

Popular Branchings and Their Dual Certificates^{*}

Telikepalli Kavitha¹, Tamás Király², Jannik Matuschke³, Ildikó Schlotter^{4,5},
and Ulrike Schmidt-Kraepelin⁶

¹ TIFR, Mumbai, India, kavitha@tifr.res.in

² Eötvös University, Budapest, Hungary, tkiraly@cs.elte.hu

³ Research Center for Operations Management, KU Leuven, Leuven, Belgium,
jannik.matuschke@kuleuven.be

⁴ Centre for Economic and Regional Studies, Budapest, Hungary

⁵ Budapest University of Technology and Economics, Budapest, Hungary,
ildi@cs.bme.hu

⁶ Technische Universität Berlin, Germany, u.schmidt-kraepelin@tu-berlin.de

Abstract. Let G be a digraph where every node has preferences over its incoming edges. The preferences of a node extend naturally to preferences over *branchings*, i.e., directed forests; a branching B is *popular* if B does not lose a head-to-head election (where nodes cast votes) against any branching. Such popular branchings have a natural application in liquid democracy. The popular branching problem is to decide if G admits a popular branching or not. We give a characterization of popular branchings in terms of *dual certificates* and use this characterization to design an efficient combinatorial algorithm for the popular branching problem. When preferences are weak rankings, we use our characterization to formulate the *popular branching polytope* in the original space and also show that our algorithm can be modified to compute a branching with *least unpopularity margin*. When preferences are strict rankings, we show that “approximately popular” branchings always exist.

1 Introduction

Let G be a directed graph where every node has preferences (in partial order) over its incoming edges. We assume that G can have parallel edges but no self-loops. In case G is simple, the preferences can equivalently be defined on in-neighbors. We define a *branching* to be a subgraph of G that is a directed forest where each node has in-degree at most 1; a node with in-degree 0 is a *root*. The *size* of a branching is the number of edges in it. The problem we consider is to find a branching that is *popular*.

Given any pair of branchings, we say a node u prefers the branching where it has a more preferred incoming edge (being a root is u 's worst choice). If neither incoming edge is preferred to the other, then u is indifferent between the two branchings. So any pair of branchings, say B and B' , can be compared by asking for the majority opinion, i.e., every node opts for the branching that it prefers,

^{*} A preliminary version of this work has appeared in the proceedings of IPCO 2020.

and it abstains if it is indifferent between them. Let $\phi(B, B')$ (resp., $\phi(B', B)$) be the number of nodes that prefer B (resp., B') in the B -vs- B' comparison. If $\phi(B', B) > \phi(B, B')$, then we say B' is *more popular* than B .

Definition 1.1. *A branching B is popular in G , if no branching in G is more popular than B . That is, $\phi(B, B') \geq \phi(B', B)$ for all branchings B' in G .*

An application in computational social choice. We see the main application of popular branchings within *liquid democracy*. Suppose there is an election where a specific issue should be decided upon, and there are several proposed alternatives. Every individual voter has an opinion on these alternatives, but might also consider certain other voters as being better informed than her. Liquid democracy is a novel voting scheme that provides a middle ground between the feasibility of representative democracy and the idealistic appeal of direct democracy [4]: Voters can choose whether they delegate their vote to another, well-informed voter or cast their vote themselves. As the name suggests, voting power flows through the underlying network, or in other words, delegations are transitive. During the last decade, this idea has been implemented within several online decision platforms such as *Sovereign* and *LiquidFeedback*⁷ and was used for internal decision making at Google [23] and political parties, such as the German *Pirate Party* or the Swedish party *Demoeax*.

In order to circumvent *delegation cycles*, e.g., a situation in which voter x delegates to voter y and vice versa, and to enhance the expressiveness of delegation preferences, several authors proposed to let voters declare a set of acceptable representatives [20] together with a preference relation among them [5, 23, 31]. A mechanism selects an approved representative for each voter, avoiding delegation cycles. Similarly as suggested in [6], we additionally assume voters to accept themselves as their least preferred approved representative.

This reveals the connection to branchings in simple graphs where nodes correspond to voters and the edge (x, y) indicates that voter x is an approved delegate of voter y .⁸ Every root in the branching casts a weighted vote on behalf of all her descendants. We assume that voters rate branchings only based on their predecessors. This is justified when approved delegates are considered to be more competent both in deciding on the issue as well as in assessing the competence of others. What is a good mechanism to select representatives for voters? A crucial aspect in liquid democracy is the *stability* of the delegation process [3, 14]. For the model described above, we propose popular branchings as a new concept of stability.

Not every directed graph admits a popular branching. Consider the following simple graph on four nodes a, b, c, d where a, b (similarly, c, d) are each other's top choices, while a, c (similarly, b, d) are each other's second choices. There is no edge between a, d (similarly, b, c). Consider the branching $B = \{(a, b), (a, c), (c, d)\}$. A more popular branching is $B' = \{(d, c), (c, a), (a, b)\}$. Observe that a and c

⁷ See www.democracy.earth and www.interaktive-demokratie.org, respectively.

⁸ Typically, such a delegation is represented by an edge (y, x) ; for the sake of consistency with downward edges in a branching, we use (x, y) .

prefer B' to B , while d prefers B to B' and b is indifferent between B and B' . We can similarly obtain a branching $B'' = \{(b, a), (b, d), (d, c)\}$ that is more popular than B' . It is easy to check that this instance has no popular branching.

1.1 Our Problem and Results

The popular branching problem is to decide if a given digraph G admits a popular branching or not, and if so, to find one. We show that determining whether a given branching B is popular is equivalent to solving a min-cost arborescence problem in an extension of G with appropriately defined edge costs (these edge costs are a *function* of the branching). The dual LP to this arborescence problem gives rise to a laminar set system that serves as a certificate for the popularity of B , if it is popular. This dual certificate is crucial in devising an algorithm for efficiently solving the popular branching problem.

Theorem 1.2. *Given a directed graph G where every node has preferences in arbitrary partial order over its incoming edges, there is a polynomial-time algorithm to decide if G admits a popular branching, and if so, to find one.*

The proof of Theorem 1.2 is presented in Section 3; it is based on a characterization of popular branchings that we develop in Section 2. In applications like liquid democracy, it is natural to assume that the preference order of every node is a *weak ranking*, i.e., a ranking of its incoming edges with possible ties. In this case, the proof of correctness of our popular branching algorithm leads to a formulation of the *popular branching polytope* \mathcal{B}_G , i.e., the convex hull of incidence vectors of popular branchings in G .

Theorem 1.3. *Let G be a digraph on n nodes and m edges where every node has a weak ranking over its incoming edges. The popular branching polytope of G admits a formulation of size $O(2^n)$ in \mathbb{R}^m , and has $\Omega(2^n)$ facets.*

Though the above formulation has an exponential number of constraints, it admits an efficient separation oracle. As a consequence, when G has edge costs and node preferences are weak rankings, the min-cost popular branching problem can be efficiently solved. Thus we can efficiently solve extensions of the popular branching problem where we minimize the largest rank used or allow constraints for forced/forbidden edges when node preferences are weak rankings. In contrast, we show that the min-cost popular branching problem is NP-hard when node preferences are in arbitrary partial order in Section 6.

Relaxing popularity. Since popular branchings need not always exist in G , this motivates relaxing popularity to *approximate popularity*—do approximately popular branchings always exist in any instance G ? An approximately popular branching B may lose an election against another branching, however the extent of this defeat will be bounded. There are two measures of unpopularity: *unpopularity factor* $u(\cdot)$ and *unpopularity margin* $\mu(\cdot)$, defined as follows:

$$u(B) = \max_{\phi(B', B) > 0} \frac{\phi(B', B)}{\phi(B, B')} \quad \text{and} \quad \mu(B) = \max_{B'} \phi(B', B) - \phi(B, B').$$

A branching B is popular if and only if $u(B) \leq 1$ or $\mu(B) = 0$. We show the following results.

Theorem 1.4. *A branching with minimum unpopularity margin in a digraph where every node has a weak ranking over its incoming edges can be efficiently computed. In contrast, when node preferences are in arbitrary partial order, the minimum unpopularity margin problem is NP-hard.*

Theorem 1.5. *Let G be a digraph where every node has a strict ranking over its incoming edges. Then there always exists a branching B in G with $u(B) \leq \lfloor \log n \rfloor$. Moreover, for every n , we can show an instance G_n on n nodes with strict rankings such that $u(B) \geq \lfloor \log n \rfloor$ for every branching B in G_n .*

Hardness results for restricted popular branching problems. A natural optimization problem here is to compute a popular branching where no tree is large. In liquid democracy, a large-sized tree shows a high concentration of power in the hands of a single voter, and this is harmful for social welfare [20]. When there is a fixed subset of root nodes in a directed graph, it was shown in [20] that it is NP-hard to find a branching that minimizes the size of the largest tree. To translate this result to popular branchings, we need to allow ties, whereas Theorem 1.6 below holds even for strict rankings. Another natural restriction is to limit the out-degree of nodes; Theorem 1.6 also shows that this variant is computationally hard.

Theorem 1.6. *Given a digraph G where each node has a strict ranking over its incoming edges, it is NP-hard to decide if there exists*

- (a) *a popular branching in G where each node has at most 9 descendants;*
- (b) *a popular branching in G with maximum out-degree at most 2.*

We also consider popular *mixed branchings*. A mixed branching P is a probability distribution (or lottery) over branchings in G , i.e., $P = \{(B_1, p_1) \dots, (B_k, p_k)\}$, where B_i is a branching in G for each i and $\sum_{i=1}^k p_i = 1$, $p_i \geq 0$ for all i . The notion of popularity extends naturally to mixed branchings and we show that popular mixed branchings always exist in G ; moreover, a popular mixed branching can be computed in polynomial time.

Background and Related Work. The notion of popularity was introduced by Gärdenfors [19] in 1975 in the domain of bipartite matchings. Algorithmic questions in popular matchings have been well-studied for the last 10-15 years [1, 2, 8, 9, 15, 16, 22, 24, 25, 26, 27, 29, 32]. Algorithms for popular matchings were first studied in the *one-sided* preferences model where vertices on only one side of the bipartite graph have preferences over their neighbors. Popular matchings need not always exist here and there is an efficient algorithm to solve the popular matching problem [1]. The functions unpopularity factor/margin were introduced in [32] to measure the *unpopularity* of a matching; it was shown in [32] that it is NP-hard to compute a matching that minimizes either of these

quantities. In the domain of bipartite matchings with *two-sided* strict preferences, popular matchings always exist since stable matchings always exist [18] and every stable matching is popular [19].

The concept of popularity has previously been applied to (undirected) spanning trees [10, 11, 12]. In contrast to our setting, voters have rankings over the entire edge set. This allows for a number of different ways to derive preferences over trees, most of which lead to hardness results.

Techniques. Starting from an LP formulation for determining the unpopularity margin of a given branching, we use LP duality to characterize popular branchings in terms of *dual certificates*. A dual certificate for a given branching is a family of subsets of the node set V exhibiting a certain combinatorial structure. A certificate of size k implies that the unpopularity margin of the branching is at most $n - k$, and thus a certificate of size n constitutes a proof that the branching is popular. This is analogous to characterizing popular matchings in bipartite graphs in terms of *witnesses* (see [15, 25, 27]). However, such witnesses are points in \mathbb{R}^n rather than set families and their structure is far simpler than that of dual certificates for popular branchings.

Our algorithm constructs a partition \mathcal{X}' of V such that if G admits popular branchings, then there has to be *some* popular branching in G with a dual certificate of size n supported by \mathcal{X}' (see Section 3). Moreover, when nodes have weak rankings, \mathcal{X}' supports some dual certificate of size n to *every* popular branching in G : this leads to the formulation of the popular branching polytope \mathcal{B}_G (see Section 6). Our positive results on low unpopularity branchings are extensions of our algorithm (these results are in Section 4 and Section 5).

Notation. The preferences of node v on its incoming edges are given by a strict partial order \prec_v , so $e \prec_v f$ means that v prefers edge f to edge e . We use $e \sim_v f$ to denote that v is indifferent between e and f , that is, neither $e \prec_v f$ nor $e \succ_v f$ holds. The relation \prec_v is a *weak ranking* if \sim_v is transitive. In this case, \sim_v is an equivalence relation and there is a strict order on the equivalence classes. When each equivalence class has size 1, we call it a *strict ranking*.

2 Dual Certificates

We add a dummy node r to $G = (V_G, E_G)$ as the *root* and make (r, v) the least preferred incoming edge of any node v in G . Let $D = (V \cup \{r\}, E)$ be the resulting graph where $V = V_G$ and $E = E_G \cup \{(r, u) : u \in V\}$. An *r -arborescence* in D is an out-tree with root r (throughout the paper, all arborescences are assumed to be rooted at r and to span V , unless otherwise stated).

Note that there is a one-to-one correspondence between branchings in G and arborescences in D (simply make r the parent of all roots of the branching). A branching is popular in G if and only if the corresponding arborescence is popular

among all arborescences in D .⁹ We will therefore prove our results for arborescences in D . The corresponding results for branchings in G follow immediately by projection, i.e., removing node r and its incident edges.

For an arborescence A in D , there is a simple way to check if A is popular in D . Let $A(v)$ be the incoming edge of v in A . For $e = (u, v)$ in D , define:

$$c_A(e) := \begin{cases} 0, & \text{if } e \succ_v A(v), \text{ i.e., } v \text{ prefers } e \text{ to } A(v); \\ 1, & \text{if } e \sim_v A(v), \text{ i.e., } v \text{ is indifferent between } e \text{ and } A(v); \\ 2, & \text{if } e \prec_v A(v), \text{ i.e., } v \text{ prefers } A(v) \text{ to } e. \end{cases}$$

For $S \subseteq E$ we define $c_A(S) = \sum_{e \in S} c_A(e)$. Observe that $c_A(A) = |V| = n$ since $c_A(e) = 1$ for every $e \in A$. Let A' be any arborescence in D and let $\Delta(A, A') = \phi(A, A') - \phi(A', A)$ be the difference in the number of votes for A and the number of votes for A' in the A -vs- A' comparison. Observe that $c_A(A') = \Delta(A, A') + n$. Since $\mu(A) = \max_{A'}(-\Delta(A, A')) = \max_{A'}(n - c_A(A'))$, we can conclude the following.

Proposition 2.1. *Let A' be a min-cost arborescence in D with respect to c_A . Then $\mu(A) = n - c_A(A')$. In particular, A is popular in D if and only if it is a min-cost arborescence in D with edge costs given by c_A .*

Consider the following linear program **LP1**, which computes a min-cost arborescence in D , and its dual **LP2**. For any non-empty $X \subseteq V$, let $\delta^-(X)$ be the set of edges entering the set X in the graph D .

$$\begin{aligned} & \text{minimize} && \sum_{e \in E} c_A(e) \cdot x_e && \text{(LP1)} \\ \text{subject to} &&& \sum_{e \in \delta^-(X)} x_e \geq 1 \quad \forall X \subseteq V, X \neq \emptyset \\ &&& x_e \geq 0 \quad \forall e \in E. \end{aligned}$$

$$\begin{aligned} & \text{maximize} && \sum_{X \subseteq V, X \neq \emptyset} y_X && \text{(LP2)} \\ \text{subject to} &&& \sum_{X: \delta^-(X) \ni e} y_X \leq c_A(e) \quad \forall e \in E \\ &&& y_X \geq 0 \quad \forall X \subseteq V, X \neq \emptyset. \end{aligned}$$

For any feasible solution y to **LP2**, let $\mathcal{F}_y := \{X \subseteq V : y_X > 0\}$ be the support of y . Inspired by Edmonds' branching algorithm [13], Fulkerson [17] gave an algorithm to find an optimal solution y to **LP2** such that y is integral. From an alternative proof (Theorem 6.13 in [30]), we get Lemma 2.2. A family of sets \mathcal{S} is called *laminar* if for any two sets $S, S' \in \mathcal{S}$ we have: $S \subseteq S'$ or $S' \subseteq S$ or $S \cap S' = \emptyset$.

⁹ Note that, by the special structure of D , an arborescence is popular among all arborescences in D if and only if it is a popular branching in D .

Lemma 2.2. *There exists an optimal, integral solution y^* to LP2 such that \mathcal{F}_{y^*} is laminar.*

Let y be an optimal, integral solution to LP2 such that \mathcal{F}_y is laminar. Note that for any nonempty $X \subseteq V$, there is an $e \in A \cap \delta^-(X)$ and thus $y_X \leq c_A(e) = 1$. This implies that $y_X \in \{0, 1\}$ for all X . We conclude that \mathcal{F}_y is a dual certificate for A in the sense of the following definition.

Definition 2.3. *A dual certificate for A is a laminar family $\mathcal{Y} \subseteq 2^V$ such that $|\{X \in \mathcal{Y} : e \in \delta^-(X)\}| \leq c_A(e)$ for all $e \in E$.*

Lemma 2.4. *The following three statements are equivalent:*

- (1) *A is popular.*
- (2) *There exists a dual certificate \mathcal{Y} for A with $|\mathcal{Y}| = n$.*
- (3) *There exists a dual certificate \mathcal{Y} for A with $|A \cap \delta^-(X)| = 1$ for all $X \in \mathcal{Y}$ and $|\{X \in \mathcal{Y} : e \in \delta^-(X)\}| = 1$ for all $e \in A$.*

Proof. Let \mathcal{Y} be a dual certificate for A of maximum cardinality and observe that \mathcal{Y} corresponds to an optimal solution to LP2 of value $|\mathcal{Y}|$. Recall from Proposition 2.1 that A is popular if and only if it is a min-cost arborescence in D with respect to c_A . Hence (1) is equivalent to A corresponding to an optimal solution to LP1. By LP duality, this is the case if and only if $c_A(A) = |\mathcal{Y}|$, which is equivalent to (2) because $c_A(A) = n$ by definition of c_A . Lastly, (3) is equivalent to the characteristic vectors of A and \mathcal{Y} fulfilling complementary slackness, which is again equivalent to A and \mathcal{Y} being optimal solutions. \square

For the rest of this section, let \mathcal{Y} be a dual certificate maximizing $|\mathcal{Y}|$.

Observation 2.5. *For every $v \in V$ we have $|\{X \in \mathcal{Y} : v \in X\}| \leq 2$.*

Observation 2.5 is implied by the fact that $e = (r, v)$ is an edge in D for every $v \in V$ and $c_A(e) \leq 2$. For the case when A is popular, Lemma 2.4 establishes the following one-to-one correspondence between the nodes in V and the sets of \mathcal{Y} : For every set $X \in \mathcal{Y}$, there is a unique edge $(u, v) \in A$ that enters X . We call v the *entry-point* for X . Conversely, we let Y_v be the unique set in \mathcal{Y} for which v is the entry-point; thus $\mathcal{Y} = \{Y_v : v \in V\}$. Laminarity of \mathcal{Y} yields the following corollary of Lemma 2.4 and Observation 2.5:

Corollary 2.6. *If $|\mathcal{Y}| = n$ and $w \in Y_v \setminus \{v\}$ for some $v \in V$, then $Y_w = \{w\}$.*

Proof. Assume $z \in Y_w \setminus \{w\}$. By laminarity either $Y_w \subseteq Y_v$ holds, in which case $z \in Y_w \cap Y_v \cap Y_z$, a contradiction to Observation 2.5, or $Y_v \subseteq Y_w$ implying that the edge from A entering Y_w enters two sets from \mathcal{Y} , namely Y_w and Y_v , a contradiction to $|\{X \in \mathcal{Y} : e \in \delta^-(X)\}| = 1$ for all $e \in A$ (see Lemma 2.4). \square

The following definition of the set of *safe* edges $S(X)$ with respect to a subset $X \subseteq V$ will be useful. Let $S(X)$ be the set of edges (u, v) in $E[X] := E \cap (X \times X)$ such that properties 1. and 2. hold:

1. (u, v) is *undominated* in $E[X]$, i.e., $(u, v) \not\prec_v (u', v) \forall (u', v) \in E[X]$.
2. (u, v) *dominates* $(w, v) \forall w \notin X$, i.e., $(u, v) \succ_v (w, v) \forall (w, v) \in \delta^-(X)$.

Observation 2.7. *If A is popular, then $A \cap E[X] \subseteq S(X)$ for all $X \in \mathcal{Y}$.*

Proof. Let $X \in \mathcal{Y}$ with $|X| > 1$. By Corollary 2.6, for every node $v \in X$ other than the entry-point for X we have $\{v\} = Y_v \in \mathcal{Y}$. So any edge $e \in \delta^-(v)$ within $E[X]$ enters exactly one dual set, i.e., $\{v\}$, implying $c_A(e) \geq 1$ by LP2. By contrast, any edge $(w, v) \in E$ where $w \notin X$ enters two of the dual sets: X and $\{v\}$, yielding $c_A((w, v)) \geq 2$. For the edge $(u, v) \in A \cap E[X]$, these conditions induce exactly the constraints 1. and 2. given above, showing that the edge $A(v)$ must be safe. \square

3 Popular Branching Algorithm

We are now ready to present our algorithm for deciding if D admits a popular arborescence or not. For each $v \in V$, step 1 of the algorithm builds the largest set X_v such that v can reach all nodes in X_v using edges in $S(X_v)$. To construct the sets X_v we make use of the *monotonicity* of $S(\cdot)$, which follows directly from the definition of safe edges:

Observation 3.1. *If $X \subseteq X' \subseteq V$, then $S(X) \subseteq S(X')$.*

We will establish later that the collection $\mathcal{X} = \{X_v : v \in V\}$ is laminar (see Lemma 3.2). In steps 2-3, the algorithm contracts each maximal set in \mathcal{X} into a single node and builds a graph D' on these nodes and r . For each contracted set $X \in \mathcal{X}$, edges leading to node X in D' correspond to undominated edges in D leading from outside X to the *candidate entry-points* of X , which are nodes $v \in X$ such that $X = X_v$. Theorems 3.3-3.4 show that D admits a popular arborescence if and only if D' admits an arborescence.

Our algorithm for computing a popular arborescence in D is given below. An example of the execution of the algorithm is shown in Figure 1.

1. For each $v \in V$ do:
 - let $X_v^0 = V$ and $i = 0$;
 - while v does not reach all nodes in the graph $D_v^i = (X_v^i, S(X_v^i))$ do:
 - X_v^{i+1} = the set of nodes reachable from v in D_v^i ; let $i = i + 1$.
 - let $X_v = X_v^i$.
2. Let $\mathcal{X} = \{X_v : v \in V\}$, $\mathcal{X}' = \{X_v \in \mathcal{X} : X_v \text{ is } \subseteq\text{-maximal in } \mathcal{X}\}$, $E' = \emptyset$.
3. For every edge $e = (u, v)$ in D such that $X_v \in \mathcal{X}'$ and $u \notin X_v$ do:
 - if e is undominated (i.e., $e \not\prec_v e'$) among all edges $e' \in \delta^-(X_v)$, then

$$f(e) = \begin{cases} (U, X_v) & \text{where } u \in U \text{ and } U \in \mathcal{X}', \\ (r, X_v) & \text{if } u = r; \end{cases}$$

- let $E' := E' \uplus \{f(e)\}$.

4. If $D' = (\mathcal{X}' \cup \{r\}, E')$ contains an arborescence A' , then
 - let $A'_D = \{e : f(e) \in A'\}$;
 - let $R = \{v \in V : v \text{ has an incoming edge in } A'_D\}$;
 - for each $v \in R$: let A_v be an arborescence in $(X_v, S(X_v))$ rooted at v ;
 - return $A^* = A'_D \cup_{v \in R} A_v$.
 Else return “No popular arborescence in D ”.

Correctness of the above algorithm. We first establish the laminarity of the sets X_u for $u \in V$ constructed in step 1 of the algorithm.

Lemma 3.2. $\mathcal{X} = \{X_v : v \in V\}$ is laminar. For any $v \in V$, if $u \in X_v$ then $X_u \subseteq X_v$.

Proof. We start by proving the second statement. Let $v \in V$ and $u \in X_v$. We first show that $X_u^i \subseteq X_v^i$ for any i , where we set $X_v^i := X_v$ whenever X_v^i is not defined by the above algorithm. The claim clearly holds for $i = 0$. Let i be the smallest index such that $x \in X_u^i \setminus X_v^i$ for some node x ; we must have $x \in X_u^{i-1} \cap X_v^{i-1}$. By the definition of X_u^i , x is reachable from u in $S(X_u^{i-1})$. Note that $X_u^{i-1} \subseteq X_v^{i-1}$ implies $S(X_u^{i-1}) \subseteq S(X_v^{i-1})$, which yields that x is reachable from u in $S(X_v^{i-1})$ as well. Moreover, u is reachable from v in $S(X_v^{i-1}) \supseteq S(X_v)$ because $u \in X_v$ and $S(\cdot)$ is monotone. Hence it follows that x is reachable from v in $S(X_v^{i-1})$ via u , contradicting the assumption that $x \notin X_v^i$. This concludes the proof of the second statement.

Now we will show the laminarity of \mathcal{X} . For contradiction, assume there exist $s, t \in V$ such that X_s and X_t cross, i.e., their intersection is non-empty and neither contains the other. Then, by the second statement of the lemma, neither $s \in X_t$ nor $t \in X_s$ can hold. So we have that $s \notin X_t$ and $t \notin X_s$.

Let (x, y) be an edge in $S(X_t)$ such that $y \in X_s \cap X_t$ but $x \in X_t \setminus X_s$; since each node in X_t is reachable from t in $S(X_t)$, such an edge exists. Since $y \in X_s \setminus \{s\}$, there also exists an edge (u, y) in $S(X_s)$. As $x \notin X_s$ but $(u, y) \in S(X_s)$, we know that $(u, y) \succ_y (x, y)$ which contradicts $(x, y) \in S(X_t)$. \square

Theorem 3.3. If the above algorithm returns an edge set A^* , then A^* is a popular arborescence in D .

Proof. We start by showing that A^* is an arborescence in D . Laminarity of \mathcal{X} implies that the sets in \mathcal{X}' are pairwise disjoint. Moreover, by construction, each node in V is included in at least one set in \mathcal{X} , namely $v \in X_v$ for each $v \in V$. Hence, \mathcal{X}' forms a partition of V . Thus each $w \in V$ has exactly one incoming edge in A^* : an edge from A'_D if $w \in R$ or an edge from the arborescence A_v for the unique node $v \in R$ with $w \in X_v$ if $w \in V \setminus R$. Now assume by contradiction that A^* contains a cycle C . Because the arborescences A_v for $v \in R$ are node-disjoint, this cycle must contain edges from A'_D and the set $\{f(e) : e \in C \cap A'_D\} \subseteq A'$ must contain a cycle, a contradiction. Thus A^* is an arborescence.

We now establish that A^* is popular by showing that $\mathcal{Y} := \{X_v : v \in R\} \cup \{\{v\} : v \in V \setminus R\}$ is a dual certificate for A^* of size n . Note that $|\mathcal{Y}| =$

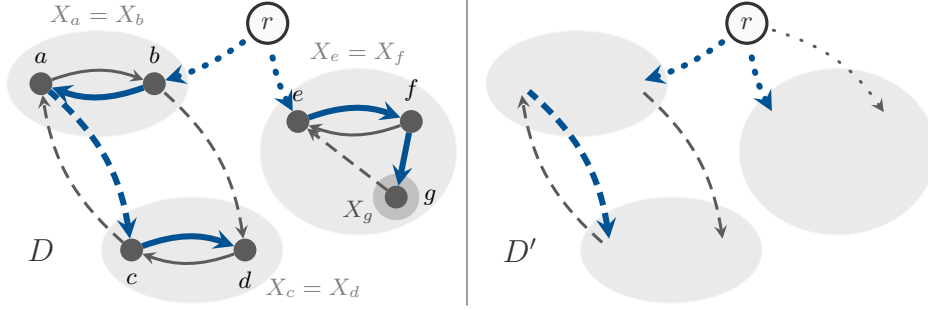


Fig. 1. The left figure shows an example for the graph D , i.e., the input instance extended by a root node r . Some outgoing arcs from the root are omitted in D . Solid arcs are preferred over dashed arcs which are preferred over dotted arcs. Grey areas correspond to sets in \mathcal{X} . The right figure depicts the corresponding graph D' constructed by our algorithm. Any arborescence in D' can be extended to a popular arborescence in D . E.g., the arborescence indicated by bold blue arcs is popular.

$|R| + |V \setminus R| = n$. We now show that for all $v \in V$, the incoming edges satisfy the constraints in [LP2](#).

Suppose $v \in R$. An edge $(w, v) \in E$ enters one set of \mathcal{Y} iff $w \notin X_v$ and no set iff $w \in X_v$. Hence, it suffices to show that $c_{A^*}((w, v)) \in \{1, 2\}$ for $w \notin X_v$. Let (u, v) be the incoming edge of v in arborescence A^* ; note that $(u, v) \in A'_D$ and $u \notin X_v$. By construction of E' , (w, v) does not dominate (u, v) and therefore $c_{A^*}((w, v)) \in \{1, 2\}$.

Suppose $v \in V \setminus R$. Let $s \in R$ be the unique node in R with $v \in X_s$. Then there is $(u, v) \in A_s \subseteq S(X_s)$ by construction of A_s . Any edge $(w, v) \in \delta^-(v)$ enters at most two sets of \mathcal{Y} : $\{v\}$ and possibly X_s . If, on the one hand, $(w, v) \in \delta^-(X_s)$, then $(u, v) \in S(X_s)$ dominates (w, v) by property 2. of $S(X_s)$, and hence $c_{A^*}((w, v)) = 2$. If, on the other hand, $w \in X_s$, then $(u, v) \in S(X_s)$ is not dominated by (w, v) by property 1. of $S(X_s)$, and hence $c_{A^*}((w, v)) \geq 1$. This proves that \mathcal{Y} is a dual certificate of size n for A^* , thus A^* is popular. \square

Theorem 3.4. *If D has a popular arborescence, then our algorithm finds one.*

Before we prove [Theorem 3.4](#), we need [Lemma 3.5](#) and [Lemma 3.6](#).

Lemma 3.5. *Let A be a popular arborescence and $\mathcal{Y} = \{Y_v : v \in V\}$ a dual certificate for A of size n . Then $Y_v \subseteq X_v$ for any $v \in V$.*

Proof. Assume for contradiction that $Y_v \setminus X_v \neq \emptyset$. Because $Y_v \subseteq V = X_v^0$, there is an iteration i with $Y_v \subseteq X_v^i$ and $Y_v \not\subseteq X_v^{i+1}$. Let $z \in Y_v \setminus X_v^{i+1}$. By [Lemma 2.4](#), A enters Y_v only at v and thus A contains a directed v - z -path using only nodes of Y_v . Because $z \notin X_v^{i+1}$, there must be an edge $(u, w) \in A$ on this path with $u \in X_v^{i+1}$ and $w \in Y_v \setminus X_v^{i+1}$. Note that $\delta^+(X_v^{i+1}) \cap S(X_v^i) = \emptyset$ by construction of X_v^{i+1} from X_v^i . Thus $(u, w) \notin S(X_v^i)$ and one of the following two cases must be true:

Case 1: There exists an edge $(x, w) \in E[X_v^i]$ that dominates (u, w) and hence $c_A((x, w)) = 0$. However, $\{w\} \in \mathcal{Y}$ by Corollary 2.6. So \mathcal{Y} violates the dual constraint for (x, w) , a contradiction.

Case 2: There is an edge $(x, w) \in \delta^-(X_v^i)$ which is not dominated by (u, w) and hence $c_A((x, w)) \in \{0, 1\}$. But since $x \notin X_v^i \supseteq Y_v$ the edge (x, w) enters both the set $\{w\}$ and the set Y_v in \mathcal{Y} , again a contradiction. \square

Lemma 3.6. *Let A be a popular arborescence in D and let $X \in \mathcal{X}'$. Then A enters X exactly once, and it enters X at some node v such that $X = X_v$.*

Proof. Let $X \in \mathcal{X}'$ and let A be a popular arborescence which enters X at some node $v \in V$ through an edge $(u, v) \in A \cap \delta^-(X)$. Moreover, let \mathcal{Y} be a dual certificate for A , and let Y_v be the set whose entry-point is v .

Let $\text{entry}(X) := \{w \in V : X_w = X\}$. We first show that $\text{entry}(X) \subseteq Y_v$. Assume for contradiction that there exists $w \in \text{entry}(X)$ such that $w \notin Y_v$. Since $X_w = X$, we know that there exists a w - v -path P in $(X, S(X))$. Hence, there exists an edge $e = (u', v') \in P$ which enters Y_v . If $v' = v$, we know that $e \in S(X)$ dominates $(u, v) \in \delta^-(X)$ and hence $c_A(e) = 0$, a contradiction to the feasibility of \mathcal{Y} . If $v' \neq v$, then e enters not only $Y_v \in \mathcal{Y}$, but also $\{v'\} \in \mathcal{Y}$. However, $c_A(e) \leq 1$ since $e \in S(X)$ must be an undominated edge, a contradiction to the feasibility of \mathcal{Y} .

We next show that $v \in \text{entry}(X)$. Let $s \in \text{entry}(X)$. By the previous paragraph and Lemma 3.5, we get $s \in Y_v \subseteq X_v$, from which Lemma 3.2 implies $X_s \subseteq X_v$. Because $s \in \text{entry}(X)$, we have $X = X_s \subseteq X_v$. Because $X \in \mathcal{X}'$ is inclusionwise maximal in \mathcal{X} , we get $X = X_v$, proving $v \in \text{entry}(X)$.

It remains to prove that A enters X only once. Suppose for contradiction that there exist two nodes $v, v' \in \text{entry}(X)$ such that $(u, v), (u', v') \in A \cap \delta^-(X)$. By $\emptyset \neq \text{entry}(X) \subseteq Y_v \cap Y_{v'}$ and the laminarity of \mathcal{Y} , we can assume w.l.o.g. that $Y_v \subseteq Y_{v'}$. Moreover, since $u \notin X \supseteq Y_{v'}$, the edge $(u, v) \in A$ enters both Y_v and $Y_{v'}$, contradicting the feasibility of the dual solution \mathcal{Y} . \square

Proof (of Theorem 3.4). Assume there exists a popular arborescence A in D ; then there exists a dual certificate $\mathcal{Y} = \{Y_v : v \in V\}$ of size n for A . We will show that there exists an arborescence in D' . By Lemma 3.6, for each $X \in \mathcal{X}'$ there exists exactly one edge $e_X = (u, v)$ of A that enters $X = X_v$.

We claim that (u, v) is not dominated by any $(u', v) \in \delta^-(X)$. Recall that by Lemma 3.5, we know $Y_v \subseteq X_v = X$. If some $(u', v) \in \delta^-(X)$ dominates $(u, v) \in A$, its cost must be $c_A((u', v)) = 0$. However, (u', v) clearly enters $Y_v \subseteq X$, and \mathcal{Y} violates the dual constraint for (u', v) , a contradiction. Hence, e_X is undominated among the edges of $\delta^-(X) \cap \delta^-(v)$ and so our algorithm creates an edge $f(e_X)$ in E' pointing to X . Using that A is an arborescence in D , it is straightforward to verify that the edges $\{f(e_X) : X \in \mathcal{X}'\}$ form an arborescence A' in D' . Thus our algorithm returns an edge set A^* , which by Theorem 3.3 must be a popular arborescence in D . \square

Observe that step 1, the bottleneck step, takes $O(mn)$ time per node. An arborescence in D' can be found by breadth first search, i.e., in time $O(n' + m')$

where n' and m' are the number of vertices and edges in D' , respectively. Thus the running time of the algorithm is $O(mn^2)$, and Theorem 1.2 follows.

4 Branchings with Minimum Unpopularity Margin

Recall the definition of the *unpopularity margin* for branchings from Section 1. Again, instead of studying minimum unpopularity margin branchings within the digraph G , we look at r -arborescences of minimum unpopularity margin within the digraph D as constructed in Section 2. It is easy to see that the unpopularity margin of a branching in G is the same as the unpopularity margin of the corresponding arborescence in D .¹⁰ Thus we are looking for an arborescence of minimum unpopularity margin in D . We further obtain the following variant of Lemma 2.4 which proves useful in bounding the popularity margin of a given arborescence in D .

Lemma 4.1. *Let A be an arborescence in D and \mathcal{Y} be a dual certificate for A . Then $\mu(A) \leq n - |\mathcal{Y}|$, with equality if \mathcal{Y} has maximum cardinality.*

Proof. Let A' be a min-cost arborescence in D with respect to c_A . Recalling the pair of integral LPs LP1 and LP2 from Section 2, we observe that A' corresponds to an optimal solution to LP1 of value $c_A(A')$ and \mathcal{Y} corresponds to a feasible solution to LP2 of value $|\mathcal{Y}|$. We conclude that $c_A(A') \geq |\mathcal{Y}|$ by LP duality. Thus, Proposition 2.1 implies that $\mu(A) = n - c_A(A') \leq n - |\mathcal{Y}|$. If \mathcal{Y} has maximum cardinality, it is optimal to LP2 and the inequality becomes an equality by strong duality. \square

4.1 A Simple Extension of our Algorithm: Algorithm MINMARGIN

Our algorithm from Section 3 can be extended to compute an arborescence with minimum *unpopularity margin* when nodes have weak rankings. If D' does not admit an arborescence, algorithm MINMARGIN below computes a maximum-size branching B' in D' and adds edges from r to all root nodes in B' to construct an arborescence. This arborescence in D' is then transformed into an arborescence in D . Figure 2 shows an example execution of the algorithm.

1. Let D' be the graph constructed in the algorithm in Section 3, and let B' be a branching of maximum size in D' .
2. Let $B'_D = \{e : f(e) \in B'\}$, $R_1 = \{v \in V : \delta^-(v) \cap B'_D \neq \emptyset\}$, $R_2 = \emptyset$.
3. For each $X \in \mathcal{X}'$ which is a root in the branching B' , select one arbitrary $v \in V$ with $X_v = X$. Add v to R_2 and add (r, v) to B'_D .
4. For each $v \in R_1 \cup R_2$, let A_v be an arborescence in $(X_v, S(X_v))$ rooted at v .
5. Return $A^* := B'_D \cup_{v \in R_1 \cup R_2} A_v$.

¹⁰ Note that, due to the special structure of D , there always exists an arborescence A' such that $A' \in \arg \max_{B \in \mathcal{B}(D)} \phi(B, A) - \phi(A, B)$, where $\mathcal{B}(D)$ is the set of branchings in D .

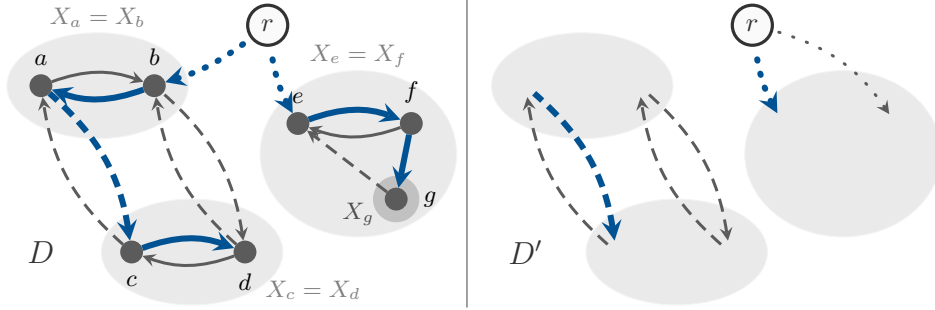


Fig. 2. Variant of the example introduced in Figure 1 with one additional incoming arc for node b , i.e., the arc (d, b) . As a consequence, D' does not contain an arborescence. On the right, the bold blue edges form a maximum-size branching in D' , which induces an arborescence of minimum unpopularity margin within the graph D . Again, in the left figure, the grey areas correspond to the sets in \mathcal{X} and some outgoing arcs of r are omitted.

Theorem 4.2. *When nodes have weak rankings, Algorithm MINMARGIN returns an arborescence with minimum unpopularity margin in D .*

To prove Theorem 4.2, we first show in Lemma 4.3 that the size of the set R_2 is an upper bound on the unpopularity margin of the arborescence A^* returned by Algorithm MINMARGIN. Then we introduce the concept of completeness of dual certificates, which can be used to show that any arborescence in D has unpopularity margin at least $|R_2|$. We remark that only the latter bound requires the properties of weak rankings, whereas the algorithm itself and Lemma 4.3 hold for arbitrary partial-order preferences.

Lemma 4.3. *Let A^*, R_2 be as computed by the algorithm. Then $\mu(A^*) \leq |R_2|$.*

Proof. By Lemma 4.1 it suffices to construct a dual certificate for A^* of size $n - |R_2|$. Define $\mathcal{Y} := \{X_v : v \in R_1\} \cup \{\{v\} : v \in V \setminus \{R_1 \cup R_2\}\}$ and note that \mathcal{Y} is a laminar family of size $n - |R_2|$. It remains to show that \mathcal{Y} fulfills the constraints of Definition 2.3. To this end, consider any edge $e = (w, v) \in E$ and let (u, v) be the unique incoming edge of v in A^* .

Suppose $v \in R_2$. Then v is not contained in any set of \mathcal{Y} , so the constraint for $e = (w, v)$ is not violated.

Suppose $v \in R_1$. Then $(u, v) \in B'_D$ and $u \notin X_v$. Edge $e = (w, v)$ enters exactly one set of \mathcal{Y} if $w \notin X_v$ and no set if $w \in X_v$. Hence, it suffices to show that $c_{A^*}(e) \in \{1, 2\}$ for $w \notin X_v$. Note that $(u, v) \in B'_D$ corresponds to an edge in D' and hence e with $w \notin X_v$ does not dominate (u, v) by construction of D' . Thus $c_{A^*}(e) \in \{1, 2\}$ in this case.

Suppose $v \in V \setminus (R_1 \cup R_2)$. Let $s \in R_1 \cup R_2$ be the unique node in $R_1 \cup R_2$ with $v \in X_s$. Then $(u, v) \in A_s$ by construction of A^* . Edge $e = (w, v)$ enters at most two sets of \mathcal{Y} if $w \notin X_s$ and exactly one set if $w \in X_s$. If $w \notin X_s$, then by construction of A_s and property 2 of $S(X_s)$, it holds that (w, v) is dominated

by (u, v) , and hence $c_{A^*}(e) = 2$. If $w \in X_s$, then by construction of A_s and property 1 of $S(X_s)$, (w, v) does not dominate (u, v) , and hence $c_{A^*}(e) \in \{1, 2\}$. Thus, \mathcal{Y} is a dual certificate for A^* of size $n - |R_2|$. \square

Lemma 4.4. *Let A be an arborescence in D . For $Z \subseteq V$ and $v \in Z$, let $\mathcal{Y}_{Z,v} := \{\{w\} : w \in Z \setminus \{v\}\} \cup \{Z\}$. If $|\{Y \in \mathcal{Y}_{Z,v} : e \in \delta^-(Y)\}| \leq c_A(e)$ for all $e = (w, x) \in E$ with $x \in Z \setminus \{v\}$, then $Z \subseteq X_v$.*

Proof. The proof is a direct analog of the proof of Lemma 3.5, with Z taking the role of Y_v and noting that this proof only makes use of dual constraints for nodes $w \in Y_v \setminus \{v\}$. \square

For Observation 4.5 recall that the *weakly connected components* of a directed graph are the connected components of its induced undirected graph.

Observation 4.5. *Let B' be a branching of maximum size in D' and $C \subseteq \mathcal{X}'$ be a weakly connected component of B' not containing r . Then, there exists $C' \subseteq C$ such that $\delta_{D'}^-(C') = \emptyset$.*

Proof. Assume for contradiction that $\delta_{D'}^-(C') \neq \emptyset$ for all $C' \subseteq C$. Hence, every $X \in C$ is reachable from $\{r\} \cup \mathcal{X}' \setminus C$ in D' . Consequently, we can modify B' by attaching each $X \in C$ to one of the other connected components of B' , increasing its size by 1. This contradicts the maximality of B' . \square

Completeness of dual certificates. Let A be an arborescence in D and \mathcal{Y} a dual certificate for A . If an edge $(u, v) \in A$ enters a set $Y \in \mathcal{Y}$ we say that Y *belongs* to v . Note that every set of \mathcal{Y} belongs to some node and, since $c_A(e) = 1$ for every $e \in A$, at most one set may belong to a node. However, if $|\mathcal{Y}| < n$, the same set of \mathcal{Y} may belong to multiple distinct nodes and there may be nodes to which no set in \mathcal{Y} belongs. We say that \mathcal{Y} is *complete on* $S \subseteq V$, if $|\{Y \in \mathcal{Y} : Y \text{ belongs to some } v \in S\}| = |S|$. Note that if \mathcal{Y} is complete on S , then for each $v \in S$ there is a unique set $Y_v \in \mathcal{Y}$ belonging to v , and for any two distinct nodes $v, v' \in S$ we have $Y_v \neq Y_{v'}$. In the following proof of Theorem 4.2 we will show that there exist $|R_2|$ disjoint node sets for which no dual certificate of any arborescence can be complete.

Proof (of Theorem 4.2). Let A be an arborescence with minimum unpopularity margin and \mathcal{Y} a dual certificate for A with maximum cardinality. We will show that $|\mathcal{Y}| \leq n - |R_2|$. By Lemmas 4.1 and 4.3 this implies $\mu(A) \geq |R_2| \geq \mu(A^*)$, which proves the theorem.

Note that there is a one-to-one correspondence between nodes in R_2 and the connected components of B' in D' not containing r . Let C be such a connected component. By Observation 4.5, there exists some $C' \subseteq C$ with $\delta_{D'}^-(C') = \emptyset$. We will show that \mathcal{Y} is not complete on $S := \bigcup_{X \in C'} X$. As the sets S are disjoint for distinct connected components C , there exist $|R_2|$ pairwise disjoint sets on which \mathcal{Y} is not complete. Since each set of \mathcal{Y} belongs to some node of V , and at most one set belongs to each node, this implies $|\mathcal{Y}| \leq n - |R_2|$.

It remains to show that \mathcal{Y} is not complete on S . Assume for contradiction that \mathcal{Y} is complete on S and for each $v \in S$ let Y_v be the unique set belonging to v . Consider any edge $e = (u, v) \in A \cap \delta_D^-(S)$ at which A enters S . We will show that $Y_v \cap S$ and v fulfill the requirements of Lemma 4.4, but that $Y_v \cap S \not\subseteq X_v$, deriving a contradiction. Thus \mathcal{Y} cannot be complete on S .

In the proof of the first claim, we rely on our assumption that the preferences are given by a weak ranking.

Claim. Let $Z := Y_v \cap S$ and $\mathcal{Y}_{Z,v} := \{\{w\} : w \in Z \setminus \{v\}\} \cup \{Z\}$. Then $|\{Y \in \mathcal{Y}_{Z,v} : e' \in \delta^-(Y)\}| \leq c_A(e')$ for all $e' = (w, x) \in E$ with $x \in Z \setminus \{v\}$.

Proof. Let $e' = (w, x) \in E$ with $x \in Z \setminus \{v\}$. By the laminarity of \mathcal{Y} and its completeness on S , we know that $Y_x \cap S = \{x\}$. Indeed, assuming that x' is a node in $Y_x \cap S$ other than x , the sets $Y_{x'}$, Y_x , and Y_v would necessarily contradict Observation 2.5. We distinguish two cases:

Case 1: $w \notin Y_v \setminus S$. In this case e' enters Y_v if and only if it enters $Z = Y_v \cap S$. Hence $|\{Y \in \mathcal{Y}_{Z,v} : e' \in \delta^-(Y)\}| = |\{Y \in \mathcal{Y} : e' \in \delta^-(Y)\}| \leq c_A(e')$.

Case 2: $w \in Y_v \setminus S$. In that case e' enters the two sets $Y_v \cap S$ and $\{x\}$. We need to prove $c_A(e') = 2$. Consider the set $X \in \mathcal{C}'$ that contains x . Note that $\delta_{D'}^-(C') = \emptyset$ implies that e' must be dominated by some edge f entering X at x . Without loss of generality we can choose f such that it is undominated among all edges in $\delta_D^-(X) \cap \delta_D^-(x)$. Clearly, f enters the set $\{x\} \in \mathcal{Y}_{Z,v}$. Note that f cannot enter S , because $\delta_{D'}^-(C') = \emptyset$ and f is not dominated by edges entering $X \subseteq S$. So we can apply the statement of the previous case to f , yielding $1 \leq |\{Y \in \mathcal{Y}_{Z,v} : f \in \delta^-(Y)\}| \leq c_A(f)$. This implies that f does not dominate $A(x)$. So either $A(x) \succ_x f \succ_x e'$ or $A(x) \sim_x f$, which also implies $A(x) \succ_x e'$ because preferences are weak rankings. Hence $c_A(e') = 2$. ■

Claim. $Y_v \cap S \not\subseteq X_v$.

Proof. Let $X \in \mathcal{X}'$ be the unique set of \mathcal{X}' with $v \in X$.

Case 1: $X_v \neq X$. Let $s \in V$ with $X_s = X$. Note that $v \in X$ implies $s \in X \subseteq S$. By construction of X_s there exists an s - v -path $P \subseteq S(X)$ containing only safe edges for X_s . Recall that a safe edge cannot be dominated by any other edge. If $s \notin Y_v$, then there is an edge $(u', v') \in P$ entering Y_v . If $v' \neq v$, then the non-dominated edge (u', v') enters two sets of \mathcal{Y} , a contradiction. If $v' = v$, then (u', v') dominates $(u, v) \in A$ but enters Y_v , again a contradiction. We conclude that $s \in Y_v$. However, as $X_v \neq X_s$, we know that $s \notin X_v$ by Lemma 3.2. We obtain $s \in (Y_v \cap S) \setminus X_v$.

Case 2: $X_v = X$. Because $f(e) \notin \delta_{D'}^-(C') = \emptyset$, there must be an edge $e' = (u', v)$ entering X_v at v and dominating e . Without loss of generality, we can choose e' such that it is undominated among all edges in $\delta_D^-(v) \cap \delta_D^-(X_v)$. Because $e \in A$, we obtain $c_A(e') = 0$ and hence e' cannot enter $Y_v \in \mathcal{Y}$. We conclude that $u' \in (Y_v \cap S) \setminus X_v$. ■

This completes the proof of Theorem 4.2. □

4.2 Unpopularity Margin under Partial Preference Orders

The following theorem shows that Algorithm MINMARGIN cannot be extended for the case where each node v has an arbitrary partial order over $\delta^-(v)$.

Theorem 4.6. *Given a directed graph where each node has a partial preference order over its incoming edges and an integer $k \leq n$, it is NP-hard to decide if there exists a branching with unpopularity margin at most k .*

Proof. We reduce from the NP-complete problem 3D-MATCHING where we are given disjoint sets X, Y, Z of equal cardinality and $T \subseteq X \times Y \times Z$, and we ask if there exists $M \subseteq T$ with $|M| = |X|$ such that for distinct $(x, y, z), (x', y', z') \in M$ it holds that $x \neq x', y \neq y'$ and $z \neq z'$; such an M is called a 3D-matching. W.l.o.g. we assume that $|X| > 3$ and every $x \in X \cup Y \cup Z$ is in some $t \in T$. For any $t = (x, y, z) \in T$ we define $S(t) := \{x, y, z\}$.

We construct a digraph $D = (V \cup \{r\}, E)$ together with a partial order \succ_v over the incoming edges of v for each $v \in V$ as follows. For every $x \in X \cup Y \cup Z$ we introduce a *node gadget* consisting of a lower node x_l and an upper node x_u . There exist two parallel edges, $d_x^{(1)}$ and $d_x^{(2)}$, from x_u to x_l , and there exist two parallel edges, $r_x^{(1)}$ and $r_x^{(2)}$, from r to x_l . Moreover, the upper node x_u has an incoming edge from the upper node of every other node gadget, i.e., $(x'_u, x_u) \in E$ for all $x' \in X \cup Y \cup Z \setminus \{x\}$. Lastly, there exists an incoming edge from r to the upper node which we call $r_x^{(3)}$.

For each $t \in T$ we introduce a *hyperedge gadget* consisting of six edges in D . More precisely, for each $x \in S(t)$ we introduce two parallel edges from x_l to x_u which we call $t_x^{(1)}$ and $t_x^{(2)}$. This finishes the definition of D .

Let us now define the preferences $\{\succ_v: v \in V\}$. A lower node x_l has the following preferences over its incoming edges:

$$d_x^{(1)} \succ r_x^{(1)}, \quad d_x^{(2)} \succ r_x^{(2)},$$

and all other pairs are not comparable. The preferences of an upper node x_u are as follows:

$$\begin{aligned} (x'_u, x_u) &\succ r_x^{(3)} && \text{for each } x' \in X \cup Y \cup Z \setminus \{x\}, \\ t_x^{(1)} &\succ (x'_u, x_u) && \text{for each } t \in T \text{ with } x \in S(t) \text{ and each } x' \in X \cup Y \cup Z \setminus S(t), \\ t_x^{(2)} &\succ (x'_u, x_u) && \text{for each } t \in T \text{ with } x \in S(t) \text{ and each } x' \in S(t) \setminus \{x\}, \\ t_x^{(1)} &\succ r_x^{(3)}, && \text{for each } t \in T \text{ with } x \in S(t), \\ t_x^{(2)} &\succ r_x^{(3)}, && \text{for each } t \in T \text{ with } x \in S(t), \end{aligned}$$

and all other pairs are not comparable. See Figure 3 for an illustration.

Note that the digraph D has the special property that every node $v \in V$ has at least one incoming edge from r . As a consequence, any branching B in D minimizing $\mu(B)$ must in fact be an arborescence rooted at r . Moreover, we can apply Lemma 4.1 to any given arborescence A in D as usual. In the following we show that there exists a 3D-matching $M \subseteq T$ with $|M| = |X|$ iff there exists an r -arborescence in D with unpopularity margin at most $2|X|$.

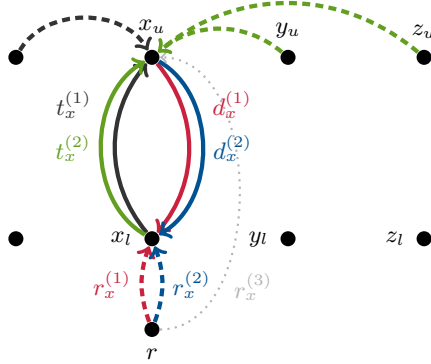


Fig. 3. Construction in the reduction for Theorem 4.6 depicting the node gadget for $x \in X$ and the hyperedge gadget for $(x, y, z) \in T$. Each solid edge dominates dashed edges of the same color; edge $r_x^{(3)}$ is dominated by all other edges; all other edge pairs are incomparable.

First, let $M \subseteq T$ be a 3D-matching with $|M| = |X|$. We construct an arborescence A together with a feasible dual certificate \mathcal{Y} with $|\mathcal{Y}| = 4|X|$. By Lemma 4.1, this suffices to show that A has unpopularity margin at most $6|X| - 4|X| = 2|X|$. We define A and \mathcal{Y} as

$$A := \{r_x^{(1)} : x \in X \cup Y \cup Z\} \cup \{t_w^{(1)} : t \in M, w \in S(t)\},$$

$$\mathcal{Y} := \{\{x_u\} : x \in X \cup Y \cup Z\} \cup \{\{x_u, y_u, z_u, x_l, y_l, z_l\} : (x, y, z) \in M\}.$$

It is easy to verify that A is indeed an r -arborescence. It remains to show that \mathcal{Y} is a feasible dual solution.

First consider a node x_l for $x \in X \cup Y \cup Z$ which has four incoming edges. The edges $d_x^{(1)}$ and $d_x^{(2)}$ do not enter any set in \mathcal{Y} and hence do not violate any constraint in the dual LP. Moreover, since node x_l is indifferent between $r_x^{(1)}$ and $r_x^{(2)}$, we obtain $c_A(r_x^{(1)}) = c_A(r_x^{(2)}) = 1$ and hence, none of the constraints corresponding to an edge entering x_l is violated.

Now consider x_u for $x \in X \cup Y \cup Z$. Let $S(t) = \{x, y, z\}$ for the hyperedge t in M containing x . We obtain $c_A((y_u, x_u)) = c_A((z_u, x_u)) = c_A(e) = 1$ for any edge e pointing from x_l to x_u , while for any other incoming edge e of x_u we get $c_A(e) = 2$. By construction of \mathcal{Y} , the first group of edges enter only the set $\{x_u\}$ of \mathcal{Y} , while the second group of edges enter two sets of \mathcal{Y} . Thus none of the constraints is violated. This shows the first direction of the equivalence.

For the reverse direction, let A be an r -arborescence of unpopularity margin at most $2|X|$. Let \mathcal{Y} be a corresponding laminar certificate of size $|\mathcal{Y}| = 4|X|$.

Since the number of sets in \mathcal{Y} is greater than the number of node gadgets, there exist node gadgets which intersect with more than one set from \mathcal{Y} . Let $x \in X \cup Y \cup Z$ be a node such that the corresponding node gadget intersects

with at least two sets $Y_1, Y_2 \in \mathcal{Y}$. We will show that $Y_2 \subseteq Y_1$, $\{x_u, x_l\} \subseteq Y_1$, $x_u \in Y_2$, and $x_l \notin Y_2$ (w.l.o.g. after possibly swapping Y_1 and Y_2).

First, assume for contradiction that $Y_1 \cap Y_2 = \emptyset$. Then, $\{r_x^{(1)}, r_x^{(2)}\} \cap A = \emptyset$ since otherwise $c_A(d_x^{(1)}) = 0$ or $c_A(d_x^{(2)}) = 0$, however, both of them enter a set from \mathcal{Y} . This implies that $\{t_x^{(1)}, t_x^{(2)}\} \cap A = \emptyset$ for all $t \in T$ such that $x \in S(t)$ since otherwise A would contain a cycle. Therefore, no matter which other incoming edge of x_u is used by A , there exists an edge $e \in E$ from x_l to x_u such that $c_A(e) = 0$. But this contradicts $Y_1 \cap Y_2 = \emptyset$, as e cannot enter any set in \mathcal{Y} . We conclude that Y_1 and Y_2 intersect and since \mathcal{Y} is laminar we can assume w.l.o.g. that $Y_2 \subseteq Y_1$. Second, assume for contradiction that $x_l \notin Y_1$. Observe that, no matter which edge points towards x_u within A , both $t_x^{(1)}$ or $t_x^{(2)}$ have a cost of at most 1 in c_A , for any triple $t \in T$ containing x . However, both of these edges enter two sets from \mathcal{Y} , a contradiction. An analogous argument shows $x_u \in Y_1$. Finally note that x_l can be contained in at most one set of \mathcal{Y} , as no arc dominates both $r_x^{(1)}$ and $r_x^{(2)}$ at the same time and therefore $\min\{c_A(r_x^{(1)}), c_A(r_x^{(2)})\} \leq 1$. Thus $x_l \notin Y_2$. We conclude that $Y_2 \subseteq Y_1$, $\{x_u, x_l\} \subseteq Y_1$, $Y_2 \cap \{x_u, x_l\} = \{x_u\}$. Note that we have proved these properties for any two sets in \mathcal{Y} that intersect $\{x_u, x_l\}$; however there cannot exist three sets in \mathcal{Y} such that any two of them fulfill these conditions (i.e., that one of them contains both x_u and x_l , the other contains only x_u but not x_l). Therefore, this also suffices to argue that there exist at most two sets from \mathcal{Y} intersecting with x 's node gadget.

Let $\mathcal{S} := \{Y_1 \in \mathcal{Y} : Y_1 \text{ is } \subseteq\text{-maximal and there exists } Y' \in \mathcal{Y}, Y' \subset Y_1\}$. As a consequence of the above observation, no node gadget can intersect with more than two sets of \mathcal{Y} and therefore $|\mathcal{Y} \setminus \mathcal{S}| \leq 3|X|$ implying $|\mathcal{S}| \geq |X|$. For every $Y_1 \in \mathcal{S}$ we select a representative $x(Y_1) \in X \cup Y \cup Z$ whose node gadget intersects with Y_1 and one other set Y_2 from \mathcal{Y} . Consider the node gadget of $x := x(Y_1)$. Recall that $x_u \in Y_1 \cap Y_2$ and $x_l \in Y_1 \setminus Y_2$ and that $A(x_u)$ must be an edge starting at x_l , as established in the preceding paragraph. We claim that $A(x_u) = t_x^{(1)}$ for some $t \in T$. Assume for contradiction that x_u is entered by $t_x^{(2)}$ for some $t \in T$. Note that $t_x^{(2)}$ is incomparable with the edges (y_u, x_u) for each $y \in X \cup Y \cup Z \setminus t$, and thus none of these edges can cross both Y_1 and Y_2 . As $Y_2 \subseteq Y_1$, we conclude that Y_1 contains x_u and all $3|X| - 3$ nodes y_u for $y \in X \cup Y \cup Z \setminus t$. As the sets in \mathcal{S} do not overlap, our assumption that $|X| > 3$ yields the contradiction $|X| \leq |\mathcal{S}| \leq 3 < |X|$. Thus, $A(x_u) = t_x^{(1)}$ for some $t = (x, y, z) \in T$. This implies $\{x_u, y_u, z_u\} \subseteq Y_1$, since $c_A((y_u, x_u)) = c_A((z_u, x_u)) = 1$. We conclude that neither y nor z is included in any other set of \mathcal{S} . Thus, $M := \{t \in T : t_{x(Y_1)}^{(1)} \in A, Y_1 \in \mathcal{S}\}$ is a 3D-matching of size $|X|$. \square

5 Branchings with Low Unpopularity Factor

Recall the definition of *unpopularity factor* from Section 1. As done in the previous section, instead of studying branchings in the digraph G , we look at r -arborescences within the digraph D . The unpopularity factor of any branching in G is the same as the unpopularity factor of the corresponding arborescence

in D . Given any arborescence A and value t , there is a simple method to verify if $u(A) \leq t$ or not. This is totally analogous to our method from Section 2 to verify popularity, and it involves computing a min-cost arborescence in D with the following edge costs. For $e = (u, v)$ in D , define:

$$c_A(e) := \begin{cases} 0 & \text{if } e \succ_v A(v), \\ 1 & \text{if } e \sim_v A(v), \\ t + 1 & \text{if } e \prec_v A(v). \end{cases}$$

Lemma 5.1. *Arborescence A satisfies $u(A) \leq t$ if and only if A is a min-cost arborescence in D with edge costs given by c_A defined above.*

Proof. For any arborescence A' , we have $c_A(A') = t \cdot \phi(A, A') - \phi(A', A) + n$. In particular, $c_A(A) = n$. Hence, A is a min-cost arborescence with respect to c_A iff $n \leq t \cdot \phi(A, A') - \phi(A', A) + n$ holds for all arborescences A' . This in turn holds iff $\frac{\phi(A', A)}{\phi(A, A')} \leq t$ for all A' which is equivalent to $u(A) \leq t$. \square

Lemma 5.2 follows from Lemma 5.1 and LP duality.

Lemma 5.2. *Arborescence A satisfies $u(A) \leq t$ if and only if there exists a dual feasible solution y to LP2 with $c_A(e)$ as defined above with $\sum_X y_X = n$.*

Proof of Theorem 1.5. We now assume that node preferences are *strict*; thus we may assume the graph to be simple, and nodes to have preferences over their in-neighbors. We modify our algorithm from Section 3 to compute an arborescence A in $D = (V \cup \{r\}, E)$ such that $u(A) \leq \lfloor \log n \rfloor$.

1. Initially all nodes in V are active. Set $X_v^0 = \{v\}$ for all $v \in V$.
2. Initialize the current edge set $E' = \emptyset$; let $i = 1$.
3. Let $E' = E' \cup \{(u, v) : v \in V \text{ is active and } u \text{ is } v\text{'s most preferred in-neighbor such that } u \notin X_v^{i-1}\}$.
4. For $v \in V$ let $X_v^i = \{w \in V : \exists v\text{-}w\text{-path in } E'\}$.
5. Let $\mathcal{X} = \{X_v^i \text{ is } \subseteq\text{-maximal in } \mathcal{X}^i\}$ where $\mathcal{X}^i = \{X_v^i : v \text{ is active}\}$.
6. For each $X \in \mathcal{X}$ do:
 - select any active node v such that $X_v^i = X$;
 - deactivate all $u \in X \setminus \{v\}$. *{now v is the only active node in X }*
 - if v is reachable from r using edges in E' , then deactivate v .
{this means all nodes in X are reachable from r }
7. If there exists any active node, then set $i = i + 1$ and go to step 3 above.
8. Compute an arborescence A in $(V \cup \{r\}, E')$ and return A .

When reaching step 8, there can be no active node. This means that every node is reachable from r using the edges in E' , and so there exists an arborescence A in $(V \cup \{r\}, E')$. We now bound its unpopularity factor.

Lemma 5.3. *\mathcal{X}^i is a laminar family for each iteration i .*

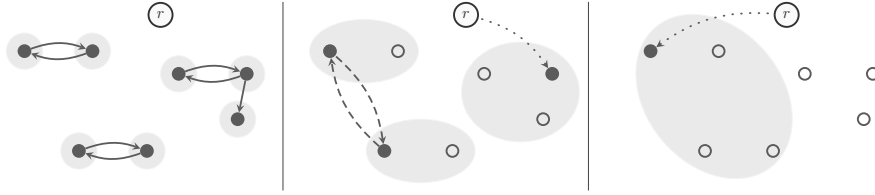


Fig. 4. The figure illustrates the execution of the algorithm from the proof of Theorem 1.5 for the example instance given in Figure 1. The three pictures illustrate the iterations $i = 1$, $i = 2$, and $i = 3$, respectively. Grey areas depict the sets in \mathcal{X}^{i-1} and only edges from the set E' as defined in step 3 are shown. Deactivated nodes are indicated by unfilled circles and active nodes by filled circles. The tie-breaking in step 6 was performed arbitrarily.

Proof. For $F \subseteq E$, let $X_v(F) := \{w \in V : \exists v\text{-}w\text{-path in } F\}$ and let $\mathcal{X}(F) := \{X_v(F) : v \in V\}$. Let e_1, \dots, e_ℓ be the edges in E' at the end of the algorithm in the order that they were added to E' (breaking ties arbitrarily among the edges added in the same iteration). Let $E_j := \{e_1, \dots, e_j\}$ and note that for every i there is a j such that $\mathcal{X}^i \subseteq \mathcal{X}(E_j)$. Since the subset of a laminar family is also laminar, it thus suffices to show that $\mathcal{X}(E_j)$ is laminar for every j , which we show by induction on j . This is trivial for $j = 0$ with $E_0 = \emptyset$. For some $j \geq 0$, let $e_{j+1} = (u, v)$. Note that, for $w \in V$, adding e_{j+1} to E_j results in $X_w(E_{j+1}) = X_w(E_j) \cup X_v(E_j)$ if $u \in X_w(E_j)$ and $X_w(E_{j+1}) = X_w(E_j)$ otherwise. Combining this with the fact that $X_v(E_j)$ is a \subseteq -maximal set in the (by the induction hypothesis) laminar family $\mathcal{X}(E_j)$, we conclude that $\mathcal{X}(E_{j+1})$ is also laminar. \square

Lemma 5.4. *If v, v' are active at the end of iteration i , then $X_v^i \cap X_{v'}^i = \emptyset$.*

Proof. Since v was not deactivated in iteration i , it means that all nodes in $X_v^i \setminus \{v\}$ were deactivated in that iteration. This implies that $v' \notin X_v^i$, and symmetrically, $v \notin X_{v'}^i$. Thus $X_v^i \cap X_{v'}^i = \emptyset$ by laminarity of \mathcal{X}^i . \square

Lemma 5.5. *If the while-loop terminates after $t + 1$ iterations, then $u(A) \leq t$.*

Proof. Let $\mathcal{Y}_i = \{X_v^{i-1} : v \in V \text{ and } v \text{ got deactivated in the } i\text{-th iteration}\}$ and let $\mathcal{Y} = \bigcup_i \mathcal{Y}_i$. For $X \subseteq V$, let $y_X = 1$ if $X \in \mathcal{Y}$ and $y_X = 0$ otherwise. For each node v , there is a corresponding set X_v^{i-1} in \mathcal{Y} , and by Lemma 5.4, the sets corresponding to distinct nodes are distinct. Hence, we have $\sum_{X \subseteq V} y_X = n$.

Consider any $v \in V$. Again by Lemma 5.4 for each i , there is at most one set in \mathcal{Y}_i containing v . Thus $\sum_{X: e \in \delta^-(X)} y_X \leq t + 1$ for any edge e pointing to v . Furthermore, the algorithm ensures that the edge $(u^*, v) \in A$ is the most preferred edge entering v among all edges with tail outside X_v^{i-1} . So every other edge $e = (u, v)$ with $u \notin X_v^{i-1}$ is ranked worse than $(u^*, v) \in A$ and thus $c_A(e) = t + 1$. This proves that y is a feasible dual solution for A , so $u(A) \leq t$. \square

Lemma 5.6. *The while-loop runs for at most $\lfloor \log n \rfloor + 1$ iterations.*

Proof. Every node v that is active at the start of some iteration either becomes reachable from r in this iteration or it joins a weakly connected component with at least one other active node. At the end of each iteration, there is at most one active node in each weakly connected component. So the number of active nodes is reduced by a factor of $1/2$ in each iteration. Thus the number of active nodes at the end of the i -th iteration of the while-loop is at most $n/2^i$. Hence the while-loop can run for at most $\lfloor \log n \rfloor + 1$ iterations. \square

Combining Lemmas 5.6 and 5.5, the first part of Theorem 1.5 follows.

A tight example. We now describe an instance on n nodes with strict preferences where every branching has unpopularity factor at least $\lfloor \log n \rfloor$. For convenience, let $n = 2^k$ for some integer k . We construct an instance G_k on vertex set $V = \{v_0, \dots, v_{n-1}\}$, where every node has in-degree $k = \log n$. The instance on 4 vertices a, b, c, d given in Section 1 is, in fact, G_2 .

- For $0 \leq i \leq n/2 - 1$, the nodes v_{2i} and v_{2i+1} are each other's top in-neighbors. Thus v_0, v_1 are each other's top choice in-neighbors, v_2, v_3 are each other's top choice in-neighbors, and so on.
- The nodes v_0, v_2 are each other's second choice in-neighbors, similarly, v_1, v_3 are each other's second choice in-neighbors, and so on. More generally, for any i , if $i \in \{4j, \dots, 4j + 3\}$, then the node v_ℓ , where $\ell = 4j + (i + 2 \bmod 4)$, is v_i 's second choice in-neighbor.
- For any i and any $t \in \{1, \dots, k\}$, if $i \in \{j2^t, \dots, (j+1)2^t - 1\}$ then the node v_ℓ , where $\ell = j2^t + (i + 2^{t-1} \bmod 2^t)$, is v_i 's t -th choice in-neighbor.

For example, v_0 's preference order is $v_1 \succ v_2 \succ v_4 \succ v_8 \succ \dots \succ v_{n/2}$. The other preference orders are analogous. As an example, let $n = 8$ and $V = \{v_0, v_1, \dots, v_7\}$. The preferences of each node is then defined as below:

$$\begin{array}{ll}
 v_0 : v_1 \succ v_2 \succ v_4, & v_1 : v_0 \succ v_3 \succ v_5, \\
 v_2 : v_3 \succ v_0 \succ v_6, & v_3 : v_2 \succ v_1 \succ v_7, \\
 v_4 : v_5 \succ v_6 \succ v_0, & v_5 : v_4 \succ v_7 \succ v_1, \\
 v_6 : v_7 \succ v_4 \succ v_2, & v_7 : v_6 \succ v_5 \succ v_3.
 \end{array}$$

For any branching in G_k on 2^k nodes, we claim its unpopularity factor is at least k . We will prove this claim by induction on k . The base case, i.e., $k = 1$, is trivial. So let us assume that we have $u(\tilde{B}) \geq i$ for any branching \tilde{B} in G_i .

Consider G_{i+1} . Note that v_{2j} and v_{2j+1} are each other's top choice in-neighbors for $0 \leq j \leq 2^i - 1$. Let B be any branching in G_{i+1} . Suppose it is the case that in B , for some j : neither v_{2j} is v_{2j+1} 's in-neighbor nor v_{2j+1} is v_{2j} 's in-neighbor. Then $u(B) = \infty$, because by making v_{2j} the in-neighbor of v_{2j+1} , no node is worse-off and v_{2j} is better-off. We assume $u(B) < \infty$. So it is enough to restrict our attention to the case where for each j we have in B :

- (*) either v_{2j} is v_{2j+1} 's in-neighbor or v_{2j+1} is v_{2j} 's in-neighbor.

For each $j \in \{0, \dots, 2^i - 1\}$, contract the set $\{v_{2j}, v_{2j+1}\}$ into a single node in the graph G_{i+1} . The new graph (call it G'_i) is on 2^i nodes and it is exactly the same as G_i except that there are 2 parallel edges between every adjacent pair of nodes now – both these edges have the same rank.

Perform the same contraction step on the branching B as well. By (*), it follows that the contracted B (call it B') is a branching such that B' uses at most 1 edge in any pair of parallel edges in G'_i . Thus B' is a branching in G_i and we can use induction hypothesis to conclude that $u(B') \geq i$.

Claim. There is a branching A' in G'_i such that $\phi(A', B') \geq i$ and $\phi(B', A') = 1$. Moreover, the lone vertex that prefers B' to A' is a root in A' .

We will first assume the above claim and finish our proof on $u(B)$. Then we will prove this claim. Opening up the size-2 supernodes in B' will create B : let us run the same “opening up” step on A' to create a branching A in G_{i+1} . So $\phi(A, B) \geq i$ and $\phi(B, A) = 1$. We will now modify A to A^* so that $\phi(A^*, B) \geq i + 1$ and $\phi(B, A^*) = 1$.

Let v_{2j} be the lone vertex that prefers B to A . By the “opening up” step in B , v_{2j+1} has v_{2j} as its in-neighbor. The branching A^* will affect only the 2 nodes v_{2j} and v_{2j+1} in A . Every other node will have the same in-neighbor in A^* as in A . The above claim tells us that v_{2j} is a root in A . Make v_{2j+1} a root in A^* and v_{2j} 's in-neighbor will be v_{2j+1} . The node v_{2j} was the only node that preferred B to A and now v_{2j} prefers A^* to B . However there is one node that prefers B to A^* : this is v_{2j+1} . Recall that v_{2j+1} 's in-neighbor in B , just as in A , is its top-choice neighbor v_{2j} while v_{2j+1} is a root in A^* . Thus $\phi(A^*, B) \geq i + 1$ and $\phi(B, A^*) = 1$.

Proof of Claim. Let \tilde{A} be a branching that maximizes $\phi(\tilde{A}, B')/\phi(B', \tilde{A})$. Let $\{u_1, \dots, u_j\}$ be the nodes that prefer B' to \tilde{A} . There is no loss in assuming that u_1, \dots, u_j are root nodes in \tilde{A} . For each i , let n_i be the number of nodes in the arborescence rooted at u_i in \tilde{A} that have different in-neighbors in \tilde{A} and B' – note that each of these nodes prefers \tilde{A} to B' (since the ones who prefer B' to \tilde{A} are root nodes in \tilde{A}).

Let $n_t = \max\{n_i : 1 \leq i \leq j\}$. Let \tilde{A}_t be the maximal sub-arborescence of \tilde{A} rooted at u_t , and let X be those n_t nodes in \tilde{A}_t that prefer \tilde{A} to B' . We construct a branching A' . Let us define an arborescence A'_t rooted at u_t by modifying \tilde{A}_t as follows: for each $w \notin \tilde{A}_t$ that is the descendant of some $v \in \tilde{A}_t$ in B' , we add $B'(w)$. We define A' as the branching that contains A'_t and for which $A'(v) = B'(v)$ for each $v \notin \tilde{A}_t$. So each node in \tilde{A}_t has the same in-neighbor in A' as in B' , except for the nodes in $X \cup \{u_t\}$.

The n_t nodes in X prefer A' to B' , and u_t prefers B' to A' , so we have $\frac{\phi(A', B')}{\phi(B', A')} = n_t$. By $u(B) < \infty$ we get $u(B') < \infty$, which implies that every node that prefers \tilde{A} to B' is contained in a sub-arborescence of \tilde{A} rooted at one of the nodes u_1, \dots, u_j . Thus we have $\phi(\tilde{A}, B') = \sum_{i=1}^j n_i$, yielding $\frac{\phi(A', B')}{\phi(B', A')} = n_t \geq \frac{1}{j} \sum_{i=1}^j n_i = \frac{\phi(\tilde{A}, B')}{\phi(B', \tilde{A})}$ and the claim follows. \square

6 The Popular Branching Polytope

We now describe the popular arborescence polytope of $D = (V \cup \{r\}, E)$ when every node has a weak ranking over its incoming edges. Projecting out variables $x_{(r,v)}$ for all $v \in V$ from these constraints will give us the formulation of the popular branching polytope \mathcal{B}_G of G .

The arborescence polytope \mathcal{A}_D of D is described below [30].

$$\sum_{e \in E[X]} x_e \leq |X| - 1 \quad \forall X \subseteq V, |X| \geq 2. \quad (1)$$

$$\sum_{e \in \delta^-(v)} x_e = 1 \quad \forall v \in V \quad \text{and} \quad x_e \geq 0 \quad \forall e \in E. \quad (2)$$

We will define a subgraph $D^* = (V \cup \{r\}, E_{D^*})$ of D : this is essentially the *expanded* version of the graph D' from our algorithm. The edge set of D^* is:

$$E_{D^*} = \bigcup_{X \in \mathcal{X}'} S(X) \cup \{(u, v) \in E : X_v \in \mathcal{X}', u \notin X_v, \text{ and } (u, v) \text{ is undominated in } \delta^-(X_v)\}.$$

Thus each set $X \in \mathcal{X}'$, which is a node in D' , is replaced in D^* by the nodes in X and with the edges in $S(X)$ between them. We also replace edges in D' between sets in \mathcal{X}' by the original edges in E .

Lemma 6.1. *If every node has a weak ranking over its incoming edges, then every popular arborescence in D is an arborescence in D^* that includes exactly $|X| - 1$ edges from $S(X)$ for each $X \in \mathcal{X}'$.*

Proof. Let A be a popular arborescence in D and let $X \in \mathcal{X}'$. By Lemma 3.6 we know $|A \cap \delta^-(X)| = 1$, and the proof of Theorem 3.4 tells us that the unique edge in $A \cap \delta^-(X)$ is contained in D^* . So A contains $|X| - 1$ edges from $E[X]$ for each $X \in \mathcal{X}'$. We show that these $|X| - 1$ edges are in $S(X)$.

Let $u \in X$. Suppose $A(u) \in E[X] \setminus S(X)$. This means that either (i) $A(u)$ is dominated by some edge in $E[X] \cup \delta^-(X)$ or (ii) u is indifferent between $A(u)$ and some edge in $\delta^-(X)$. Let $\mathcal{Y} = \{Y_v : v \in V\}$ be a dual certificate for A . We know that $Y_u \subseteq X_u \subseteq X$ (by Lemma 3.5). Since the entry point of A into X is not in Y_u , there is an edge $e \in S(X) \cap \delta^-(Y_u)$.

Let e enter $w \in Y_u$. Since $e \in S(X)$, we have $e \succ_w A(w)$ or $e \sim_w A(w)$, hence $c_A(e) \in \{0, 1\}$. If $w \neq u$, then $\{w\} \in \mathcal{Y}$ by Corollary 2.6, and hence e enters two sets Y_u and $\{w\}$ —thus the constraint in LP2 corresponding to edge e is violated. If $w = u$, then $e \in S(X)$ and $A(u) \in E[X] \setminus S(X)$ imply that $e \sim_u A(u)$ is not possible, because u has a weak ranking over its incoming edges (note that for any two edges f and f' in $\delta^-(u) \cap E[X]$ that are tied in the preference list of u , either $\{f, f'\} \subseteq S(X)$ or $\{f, f'\} \cap S(X) = \emptyset$). Since e is undominated, this implies $e \succ_u A(u)$, yielding $c_A(e) = 0$. Since e enters Y_u , the constraint corresponding to e in LP2 is again violated. So $A(u) \in S(X)$, i.e., $A \cap E[X] \subseteq S(X)$. \square

Hence, every popular arborescence in D satisfies constraints (1)-(2) along with constraints (3) given below, where E_{D^*} is the edge set of D^* .

$$\sum_{e \in E[X]} x_e = |X| - 1 \quad \forall X \in \mathcal{X}', |X| \geq 2 \quad \text{and} \quad x_e = 0 \quad \forall e \in E \setminus E_{D^*} \quad (3)$$

Note that constraints (3) define a face \mathcal{F} of the arborescence polytope \mathcal{A}_D of D . Thus every popular arborescence in D belongs to face \mathcal{F} .

Consider a vertex in face \mathcal{F} : this is an arborescence A in D of the form $A' \cup_{X \in \mathcal{X}'} A_X$ where (i) A_X is an arborescence in $(X, S(X))$ whose root is an entry-point of X and (ii) $A' = \{e_X : X \in \mathcal{X}'\}$ where e_X is an edge in D^* entering the root of A_X . Observe that A is a possible output of the algorithm presented in Section 3 (with appropriate tie breaking). Therefore, by Theorem 3.3, A is popular in D . Thus we can conclude Theorem 6.2.

Theorem 6.2. *If each node has a weak ranking over its incoming edges, then face \mathcal{F} defined by constraints (1)-(3) is the popular arborescence polytope of D .*

So the popular branching polytope \mathcal{B}_G is described by constraints (1)-(3) where for every $v \in V$, $\sum_{e \in \delta^-(v)} x_e = 1$ in (2) is replaced by $\sum_{e \in \delta_G^-(v)} x_e \leq 1$, i.e., we project out $x_{(r,v)}$. This shows the upper bound given in Theorem 1.3.

A compact extended formulation. We now describe a compact extended formulation of the popular arborescence polytope of D when node preferences are weak rankings. We know from Lemma 6.1 that every popular arborescence in D is an arborescence in D^* that includes exactly $|X| - 1$ edges from $S(X)$ for each $X \in \mathcal{X}'$. Conversely, any such arborescence in D^* is a popular arborescence in D (as a consequence of Theorem 3.3).

Thus the popular arborescence polytope of D is the face of the arborescence polytope of D^* that corresponds to the constraints $\sum_{e \in E_{D^*}[X]} x_e = |X| - 1$ for all $X \in \mathcal{X}'$. Let \mathcal{A}_{D^*} be the arborescence polytope of $D^* = (V \cup \{r\}, E_{D^*})$. We will now use a compact extended formulation of \mathcal{A}_{D^*} .

Recall that $|V| = n$. Let \mathcal{P}_{D^*} be the polytope defined by constraints (4)-(7) on variables x_e, f_e^v for $e \in E_{D^*}$ and $v \in V$. It is known [7] that \mathcal{P}_{D^*} is a compact extended formulation of the arborescence polytope \mathcal{A}_{D^*} . Note that \mathcal{A}_{D^*} is the projection of \mathcal{P}_{D^*} on to x -space.

$$x_e \geq f_e^v \geq 0 \quad \forall v \in V \text{ and } e \in E_{D^*} \quad (4)$$

$$\sum_{e \in \delta^+(r)} f_e^v = 1 \quad \forall v \in V \quad (5)$$

$$\sum_{e \in \delta^+(u)} f_e^v - \sum_{e \in \delta^-(u)} f_e^v = 0 \quad \forall u, v \in V, u \neq v \quad (6)$$

$$\sum_{e \in E_{D^*}} x_e = n. \quad (7)$$

For any $X \subseteq V$ with $|X| \geq 2$, the constraint $\sum_{e \in E_{D^*}[X]} x_e \leq |X| - 1$ is a valid inequality for \mathcal{A}_{D^*} and also for \mathcal{P}_{D^*} . Thus the intersection of \mathcal{A}_{D^*} along with the tight constraints $\sum_{e \in E_{D^*}[X]} x_e = |X| - 1$ for all $X \in \mathcal{X}'$ is a face of \mathcal{A}_{D^*} . Call this face \mathcal{F}_{D^*} —this is the popular arborescence polytope of D .

Consider the face of \mathcal{P}_{D^*} that is its intersection with $\sum_{e \in E_{D^*}[X]} x_e = |X| - 1$ for all $X \in \mathcal{X}'$. This face of \mathcal{P}_{D^*} is an extension \mathcal{F}_{D^*} . The total number of constraints used to describe this face of \mathcal{P}_{D^*} is $O(mn)$.

Lower bound for the popular branching polytope \mathcal{B}_G of G . Let $G = (V, E)$ be the complete bidirected graph where every node $v \in V$ regards all other nodes $u \in V$ as top-choice in-neighbors. Here $\mathcal{X}' = \{V\}$. We claim that in any minimal system contained in (1)-(3) (after projecting out variables $x_{(r,v)}$ for all $v \in V$), the constraint $\sum_{e \in E[X]} x_e \leq |X| - 1$ for every $X \subset V$ with $|X| \geq 2$ has to be present. This is because a cycle on the nodes in X along with any rooted arborescence A on $V \setminus X$ satisfies all the remaining constraints.

Thus any minimal system of inequalities from (1)-(3) that describes \mathcal{B}_G has to contain $2^n - n - 2$ inequalities from (1): one for every $X \subset V$ with $|X| \geq 2$. Since inequalities in a minimal system are in one-to-one correspondence with the facets of the polyhedron they describe [7, Theorem 3.30], the lower bound given in Theorem 1.3 follows.

Min-cost Popular Branching under Partial Preference Orders. The results above imply that the min-cost popular branching problem has a polynomial time algorithm when every node has a weak ranking over its incoming edges. The following theorem reveals that the assumption of weak rankings is indeed crucial, as the problem becomes NP-hard when nodes preferences are in arbitrary partial order. At an intuitive level, the construction makes use of the fact that for arbitrary partial orders, an edge that is incomparable to a safe edge need not be safe itself, and hence popular arborescences can contain differing numbers of safe edges with respect to the sets in \mathcal{X}' . We show that finding a popular arborescence with a minimum number of such safe edges is NP-hard.

Theorem 6.3. *Given a directed graph where node preferences are in arbitrary partial order, a cost vector γ on the edges, and a number k , it is NP-hard to decide whether there exists a popular branching of cost at most k .*

Proof. We reduce from VERTEX COVER: Given an undirected graph $G' = (V', E')$ and a number k' , does there exist a subset of at most k' vertices of G' that contains at least one endpoint of every edge? We construct an instance of the minimum-cost popular branching problem as follows. First we construct a digraph $D = (V \cup \{r\}, E)$. This node set of D comprises the root r , a special node w , the node set of G' , and a node w_e for every edge $e \in E'$, i.e., $V = V' \cup W' \cup \{w\}$ where $W' = \{w_e : e \in E'\}$.

The node w has an incoming edge f_e from w_e for each $e \in E'$, with $\gamma(f_e) = \infty$. It also has an incoming edge r_w from the root r , with $\gamma(r_w) = 0$. Node w 's preferences are such that w is indifferent about the edges f_e for $e \in E'$, but each of these edges dominates r_w .

Each node w_e for some $e \in E'$ with endpoints $u, v \in V'$ has five incoming edges in D :

- two incoming edges $p_{e,u}$ and $p_{e,v}$ from w with $\gamma(p_{e,u}) = \gamma(p_{e,v}) = 0$,
- an incoming edge $q_{e,u}$ from u with $\gamma(q_{e,u}) = \infty$ and an incoming edge $q_{e,v}$ from v with $\gamma(q_{e,v}) = \infty$,
- an incoming edge r_e from r with $\gamma(r_e) = \infty$.

The preferences of w_e are as follows: $p_{e,u}$ dominates $q_{e,v}$, $p_{e,v}$ dominates $q_{e,u}$, and r_e is dominated by every other incoming edge. All other pairs of edges are incomparable for w_e .

Finally, every node $v \in V'$ has three incoming edges: an edge r_v from r with $\gamma(r_v) = \infty$ and two edges z_v^{yes} , z_v^{no} from w , with $\gamma(z_v^{\text{yes}}) = 1$ and $\gamma(z_v^{\text{no}}) = 0$. Node v prefers z_v^{yes} over r_v but all other incoming edges are incomparable.

Note that every node in V has an incoming edge from r , and hence the popular branchings in D are exactly the r -rooted popular arborescences in D . See Figure 5 for a sketch of the construction. We show that D admits a popular arborescence of cost k' if and only if G' has a vertex cover of cardinality k' .

Let A be a popular arborescence in D with cost k' and let $\mathcal{Y} = \{Y_x \mid x \in V\}$ be the corresponding dual certificate. Without loss of generality we can assume $k' < |V'|$ (as finding a vertex cover of size $|V'|$ is trivial). Hence A contains no edge of infinite cost. Let $S = \{v \in V' : A(v) = z_v^{\text{yes}}\}$. Note that $|S| = \sum_{a \in A} \gamma(a)$ as the edges z_v^{yes} are the only edges with finite non-zero cost in D .

We show that S is a vertex cover in G' . To this end, consider any edge $e \in E'$ with endpoints u and v . Note that $A(w_e) \in \{p_{e,u}, p_{e,v}\}$ as these are the only incoming edges of w_e with finite cost. Without loss of generality assume that $A(w_e) = p_{e,u}$. Note that this implies $c_A(q_{e,u}) = 1$ because $p_{e,u}$ and $q_{e,u}$ are incomparable with respect to w_e 's preferences. Observe that $A(w) = r_w$ as this is the only incoming edge to w with finite cost. Because $f_e \succ_w r_w$, this implies $c_A(f_e) = 0$ and hence $w_e \in Y_w$. By Corollary 2.6, we get $Y_{w_e} = \{w_e\}$. This implies $u \in Y_w$, as otherwise $q_{e,u}$ would enter both Y_w and Y_{w_e} , contradicting the feasibility of the dual certificate, as $c_A(q_u) = 1$. We deduce that r_u enters both Y_w and Y_u , and thus $c_A(r_u) = 2$ by dual feasibility. This is only possible if $A(u) = z_u^{\text{yes}}$, as z_u^{yes} is the only edge dominating r_u . We conclude that $u \in S$ and hence e is covered in S . This shows that a popular arborescence of cost k' in D can be turned into a vertex cover of cardinality k' in G' .

For the reverse direction, consider any vertex cover $S \subseteq V'$ in G' . We construct an arborescence A . For each $v \in V'$, we let $A(v) = z_v^{\text{yes}}$ if $v \in S$ and $A(v) = z_v^{\text{no}}$ otherwise. For each $e \in E'$, let $A(w_e) = p_{e,v}$ where $v \in S$ is one of the endpoints of e that is in the vertex cover. Finally, let $A(w) = r_w$. It is easy to see that A constructed in this way is an arborescence of cost $|S|$, as the only edges with non-zero cost are the edges z_v^{yes} for $v \in S$.

To show that A is popular, we construct a dual certificate $\mathcal{Y} = \{Y_x \mid x \in V\}$ as follows. Let $Y_w = \{w\} \cup W' \cup S$ and let $Y_x = \{x\}$ for all $x \in V \setminus \{w\}$. We verify that this defines a dual certificate for A . First, consider any edge entering w . This edge is either the edge $r_w \in A$, which enters exactly the set Y_w , or it is an edge of type f_e for some $e \in E'$. Because $w_e \in Y_w$, such an edge does not enter any set in \mathcal{Y} . Next consider the edges entering a node w_e for some $e \in E'$ with endpoints u and v . W.l.o.g., let $u \in S$ be the endpoint with $A(w_e) = p_{e,u}$. Note that $c_A(p_{e,u}) = c_A(p_{e,v}) = 1$ and both $p_{e,u}$ and $p_{e,v}$ enter exactly the set Y_{w_e} . Furthermore, $c_A(q_{e,u}) = 1$, $c_A(q_{e,v}) = c(r_e) = 2$ as $p_{e,u}$ is incomparable to $q_{e,u}$ but dominates both $q_{e,v}$ and r_e . As $u \in S \subseteq Y_w$, the edge $q_{e,u}$ enters only the set Y_{w_e} . As w_e is contained in two sets of \mathcal{Y} , the dual

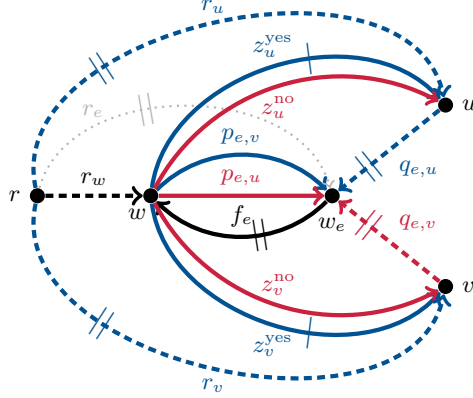


Fig. 5. Sketch of the construction used in the proof of Theorem 6.3. The figure shows the graph D resulting from an instance of VERTEX COVER with two nodes u and v connected by a single edge e . Each solid edge dominates dashed edges of the same color; edge r_e is dominated by all other edges. Edges marked with two crossbars have infinite cost, edges marked with a single crossbar have cost 1, and edges marked with no crossbar have cost 0.

feasibility constraints for the edge $q_{e,v}$ and r_e are also fulfilled. Now consider the edges entering some node $u \in S$. Note that u is contained exactly in the sets Y_u and Y_w . Furthermore $c_A(z_u^{\text{yes}}) = c_A(z_u^{\text{no}}) = 1$ and $c_A(r_u) = 2$, since $A(u) = z_u^{\text{yes}}$ dominates r_u and is incomparable to z_u^{no} . The former two edges originate at $w \in Y_w$ and thus only enter Y_u , hence dual feasibility is fulfilled for all edges entering u . Finally, consider any edge entering some node $u \in V' \setminus S$. Note that $c_A(z_u^{\text{yes}}) = c_A(z_u^{\text{no}}) = c_A(r_u) = 1$ as $A(u) = z_u^{\text{no}}$, which is not dominated by any edge. Because u is only contained in Y_u , dual feasibility is fulfilled at u . We thus showed that \mathcal{Y} is a dual certificate for A . \square

7 Restricted popular branching problems

In this section we prove Theorem 1.6 which we restate here for convenience.

Theorem 1.6. *Given a digraph G where each node has a strict ranking over its incoming edges, it is NP-hard to decide if there exists*

- (a) a popular branching in G where each node has at most 9 descendants;
- (b) a popular branching in G with maximum out-degree at most 2.

In fact, we will show that Theorem 1.6 holds for simple graphs, therefore in this section we will assume that nodes have preferences over their in-neighbors. Nevertheless, we will say that an edge (u, v) is a *top-choice edge*, if u is the best choice for v . The following lemma will be useful to prove Theorem 1.6.

Lemma A. *Let D be a digraph where each node has a strict ranking over its in-neighbors, and let A be a popular arborescence in D with dual certificate \mathcal{Y} . If C is a directed cycle consisting of only top-choice edges, then A enters C exactly once. Let a be the unique edge in $A \cap \delta^-(C)$ (guaranteed by Lemma 2.4), and let Y_a be the unique set in \mathcal{Y} entered by a . Then $C \subseteq Y_a$.*

Proof. Observe that $c_A(e) \leq 1$ for any edge e in C , as e is a top-choice edge. Let c_1, \dots, c_k be the nodes of C in this order, with a pointing to c_k . Since c_k prefers c_{k-1} in C to the tail of a , we get $c_A((c_{k-1}, c_k)) = 0$ and thus $c_{k-1} \in Y_a$ by the constraints of LP2. Supposing $c_{k-2} \notin Y_a$ we get that $(c_{k-2}, c_{k-1}) \notin A$ because exactly one edge of A enters Y_a , by Lemma 2.4. Using that c_{k-1} has strict linear preferences and prefers c_{k-2} most, we obtain $c_A((c_{k-2}, c_{k-1})) = 0$, but this contradicts the constraints of LP2. Hence we get $c_{k-2} \in Y_a$ as well. Repeatedly applying this argument, we get that $C \subseteq Y_a$. \square

Proof of Theorem 1.6, part (a). The reduction is from the NP-hard problem 3-SAT where we are given a 3-CNF formula $\varphi = \bigwedge_{j=1}^m c_j$ over variables x_1, \dots, x_n with each clause c_j containing at most 3 literals; the task is to decide whether φ can be satisfied. It is well known that the special case where each variable occurs at most 3 times is NP-hard as well, so we assume this holds for φ .

We define a digraph D_φ as follows. For each variable x_i we define a variable-gadget consisting of a directed 9-cycle A_i on nodes a_i^1, \dots, a_i^9 , together with nodes t_i and f_i , both having in-degree 0 in D_φ . The top choice for any node a_i^k on A_i is its in-neighbor a_i^{k-1} on A_i , its second choice is t_i if $k = 1$ and f_i otherwise.¹¹ Next, for each clause c_j we define a clause-gadget as a directed cycle C_j on nodes c_j^1, \dots, c_j^h where h is the number of literals in c_j ; we may assume $h \in \{2, 3\}$. The top choice for any node c_j^k on C_j is its in-neighbor on C_j . The second choice of c_j^k depends on the k -th literal ℓ_j^k in c_j : it is t_i if $\ell_j^k = x_i$, and it is f_i if $\ell_j^k = \bar{x}_i$. We claim that the digraph D_φ defined this way admits a popular branching where every node has at most 9 descendants if and only if φ is satisfiable.

First let us suppose that we have a satisfying truth assignment for φ ; we create a branching B . If variable x_i is true, then we add to B the edge (f_i, a_i^2) and all edges of A_i except for (a_i^1, a_i^2) ; if x_i is false we add to B the edge (t_i, a_i^1) and all edges of A_i except for (a_i^9, a_i^1) . For each $j \in [m]$ let us choose a literal ℓ_j^k in clause c_j that is true according to our truth assignment. If $\ell_j^k = x_i$, then we let B contain the edge (t_i, c_j^k) ; if $\ell_j^k = \bar{x}_i$, then we let B contain the edge (f_i, c_j^k) . In either case, we also add to B all edges of C_j but the one going into c_j^k ; this finishes the definition of B . Observe that if x_i is true, then the descendants of f_i in B are the nodes of A_i , and the descendants of t_i are among the nodes of those cycles C_j where x_i is a literal of C_j ; the case when x_i is false is analogous. Hence, each node in B has at most 9 descendants as promised.

¹¹ Throughout the rest of the proof, we treat superscripts in a circular way, that is, modulo length of the cycle in question.

Let us prove that B is popular. To this end, we define the graph D'_φ by adding a new dummy root r_0 to D_φ and making it the worst choice for every node in D_φ ; moreover, we define an arborescence A in D'_φ by adding an edge from r_0 to each root of B . Then B is a popular branching in D_φ if and only if A is a popular arborescence in D'_φ . To show the latter, we define a dual certificate \mathcal{Y} that contains the set $V(A_i)$ for each $i \in [n]$, the set $V(C_j)$ for each $j \in [m]$, and a singleton for each node except for those at which an edge of B enters some cycle A_i or C_j . It is straightforward to check that \mathcal{Y} is indeed a dual solution proving the popularity of A in D'_φ , and therefore of B in D_φ .

Let us now suppose that we have a popular branching B with each node having at most 9 descendants; we are going to define a satisfying truth assignment for φ . Note that the only possible roots in B are the nodes in $R = \bigcup_{i \in [n]} \{t_i, f_i\}$, since any other node v has an in-neighbor in R (assuming v to be a root in B , adding an edge from R to v results in a branching more popular than B). Let A be the popular arborescence corresponding to B , and let \mathcal{Y} be a dual certificate proving the popularity of A .

Let e_i be the edge entering A_i in B , and let Y_{e_i} be the unique set in \mathcal{Y} entered by e_i . Similarly, let e'_j be the edge entering C_j in B , and let $Y_{e'_j}$ be the unique set in \mathcal{Y} entered by e'_j . By Lemma A, we know that $A_i \subseteq Y_{e_i}$ and $C_j \subseteq Y_{e'_j}$.

Let us define a truth assignment by setting x_i true if and only if the head of e_i is f_i . Note that all 9 nodes of A_i are descendants of the head of e_i . Hence, the head of an edge e'_j can only be f_i if x_i is false, and similarly, it can only be t_i if x_i is true. Thus, any cycle C_j must be the descendant of a node representing a true literal (where t_i and f_i represent x_i and \bar{x}_i , respectively). By the construction of D_φ , we have that any clause contains a literal set to true by the truth assignment, so φ is satisfiable, proving the theorem. \square

Proof of Theorem 1.6, part (b). We give a reduction from the variant of the DIRECTED HAMILTONIAN PATH problem where the input digraph has a root r with in-degree 0 that is the parent of all other nodes; it is easy to see that this version is also NP-hard. Let $G = (V \cup \{r\}, E)$ be our given input. For each node we fix an arbitrary ordering on its in-neighbors, and we denote by $n(v, i)$ the i -th in-neighbor of a node $v \in V$.

We are going to construct a digraph D that consists of a node gadget \mathcal{G}_v for each $v \in V$, together with extra nodes r (having in-degree 0) and r' . The gadget \mathcal{G}_v consists of a *core cycle* C_v together with *pendant cycles* $P_{v,1}, \dots, P_{v,d_v}$, each of length d_v , where d_v denotes the in-degree of v in G . The nodes in the core cycle are $c_v^1, \dots, c_v^{d_v}$, those in the i -th pendant cycle $P_{v,i}$ are $p_{v,i}^1, \dots, p_{v,i}^{d_v}$; we treat superscripts modulo d_v . The top choice for any node on these cycles is its in-neighbor within the cycle. The preferences are as follows, where for simplicity we define $c_r^1 := r$.

$$\begin{aligned} c_v^j &: c_v^{j-1} \succ c_{n(v,j)}^1 \succ c_{n(v,j+1)}^1 \succ \dots \succ c_{n(v,d_v)}^1 \succ c_{n(v,1)}^1 \succ \dots \succ c_{n(v,j-1)}^1; \\ p_{v,i}^j &: p_{v,i-1} \succ c_v^j; \\ r' &: r. \end{aligned}$$

This finishes the definition of D .

We claim that G has a Hamiltonian path if and only if D has a popular branching with out-degree at most 2.

For the first direction, suppose that D has such a branching B . Clearly, r is a root of B , since it has in-degree 0. We claim that B is an arborescence with root r . First observe that any pendant cycle must be entered by A once, as otherwise there exists a root of B in the cycle, and adding the second-choice edge of this root node to B (coming from a core cycle unreachable from the pendant cycle) we obtain a branching B' that is more popular than B . Since $(r, v) \in E$ for each $v \in V$, each node c_v^j in a core cycle has an incoming edge from r , so such a node cannot be a root in B either, proving that B is indeed an arborescence. In particular, $\delta^-(C_v) \cap B \neq \emptyset$ for each $v \in V$.

By Lemma A we know that B enters any core cycle C_v exactly once, and therefore $|B \cap C_v| = d_v - 1$. In addition, there are exactly d_v edges of B pointing from C_v to the pendant cycles $P_{v,j}$, $j \in [d_v]$, because B is an arborescence. This implies that there can be at most 1 edge of B leaving C_v and pointing to another core cycle C_u , as otherwise the d_v nodes in C_v would together have more than $2d_v$ outgoing edges in B , yielding that at least one of them would have out-degree 3 in B , a contradiction.

Let us now define a set H of edges in G as follows: for each $u, v \in V$, we add (u, v) to H if and only if there is an edge from C_u to C_v in B . Furthermore, we add the edge (r, v) to H if and only if there is an edge from r to C_v in B ; note that there can be at most one such edge, because $(r, r') \in B$ and B has out-degree at most 2. Observe that by the construction of D , we have $H \subseteq E$. Recall that by the previous paragraph, $|\delta^+(v) \cap H| \leq 1$ for each $v \in V \cup \{r\}$, and that $|\delta^-(v) \cap H| \geq 1$ for each $v \in V$. Moreover, H is acyclic, since any cycle in H would imply the existence of a cycle in B as well. Therefore, H must be a Hamiltonian path.

For the other direction, let H be a Hamiltonian path in G , starting from r . We define a popular branching B that happens to be an arborescence. First, for each $(u, v) \in H$ we add (c_u^1, c_v^j) to B where u is the j -th in-neighbor of v , and we also add all edges of C_v to B except for the one pointing to c_v^j ; note that here we cover the case where $u = r$ as well. Notice that for each $v \in V$ there are at most d_v edges of B whose tail is in C_v . Hence, there exist d_v edges in $\delta^+(C_v)$ whose addition to B does not violate our bound on the out-degree and such that each of these edges points to a distinct pendant cycle $P_{v,j}$ (note that any pendant cycle can be connected to any node on C_v). Let us add these edges to B as well, together with all edges in $P_{v,j}$ except for the one whose head already has an incoming edge in B , for each $j \in [d_v]$. Finally, we add the edge (r, r') to B . It is easy to verify that the edge set B obtained this way is indeed an arborescence with root r , and has out-degree at most 2.

It remains to show that B is popular. To this end, we define the graph D' by adding a new dummy root r_0 to D and making it the worst choice for every node in D ; moreover, we define an arborescence A in D' by adding the edge (r_0, r) to B . Then B is a popular branching in D if and only if A is a popular

arborescence in D' . To prove the latter, we provide a dual certificate \mathcal{Y} as follows. For each core or pendant cycle C , we put the set $V(C)$ into \mathcal{Y} , together with a singleton $\{v\}$ for each $v \in V(C)$ except for the node at which B enters C . We also add singletons $\{r'\}$ and $\{r\}$. The set system \mathcal{Y} so obtained contains exactly $|V(D)|$ sets, so it remains to show that it fulfills the conditions of **LP2**. First note that any edge may enter at most two sets from \mathcal{Y} . Note also that if v is a node such that B enters a core or pendant cycle C at v , then $\delta^-(v) \cap B$ is the second choice for v (and its best choice is within C , the set of \mathcal{Y} corresponding to v); otherwise $\delta^-(v) \cap B$ is the best choice for v . From these facts it is straightforward to verify the constraints of **LP2**, so the popularity of B and hence the theorem follows. \square

8 Popular Mixed Branchings

Recall that a *mixed branching* P is a probability distribution over branchings in G , i.e., $P = \{(B_1, p_1) \dots, (B_k, p_k)\}$, where B_i is a branching in G for each i and $\sum_{i=1}^k p_i = 1$, $p_i \geq 0$ for all i . Popular mixed *matchings* were studied in [29] where it was shown that popular mixed matchings always exist and can be efficiently computed. Using the proof and method in [29], we now show that popular mixed branchings also always exist and such a mixed branching can be computed in polynomial time.

The function $\phi(B, B')$ that allowed us to compare two branchings B, B' generalizes to mixed branchings in a natural way. For mixed branchings $P = \{(B_1, p_1) \dots, (B_k, p_k)\}$ and $Q = \{(B'_1, q_1) \dots, (B'_\ell, q_\ell)\}$, the function $\phi(P, Q)$ is the expected number of nodes that prefer B to B' where B and B' are drawn from the probability distributions P and Q respectively; in other words,

$$\phi(P, Q) = \sum_{i=1}^k \sum_{j=1}^{\ell} p_i q_j \phi(B_i, B'_j).$$

Definition 8.1. *A mixed branching P is popular if $\phi(P, Q) \geq \phi(Q, P)$ for all mixed branchings Q .*

Consider the instance on 4 nodes a, b, c, d described in Section 1 that did not admit any popular branching. Let $B_1 = \{(a, b), (b, d), (d, c)\}$, $B_2 = \{(b, a), (a, c), (c, d)\}$, $B_3 = \{(c, d), (d, b), (b, a)\}$, and $B_4 = \{(d, c), (c, a), (a, b)\}$. It can be verified that the mixed matching $P = \{(B_1, 1/4), (B_2, 1/4), (B_3, 1/4), (B_4, 1/4)\}$ is popular.

Proposition 8.2. *Every instance G admits a popular mixed branching.*

The proof of the above proposition is the same as the one given in [29] for popular mixed matchings. Consider a two-player zero-sum game where the rows and columns of the payoff matrix M are indexed by all branchings B_1, \dots, B_N in G . The (i, j) -th entry of the matrix M is $\Delta(B_i, B_j) = \phi(B_i, B_j) - \phi(B_j, B_i)$. A mixed strategy of the row player is a probability distribution $\langle p_1, \dots, p_N \rangle$ over

the rows of M ; similarly, a mixed strategy of the column player is a probability distribution $\langle q_1, \dots, q_N \rangle$ over the columns of M .

The column player seeks to find a mixed branching Q that minimizes $\max_P \Delta(P, Q)$. The row player seeks to find a mixed branching P that maximizes $\min_Q \Delta(P, Q)$. We have:

$$0 \leq \min_Q \max_P \Delta(P, Q) = \max_P \min_Q \Delta(P, Q) \leq 0,$$

where the first inequality follows by taking $P = Q$, the last inequality follows by taking $Q = P$, and the (middle) equality follows from Von Neumann's minimax theorem. Thus $\max_P \min_Q \Delta(P, Q) = 0$, i.e., there exists a probability distribution P over branchings such that $\Delta(P, Q) \geq 0$ for all mixed branchings Q . In other words, P is a popular mixed branching.

Computing a popular mixed branching. As branchings in G and r -arborescences in $D = (V \cup \{r\}, E)$ are equivalent with respect to popularity, we will work in the graph D now. Analogous to [29], instead of *mixed* arborescences, it will be more convenient to deal with *fractional* arborescences.

A fractional arborescence x is a point in the arborescence polytope \mathcal{A} of D , i.e., x is a point that satisfies constraints (1)-(2). So x is a convex combination of arborescences in D , i.e., it is a mixed arborescence $\{(A_1, \alpha_1) \dots, (A_k, \alpha_k)\}$ where $x = \sum_j \alpha_j I_{A_j}$ (note that there may be multiple ways of expressing x as a mixed arborescence).

Conversely, every mixed arborescence $P = \{(A'_1, p_1) \dots, (A'_t, p_t)\}$ maps to a fractional arborescence $\sum_k p_k I_{A'_k}$, where $I_{A'_k}$ is the incidence vector of arborescence A'_k . Thus there is a many-to-one mapping between mixed arborescences and fractional arborescences. Given a fractional arborescence x , there exists a polynomial time algorithm to find a mixed arborescence corresponding to x . More precisely, as the class of arborescence polytopes is solvable, we can apply the technique described in [21, Theorem 3.9].

For any two fractional arborescences x, y , define $\Delta(x, y)$ as follows:

$$\Delta(x, y) = \sum_{u \in V} \sum_{\substack{e \in \delta^-(u) \\ e' \in \delta^-(u)}} x_e y_{e'} \text{vote}_u(e, e'),$$

where $\text{vote}_u(e, e') \in \{1, 0, -1\}$ corresponds to $e \succ_u e'$, $e \sim_u e'$, and $e \prec_u e'$, respectively. Let P, Q be two mixed arborescences and let x, y be the corresponding fractional arborescences. It is easy to show that $\Delta(P, Q) = \Delta(x, y)$.

A fractional arborescence x is popular if $\Delta(x, y) \geq 0$ for all fractional arborescences y . It follows from Proposition 8.2 that popular fractional arborescences always exist in D . The following polytope is the set of all popular fractional arborescences.

$$\begin{aligned} \Delta(x, A) &\geq 0 \quad \forall \text{ arborescences } A \text{ in } D \\ x &\in \mathcal{A} \end{aligned}$$

Observe that the above region is the set of fractional arborescences that do not lose to any *integral* arborescence. This immediately implies that such a fractional arborescence is a popular fractional arborescence.

There are two sets of exponentially many constraints in the above formulation. Both sets of constraints admit efficient separation oracles: to decide if $x \in \mathcal{A}$ or not, a min r -cut needs to be computed in D with edge capacities given by x . If this cut $(S \cup \{r\}, V \setminus S)$ has value less than 1, then the set $V \setminus S$ forms a violating constraint w.r.t. (1); else $x \in \mathcal{A}$.

To decide if $\Delta(x, A) \geq 0$ for all arborescences A , we compute a min-cost arborescence in D with the following edge costs:

$$c_x(e) = \sum_{e' \succ_u e} x_{e'} - \sum_{e' \prec_u e} x_{e'} \quad \forall e \in E.$$

It is simple to check that for any arborescence A , we have $c_x(A) = \Delta(x, A)$. Thus x is *unpopular* if and only if there is an arborescence A with $c_x(A) < 0$.

Since a min-cost arborescence can be computed in polynomial time [13, 30], we can efficiently find a violating constraint $\Delta(x, A) < 0$ if x is unpopular. Thus we can compute a popular mixed arborescence in polynomial time using the ellipsoid method. Hence we have shown the following theorem.

Theorem 8.3. *A popular mixed branching in a digraph G where every node has preferences in arbitrary partial order over its incoming edges can be computed in polynomial time.*

Acknowledgement

Part of this work was done at the 9th Emléktábla workshop in Gárdony, Hungary. We are grateful to the reviewers for their very helpful comments and suggestions. Thanks to Markus Brill for helpful discussions on liquid democracy, and to Nika Salia for our conversations in Gárdony.

Telikepalli Kavitha is supported by the DAE, Government of India, under project no. RTI4001. Tamás Király is supported by the National Research, Development and Innovation Fund of Hungary, grant no. K120254 and Application Domain Specific Highly Reliable IT Solutions project under Thematic Excellence Programme TKP2020-NKA-06. Jannik Matuschke is supported by internal funds of KU Leuven. Ildikó Schlotter was supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2016-3/2018) and Cooperation of Excellences Grant (KEP-6/2018), and by the Hungarian Scientific Research Fund through NFKIH grants no. K128611 and K124171. Ulrike Schmidt-Kraepelin is supported by the Deutsche Forschungsgemeinschaft (DFG) under grant BR 4744/2-1.

References

- [1] D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1034, 2007.

- [2] P. Biró, R. W. Irving, and D. F. Manlove. Popular matchings in the marriage and roommates problems. In *Proceedings of the 7th International Conference on Algorithms and Complexity (CIAC)*, volume 6078 of *Lecture Notes in Computer Science (LNCS)*, pages 97–108. Springer, 2010.
- [3] D. Bloembergen, D. Grossi, and M. Lackner. On rational delegations in liquid democracy. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019.
- [4] C. Blum and C. I. Zuber. Liquid democracy: Potentials, problems, and perspectives. *Journal of Political Philosophy*, 24(2):162–182, 2016.
- [5] M. Brill. Interactive democracy. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) Blue Sky Ideas Track*, pages 1183–1187, 2018.
- [6] Z. Christoff and D. Grossi. Binary voting with delegable proxy: An analysis of liquid democracy. In *Proceedings of the 16th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 134–150, 2017.
- [7] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer Programming*, volume 271 of *Graduate Texts in Mathematics*. Springer, 2014.
- [8] Á. Cseh, C.-C. Huang, and T. Kavitha. Popular matchings with two-sided preferences and one-sided ties. *SIAM Journal on Discrete Mathematics*, 31(4):2348 – 2377, 2017.
- [9] Á. Cseh and T. Kavitha. Popular edges and dominant matchings. *Mathematical Programming*, 172(1):209 – 229, 2017.
- [10] A. Darmann. Popular spanning trees. *International Journal of Foundations of Computer Science*, 24(5):655 – 677, 2013.
- [11] A. Darmann. It is difficult to tell if there is a Condorcet spanning tree. *Mathematical Methods of Operations Research*, 84(1):94 – 104, 2016.
- [12] A. Darmann, C. Klamler, and U. Pferschy. Finding socially best spanning trees. *Theory and Decision*, 70(4):511 – 527, 2011.
- [13] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B(4):233 – 240, 1967.
- [14] B. Escoffier, H. Gilbert, and A. Pass-Lanneau. The convergence of iterative delegations in liquid democracy in a social network. In *Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT)*, volume 11801 of *Lecture Notes in Computer Science (LNCS)*, pages 284 – 297. Springer, 2019.
- [15] Y. Faenza and T. Kavitha. Quasi-popular matchings, optimality, and extended formulations. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 325–344, 2020.
- [16] Y. Faenza, T. Kavitha, V. Powers, and X. Zhang. Popular matchings and limits to tractability. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2790–2809, 2019.
- [17] D. R. Fulkerson. Packing rooted directed cuts in a weighted directed graph. *Mathematical Programming*, 6(1):1 – 13, 1974.
- [18] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

- [19] P. Gärdenfors. Match making: Assignments based on bilateral preferences. *Behavioral Science*, 20(3):166–173, 1975.
- [20] P. Gözl, A. Kahng, S. Mackenzie, and A. Procaccia. The fluid mechanics of liquid democracy. In *Proceedings of the 14th International Workshop on Internet and Network Economics (WINE)*, volume 11316 of *Lecture Notes in Computer Science (LNCS)*, pages 188–202. Springer, 2018.
- [21] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [22] S. Gupta, P. Misra, S. Saurabh, and M. Zehavi. Popular matching in roommates setting is NP-hard. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2810 – 2822, 2019.
- [23] S. Hardt and L. Lopes. Google votes: A liquid democracy experiment on a corporate social network. Technical report, Technical Disclosure Commons, 2015.
- [24] C.-C. Huang and T. Kavitha. Popular matchings in the stable marriage problem. *Information and Computation*, 222:180 – 194, 2013.
- [25] C.-C. Huang and T. Kavitha. Popularity, mixed matchings, and self-duality. *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2294–2310, 2017.
- [26] T. Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43(1):52–71, 2014.
- [27] T. Kavitha. Popular half-integral matchings. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.
- [28] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, and U. Schmidt-Kraepelin. Popular branchings and their dual certificates. Technical Report 1912.01854, arXiv, 2019. <https://arxiv.org/abs/1912.01854>.
- [29] T. Kavitha, J. Mestre, and M. Nasre. Popular mixed matchings. *Theoretical Computer Science*, 412(24):2679–2690, 2011.
- [30] B. Korte and J. Vygen. *Combinatorial Optimization*. Springer, 2012.
- [31] G. Kotsialou and L. Riley. Incentivising participation in liquid democracy with breadth first delegation. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 638–644, 2020.
- [32] 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)*, volume 4957 of *Lecture Notes in Computer Science (LNCS)*, pages 593–604. Springer, 2008.