E \setminus \{(b,a),(b,c)\}$ and $I = \{(r,a),(c,b),(c,d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = \text{rank}(C_1)$ and $|I \cap C_2| = 2 < 3 = \text{rank}(C_2)$, the set $C_2$ is updated to $\text{span}(I \cap C_2) = E_{bcd}$.




6. $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{bcd}, E_{bcd}, E_{abcd}, E \rangle$. Then $E(\mathcal{C}) = E \setminus \{(b,c),(c,b)\}$ and $I = \{(r,a),(a,b),(c,d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_1| = 1 < 2 = \text{rank}(C_1)$, the set $C_1$ is updated to $\text{span}(I \cap C_1) = E_{cd}$.




7. $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{cd}, E_{bcd}, E_{abcd}, E \rangle$. Then $E(\mathcal{C}) = E$ and $I = \{(r,a),(a,b),(b,c),(c,d)\}$ is a lex-maximal branching in $E(\mathcal{C})$. Since $|I \cap C_i| = \text{rank}(C_i)$ holds for $i = 1,2,3,4$, the algorithm returns $I$.




## References


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

[2] F. C. Bock. An algorithm to construct a minimum directed spanning tree in a directed network. In B. Avi-Itzak, editor, *Developments in Operations Research*, pages 29–44. Gordon and Breach, New York, 1971.

[3] R. A. Brualdi. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1(2):161–167, 1969.

[4] Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

[5] M. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie Royale, 1785.

[6] Á. Cseh. Popular matchings. In U. Endriss, editor, *Trends in computational social choice*, chapter 6, pages 105–122. AI Access, 2017.

[7] A. Darmann. Popular spanning trees. *International Journal of Foundations of Computer Science*, 24(5):655 – 677, 2013.

[8] A. Darmann. It is difficult to tell if there is a Condorcet spanning tree. *Mathematical Methods of Operations Research*, 84(1):94 – 104, 2016.

[9] A. Darmann, C. Klamler, and U. Pferschy. Finding socially best spanning trees. *Theory and Decision*, 70(4):511 – 527, 2011.

[10] J. Edmonds. Optimum branchings. *Journal of Research of the National Institute of Standards*, 71B:233–240, 1967.

[11] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, 1970.

[12] A. Frank. A weighted matroid intersection algorithm. *Journal of Algorithms*, 2:328–336, 1981.

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

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

[15] M. Goemans. Combinatorial optimization. `http://www-math.mit.edu/~goemans/18453S17/18453.html`.

[16] S. Hardt and L. Lopes. Google votes: A liquid democracy experiment on a corporate social network. Technical report, Technical Disclosure Commons, 2015.

[17] N. Kamiyama. Popular matchings with ties and matroid constraints. *SIAM Journal on Discrete Mathematics*, 31(3):1801–1819, 2017.

[18] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, and U. Schmidt-Kraepelin. The popular assignment problem: when cardinality is more important than popularity. In *SODA 2022: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 103–123. SIAM, 2022.

[19] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, and U. Schmidt-Kraepelin. Popular branchings and their dual certificates. *Mathematical Programming*, 192(1):567–595, 2022.

[20] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, and U. Schmidt-Kraepelin. The popular assignment problem: when cardinality is more important than popularity, 2023. `arXiv:2110.10984` [cs.DS] (full version).

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

[22] T. Király. Popular arborescences. 9th Emléktábla Workshop (Matching Theory), `https://users.renyi.hu/~emlektab/emlektabla9problems.pdf`, 2019.

[23] M. Mahdian. Random popular matchings. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, pages 238–242. ACM, 2006.

[24] R. M. McCutchen. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of the 8th Latin American Conference on Theoretical Informatics (LATIN 2008)*, volume 4957 of *Lecture Notes in Computer Science*, pages 593–604. Springer, 2008.

[25] S. Merrill and B. Grofman. *A Unified Theory of Voting: Directional and Proximity Spatial Models*. Cambridge University Press, 1999.

[26] J. Mestre. Weighted popular matchings. *ACM Transactions on Algorithms*, 10(1)(2), 2014.

[27] K. Natsui and K. Takazawa. Finding popular branchings in vertex-weighted directed graphs. *Theoretical Computer Science*, 953, 2023.

[28] U. Schmidt-Kraepelin. Models and algorithms for scalable collective decision making. PhD Thesis, TU Berlin, 2022.

[29] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency. Vol. 24 of Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003.

[30] C. T. S. Sng and D. F. Manlove. Popular matchings in the weighted capacitated house allocation problem. *Journal of Discrete Algorithms*, 8(2):102–116, 2010.