

Color-Constrained Arborescences in Edge-Colored Digraphs

P. S. Ardra* Jasine Babu* R. Krithika*
Deepak Rajendraprasad*

Submitted: Aug 4, 2025; Accepted: Feb 5, 2026; Published: Feb 27, 2026
© The authors. Released under the CC BY-ND license (International 4.0).

Abstract

Consider a multigraph G whose edges are colored from $[q]$ (q -colored graph) and $\alpha = (\alpha_1, \dots, \alpha_q) \in \mathbb{N}^q$ ($color$ -constraint). A subgraph H of G is called α -colored if H has exactly α_i edges of color i for each $i \in [q]$. In this paper, we focus on α -colored arborescences (spanning out-trees) in q -colored multidigraphs. We study the decision, counting and search versions of this problem. It is known that the decision and search problems are polynomial-time solvable when $q = 2$ [Barahona and Pulleyblank, Discret. Appl. Math. 1987] and that the decision problem is NP-complete when q is arbitrary [Ardra et al., arXiv 2024]. However the complexity status of the problem for fixed q was open for $q > 2$.

We solve this problem using an algebraic approach. Given a q -colored digraph G and a vertex s in G , we construct a symbolic matrix in $q - 1$ indeterminates such that the number of α -colored arborescences in G rooted at s for all color-constraints $\alpha \in \mathbb{N}^q$ can be read from its determinant polynomial. This result extends Tutte's matrix-tree theorem and gives a polynomial-time algorithm for the counting and decision problems for fixed q . We use it to design an algorithm that finds an α -colored arborescence when one exists. We also study the weighted variant of the problem and give a polynomial-time algorithm (when q is fixed and weights are polynomially bounded) which finds a minimum weight solution.

Mathematics Subject Classifications: 05C88, 05C89

*Department of Computer Science and Engineering, Indian Institute of Technology Palakkad, Palakkad, India (111914001@smail.iitpkd.ac.in, jasine@iitpkd.ac.in, krithika@iitpkd.ac.in, deepak@iitpkd.ac.in).

1 Introduction

Colors must fit together as pieces in a puzzle or cogs in a wheel.

Hans Hofmann, Painter

Many textbook problems in graph theory are on the existence of subgraphs of a specific kind – triangles, cycles, spanning trees, perfect matchings, Hamiltonian cycles, and so on. A similar, albeit shorter, list of problems entice us on directed graphs too – directed cycles, arborescences, directed Hamiltonian cycles. The corresponding algorithmic questions – existence, counting, finding, enumerating – serve as common invitations to the study of graph algorithms. One way to spice up these problems is to color the edges of the input graph and look for solutions in which the colors “fit together”. More precisely, we assume that the edges of the input graph (undirected or directed, simple or otherwise) are colored with a palette $[q]$ of colors and that a color-constraint $\alpha \in \mathbb{N}^q$ is specified. We then restrict the solution-space to those subgraphs which are α -colored, that is, those which have exactly α_i edges of color i for each $i \in [q]$. We call these color-constrained subgraph problems.

These problems take a special flavor when the subgraphs we are looking for turn out to be bases of a matroid (spanning tree) or common bases of two or more matroids (perfect matching in bipartite graphs, arborescences in digraphs). The color-constraint we formalized above is also matroidal – the edge-set of the input graph forms the ground set, the color-classes partition the ground set, and the color-constraint gives the capacity constraint for a partition matroid. Hence adding the color-constraint adds one more matroidal dimension to the problem. In particular, a color-constrained spanning tree is a common basis of two matroids while a color-constrained bipartite perfect matching and a color-constrained arborescence are common bases of three matroids. Finding a largest common independent set of k matroids on the same ground set is the k -MATROID INTERSECTION problem. The problem is NP-hard for $k \geq 3$ [20]. A greedy algorithm solves it efficiently when $k = 1$. There are many general polynomial-time algorithms to solve the case when $k = 2$ and more efficient specialized algorithms for common special cases like bipartite matching and arborescence [20].

Color-constrained subgraph problems of this matroidal nature have received considerable attention since 1980s [2, 7, 8, 18, 19]. Work on finding an α -colored spanning tree in a q -colored undirected graph was started by Papadimitriou and Yannakakis in 1982 with a solution for the case of $q = 2$ [18, Proposition 3]. They employed an elegant exchange procedure that starts with an efficiently-computable spanning tree, performs a sequence of “edge swaps” and ends with the required solution. Gabow and Tarjan [7] observed that a similar “exchange sequence” exists in the more general setting of matroids and used it to find a min-weight α -colored spanning tree of a weighted 2-colored graph in 1984. An alternative algorithm for the same problem was given by Gusfield [8] at around the same time. Later, Rendl and Leclerc gave a polynomial-time algorithm for arbitrary number of colors [19] in 1989.

The problem of finding a $(k, n/2 - k)$ -colored perfect matching in an n -vertex graph, often called EXACT PERFECT MATCHING, was also born out of the same 1982 paper by Papadimitriou and Yannakakis [18]. This one, being a special case of 3-MATROID INTERSECTION, turned out to be much tougher to tame than its sibling. Papadimitriou and Yannakakis [18] conjectured that the problem is NP-hard. Mulmuley, Vazirani and Vazirani [17] gave a randomized polynomial-time algorithm for it in 1987. Despite several subsequent works on the problem, it still remains as one of those rare examples of a problem which has a randomized polynomial-time algorithm but no known deterministic polynomial-time algorithm. However, finding a color-constrained perfect matching in a q -colored bipartite graph is NP-hard if q is unbounded [11, 12].

In this work, we are interested in finding and counting α -colored arborescences rooted at a given vertex s in q -colored digraphs. Before describing our work further, we make the decision variant of the problem precise. For reasons that will become clear soon, we only specify the constraints on the first $q - 1$ colors. Given a color-constraint $\alpha = (\alpha_1, \dots, \alpha_{q-1}) \in \mathbb{N}^{q-1}$, an arborescence of a q -colored n -vertex digraph is said to be an α -colored s -arborescence if it is rooted at s and has exactly α_i edges of color i for each $i \in [q - 1]$. The number of edges of color q is automatically constrained.

COLOR-CONSTRAINED ARBORESCENCE (CC-ARB)

Input: A q -colored n -vertex multidigraph G , a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$.

Question: Does G have an α -colored s -arborescence?

CC-ARB is a special case of 3-matroid intersection and we do not have any off-the-matroid-shelf algorithm for the problem. In 1987, Barahona and Pulleyblank [2] gave a polynomial-time algorithm for the case of $q = 2$. Recently, Ardra et al. [1] showed that the problem is NP-complete for arbitrary q . The complexity status of the problem for fixed q was open for $q > 2$.

We show that, for every fixed q , CC-ARB is polynomial-time solvable. Our approach is algebraic. We associate a symbolic Laplacian matrix in $q - 1$ indeterminates with a q -colored multidigraph G (Definition 1) and show that the number of α -colored arborescences in G rooted at a vertex s is the coefficient of a particular monomial of one of its minors (Theorem 2). Theorem 2 may be viewed as a colored analogue of Tutte's matrix-tree theorem [21]. We prove it using a result of Moon [15] which is a generalized form of the matrix-tree theorem. Theorem 2 gives a way to solve CC-ARB in the time required to compute the determinant of an $(n - 1) \times (n - 1)$ matrix in which each entry is a linear polynomial in $q - 1$ indeterminates. We also show how to efficiently find an α -colored s -arborescence (when one exists), by iterating the decision algorithm over all the edges (Theorem 10).

We need some additional work to extend our techniques to the weighted version of the problem. Here, the objective is to find a min-weight α -colored s -arborescence in a weighted q -colored multidigraph. We need to define a weighted symbolic Laplacian matrix (Definition 11). If the min-weight α -colored s -arborescence is unique, then we can extract

its weight from a minor of this matrix. Since we cannot assume this in general, we devise a method using multiple weighted symbolic Laplacian matrices. Note that the standard way to workaroud the non-uniqueness is to use the Isolation Lemma [16] that makes the algorithm randomized. Our algorithm is deterministic and its running time is polynomial in n when the weights are polynomially bounded. Once we have an algorithm to find the minimum weight, we show how to use it to find a min-weight solution (Theorem 15).

In single-precision arithmetic, we typically assume that any intermediate value during the computation fits into a single register. Since the determinant computations in our algorithms may involve exponentially large integers, it is impractical to assume that we can do these computations in single-precision arithmetic. Hence we use modular arithmetic over multiple single-precision primes [9] to break down the computation. However, we first report the running times in number of multi-precision operations to keep the analysis oblivious to the above issue. Then we separately analyze the blowup when using multiple single-precision computations.

An independent recent work on group-labeled matroid bases by Hörsch et al. [10] applies to arborescences and may be used to show that CC-ARB is polynomial-time solvable for every fixed q . However, our approach is different and the definition of a symbolic matrix that is a natural generalization of the classical Laplacian matrix to colored graphs enables us to prove a directed matrix tree theorem for colored graphs. This brings to light a close connection between the determinant of the generalized Laplacian and the number of arborescences of each color signature.

Terminology. For $q \in \mathbb{N}$, $[q]$ denotes the set $\{1, 2, \dots, q\}$. We refer to the book by Diestel [6] for standard graph-theoretic definitions and terminology not defined here. For a digraph G , let $V(G)$ and $E(G)$ denote the sets of its vertices and edges/arcs, respectively. The *out-degree* (resp., *in-degree*) of a vertex v is the number of edges leaving (resp. entering) v . For a digraph G , let \overleftarrow{G} denote the graph obtained from G by reversing its edges. Given a vertex s in a directed multigraph G , a *spanning out-tree* (resp., *spanning in-tree*) T rooted at s is an acyclic spanning subgraph of G in which every vertex other than s has in-degree (resp. out-degree) one. A spanning out-tree rooted at s is also called an *s-arborescence*.

We assume that the input q -colored digraph is in the adjacency matrix representation whose entries are elements of \mathbb{N}^q which encode the number of edges of each color. For a q -colored digraph G and a vertex $s \in V(G)$, the q -colored digraph obtained from G by adding a self-loop of color q at s is denoted as G^s . For vertices i, j and color $c \in [q]$, let d_{ijc} denote the number of c -colored edges from i to j . A *functional digraph* is a digraph in which every vertex has out-degree exactly one. A spanning functional subgraph of an n -vertex digraph has exactly n edges. Given a color-constraint $\alpha = (\alpha_1, \dots, \alpha_{q-1}) \in \mathbb{N}^{q-1}$, a spanning functional subgraph of a q -colored digraph is said to be α -colored if it has exactly α_i edges of color i for each $i \in [q-1]$.

2 Color-Sensitive Matrix-Tree Theorems

Two classical Laplacian matrices associated with a directed multigraph G on the vertex-set $[n]$ are the out-degree Laplacian L_G defined as

$$L_G[i, j] = \begin{cases} -d_{ij}, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n d_{ik}, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (1)$$

and the in-degree Laplacian L'_G defined as

$$L'_G[i, j] = \begin{cases} -d_{ji}, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n d_{ki}, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (2)$$

where d_{ij} is the number of edges from vertex i to vertex j . It is easy to verify that the in-degree Laplacian of G is same as the out-degree Laplacian of \overleftarrow{G} and vice-versa.

Tutte's directed matrix-tree theorem [21] states that, if G is a digraph without self-loops, then the determinant of the submatrix obtained by deleting the i -th row and column of the out-degree Laplacian L_G (resp. the in-degree Laplacian L'_G) is the number of spanning in-trees (resp. arborescences) of G rooted at i . Even though our results are stated on arborescences, we will work with the out-degree Laplacian and in-trees in the proof of Theorem 2 so that we can use the language of functional digraphs. Instead of doing the row and column deletions on L_G , one can also add a self-loop at i to (the otherwise loopless) G and then compute the determinant of the out-degree Laplacian of the resulting digraph G^i . One can verify this equivalence easily by seeing the new Laplacian L_{G^i} as the result of deleting the $(n+1)$ -th row and column of the Laplacian of a supergraph of G obtained by adding a new vertex $n+1$ and a new edge $(i, n+1)$ to G . A similar observation holds good for the in-degree Laplacian too.

In order to count color-constrained arborescences in q -colored digraphs, we define two symbolic matrices in $q-1$ indeterminates to take the role of the two Laplacians.

Definition 1. For a q -colored multidigraph G on the vertex-set $[n]$, the symbolic out-degree Laplacian matrix L_G and the symbolic in-degree Laplacian matrix L'_G are

$$L_G[i, j] = \begin{cases} -\sum_{c=1}^q d_{ijc} x_c, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n \sum_{c=1}^q d_{ikc} x_c, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (3)$$

and

$$L'_G[i, j] = \begin{cases} -\sum_{c=1}^q d_{jic} x_c, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n \sum_{c=1}^q d_{kic} x_c, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (4)$$

where x_1, \dots, x_{q-1} are indeterminates, $x_q = 1$ and d_{ijc} is the number of c -colored edges from vertex i to vertex j .

When each x_c is set to 1, L_G collapses to the out-degree Laplacian of the underlying uncolored digraph. It is also interesting that we can decompose L_G as

$$L_G = \sum_{c=1}^q L_{G_c} x_c, \tag{5}$$

where L_{G_c} is the classical out-degree Laplacian of the spanning subgraph G_c of G consisting of the c -colored edges of G . A similar collapse and decomposition applies to the symbolic in-degree Laplacian too. The course is now set to introduce the workhorse of this paper.

Theorem 2 (Extended Tutte's Theorem). *Given a q -colored loopless multidigraph G , a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, the number of α -colored s -arborescences of G is the coefficient of the monomial $\prod_{c=1}^{q-1} x_c^{\alpha_c}$ in the determinant polynomial of the submatrix obtained by deleting the row and column corresponding to s from the symbolic in-degree Laplacian L'_G of G .*

The original Tutte's theorem has many proofs [3, 4, 5, 21], the first of which predates that of Tutte by nearly a century [3]. A long but instructive way to prove Theorem 2 would be to adapt one of the elementary proofs of Tutte's theorems which expands the determinant of $L_{\widehat{G^s}}$ (where G^s is G with a self-loop added to s) using the Leibniz formula where each non-zero term corresponds to a product over the edges of a functional subgraph of $\widehat{G^s}$ and then showing that all terms corresponding to functional subgraphs with non-singleton cycles cancel. However, in the interest of space, we derive our result from the following generalization of Tutte's theorem by Moon [15].

Theorem 3 (Corollary 4.1 in [15]). *Let M be an $n \times n$ matrix defined as*

$$M[i, j] = \begin{cases} -z_{ij} & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n z_{ik} & \text{if } i = j, 1 \leq i \leq n \end{cases}, \tag{6}$$

where z_{ij} are indeterminates for each $i, j \in [n]$. Then

$$\det(M) = \sum_{D \in \mathcal{D}} \prod_{(i,j) \in E(D)} z_{ij}, \tag{7}$$

where \mathcal{D} is the set of all functional digraphs D with $V(D) = [n]$ such that each cycle of D has length 1.

We refer the reader to the remark after Corollary 4.1 in [15] to see how Tutte's theorem follows from Theorem 3.

The role of our symbolic Laplacians in counting is first illustrated for functional subgraphs rather than for arborescences so that the key idea is not lost in self-loops and direction reversals. The shorthand x^α will be used to denote the monomial $\prod_{c=1}^{q-1} x_c^{\alpha_c}$ henceforth.

Lemma 4. *Given a q -colored multidigraph G on the vertex set $[n]$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, the coefficient of the monomial x^α in the determinant polynomial of the symbolic out-degree Laplacian L_G is the number of α -colored spanning functional subgraphs F of G such that each cycle of F has length 1.*

Proof. If we set $z_{ij} = \sum_{c=1}^q d_{ijc}x_c$ with $x_q = 1$ for all $i, j \in [n]$ (including $i = j$), then the matrix M in (6) is equal to the symbolic out-degree Laplacian L_G of G (Eqn. (3)). Hence Theorem 3 gives

$$\det(L_G) = \sum_{D \in \mathcal{D}} P_D, \quad (8)$$

where \mathcal{D} is the set of all functional digraphs D with $V(D) = [n]$ such that each cycle of D has length 1 and

$$P_D = \prod_{(i,j) \in E(D)} \sum_{c=1}^q d_{ijc}x_c. \quad (9)$$

The polynomial P_D only contains terms coming from the set \mathcal{F} of spanning functional subgraphs of G whose underlying uncolored digraph is D . Each α -colored $F \in \mathcal{F}$ adds a one to the coefficient of x^α in the polynomial P_D and this is the only contribution of F to P_D . Hence the coefficient of the monomial x^α in P_D is the number of α -colored spanning functional subgraphs of G whose underlying uncolored digraph is D . Now it is easy to see that the coefficient of the monomial x^α in $\det(L_G)$ is the total number of α -colored spanning functional subgraphs F of G such that each cycle of F has length 1. \square

We now show how to use Lemma 4 to count arborescences instead of functional subgraphs.

Lemma 5. *Let G be a q -colored loopless multidigraph on the vertex set $[n]$, $s \in V(G)$ and $\alpha \in \mathbb{N}^{q-1}$. Let G^s denote the multidigraph obtained from G by adding a self-loop of color q at the vertex s . Then the coefficient of the monomial x^α in the determinant polynomial of the symbolic in-degree Laplacian L'_{G^s} is the number of α -colored s -arborescences in G .*

Proof. Let \mathcal{T} denote the set of all α -colored s -arborescences in G . Let $\overleftarrow{G^s}$ be the digraph obtained by adding a self-loop of color q at the vertex s to G and reversing the direction of all the edges. Recall that $L'_{G^s} = L_{\overleftarrow{G^s}}$. Let \mathcal{F} denote the set of all α -colored spanning functional subgraphs F of $\overleftarrow{G^s}$ such that all cycles in F have length one. It is easy to see that for any $T \in \mathcal{T}$, the graph $\psi(T)$ obtained by reversing all the edges of T and adding a self-loop of color q at s is in \mathcal{F} . Moreover, ψ is a bijection since all the functional digraphs $F \in \mathcal{F}$ contain only one self-loop – the q -colored self-loop at s and hence there is a unique $T \in \mathcal{T}$ obtained by reversing the edges of F and deleting the self-loop at s . Hence $|\mathcal{T}| = |\mathcal{F}|$. This allows us to count $|\mathcal{T}|$ using Lemma 4. \square

Now we are ready to prove Theorem 2. Since the incoming edges at s do not appear in any s -arborescence, we can as well assume that there are no incoming edges at s in G . Hence the s -th row of L'_{G^s} has x_q (which is 1) at the s -th position (due to the self-loop added) and 0 everywhere else. Hence the determinant of L'_{G^s} is equal to the determinant

of the submatrix of L'_G obtained by deleting the s -th row and column. This completes the proof of Theorem 2.

We end this section with the undirected analogue of Theorem 2. Towards this, we define the symbolic Laplacian of a q -colored undirected multigraph as follows.

Definition 6. For a q -colored multigraph G on the vertex-set $[n]$, the symbolic Laplacian matrix L_G is

$$L_G[i, j] = \begin{cases} -\sum_{c=1}^q d_{ijc} x_c, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n \sum_{c=1}^q d_{ikc} x_c, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (10)$$

where x_1, \dots, x_{q-1} are indeterminates, $x_q = 1$ and d_{ijc} is the number of c -colored edges between vertices i and j .

Given a q -colored undirected multigraph G , let D denote the digraph obtained from G by replacing every edge $e = \{u, v\}$ of G by the pair of edges (u, v) , (v, u) of the same color as that of e . It is easy to verify that, once we fix a vertex r in G , there is a bijection between the set of all spanning trees of G and the set of all r -arborescences of D . Now applying Theorem 2 on D with r as the source vertex, we obtain the following result.

Theorem 7 (Extended Kirchhoff's Theorem). *Given a q -colored loopless multigraph G and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, the number of α -colored spanning trees of G is the coefficient of the monomial $\prod_{c=1}^{q-1} x_c^{\alpha_c}$ in the determinant polynomial of the submatrix obtained by deleting the first row and column of the symbolic Laplacian L_G of G .*

3 Algorithms for Unweighted Digraphs

Theorem 2 gives us a way to solve CC-ARB and its counting variant. The time-complexity for both is that of computing the determinant of an $(n-1) \times (n-1)$ matrix where each entry is a linear polynomial in $q-1$ indeterminates. Once we do that, we can count the number of α -colored s -arborescences in the given q -colored graph for all legal α in one shot.

There are multiple algorithms that compute the determinant of a matrix of multivariate polynomials – like Gaussian Elimination, Expansion by Minors, Characteristic Polynomial Method and Evaluate-Interpolate Method – whose running time depends differently on the size of the matrix, the total number of indeterminates, the degree of each polynomial and also on the sparsity of the matrix. Horowitz and Sahni [9] have done a theoretical and empirical comparison of these methods. Marco and Martínez [14] have analyzed the parallelizability of these methods. Since we are in the regime where the number of indeterminates $q-1$ is a fixed quantity and each indeterminate appears with a degree one in each entry of the matrix, the Evaluate-Interpolate method with the following time-complexity is a good choice.

Theorem 8. [9] *Let M be an $n \times n$ matrix whose entries are polynomials in r variables x_1, \dots, x_r and each entry of M has degree at most d in each of the variables. Then the*

Evaluate-Interpolate Algorithm computes $\det(M)$ using $\tilde{\mathcal{O}}(((d+1)n)^r(n^3 + (d+1)n^2))$ arithmetic operations.

Horowitz and Sahni [9] analyze the number of coefficient multiplications needed to compute $\det(M)$ and explicitly ignore the time taken for merging the partial sums. We use $\tilde{\mathcal{O}}$ notation to account for this cost since it at most adds a poly-logarithmic factor. As our algorithms are algebraic, we first analyze the running time in terms of the number of arithmetic operations on the coefficients of the polynomials involved. Though each polynomial in the input matrix has coefficients upper bounded by m , the number of edges in the digraph, the subsequent computations may result in coefficients that are as large as $(2mn)^n$ which makes single-precision arithmetic infeasible even for moderate input size and we will consider this issue separately.

Lemma 9. *Given a q -colored multidigraph G on n vertices and m edges, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, we can count the number of α -colored s -arborescences of G using $\tilde{\mathcal{O}}(2^q \cdot n^{q+2})$ multi-precision operations. Moreover, if there is at least one α -colored s -arborescence in G , then we can find one using $\tilde{\mathcal{O}}(m \cdot 2^q \cdot n^{q+2})$ multi-precision operations.*

Proof. Given G , relabeling its vertices with labels from $[n]$ and constructing L'_G takes $\mathcal{O}(n^2q)$ time. Let M be the sub-matrix of L'_G obtained by deleting the s -th row and column. Observe that each entry of M has degree at most 1 in each of the $q-1$ variables. Theorem 8 guarantees that the Evaluate-Interpolate method takes $\tilde{\mathcal{O}}(2^q \cdot n^{q+2})$ operations. Once the determinant is obtained, a simple linear scan over all monomials, which are $\mathcal{O}(n^q)$ in number, gives us the coefficient of x^α . Therefore, the number of α -colored s -arborescences in G can be determined using $\tilde{\mathcal{O}}(2^q \cdot n^{q+2})$ operations.

Next we describe how to find a solution when one exists. We iterate through each edge e of G , removing it from G if $G \setminus e$ has an α -colored s -arborescence. Since the remaining graph still contains an s -arborescence after each removal, the procedure ends with a subgraph G' of G which has an s -arborescence. If G' has even one more edge than an s -arborescence, then the earliest such edge to be processed in the iteration would have been deleted. Hence G' is itself an α -colored s -arborescence. This procedure takes $\tilde{\mathcal{O}}(m \cdot 2^q \cdot n^{q+2})$ operations. Note that we can remove any duplicate edges (parallel edges of the same color) before the search so that m is at most n^2q . \square

When single-precision arithmetic is infeasible due to blow-up in the size of coefficients, Horowitz and Sahni [9] recommends solving the problem using modular arithmetic. For this, we need to choose a sequence of single-precision odd-primes $p_1 < \dots < p_k$ satisfying the following two conditions and repeat the determinant computations over each \mathbb{Z}_{p_i} . Firstly, we need $\prod_{i=1}^k p_i$ to be larger than the largest coefficient in the final answer which is at most m^n in our case. Secondly, since the Evaluate-Interpolate method needs to evaluate the intermediate polynomials at $2n+1$ distinct points (when all the entries in the input $n \times n$ matrix are linear polynomials), we need each p_i to be at least $2n+1$. Both these conditions can be easily satisfied by choosing p_1 larger than $\max\{m, 2n\}$ and setting

$k = n$. The time to convert the coefficients in the input matrix to \mathbb{Z}_{p_i} can be ignored since every coefficient is at most m which is less than p_i . The time for final reconstruction is $\mathcal{O}(n^2)$ using the Chinese Remainder algorithm [23]. Thus if the number of multi-precision arithmetic operations performed by our algorithm is $t(m, n, q)$ then the overall running time is $\mathcal{O}(n \cdot t(m, n, q) + n^2)$ single-precision arithmetic operations. This gives the main result of this section.

Theorem 10. *Given a q -colored multidigraph G on n vertices and m edges, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, we can count the number of α -colored s -arborescences of G in $\tilde{\mathcal{O}}(2^q \cdot n^{q+3})$ time. Moreover, if there is at least one α -colored s -arborescence in G , then we can find one in $\tilde{\mathcal{O}}(m \cdot 2^q \cdot n^{q+3})$ time.*

The NP-hardness of CC-ARB for arbitrary q was shown by a polynomial-time reduction from HAMILTONIAN PATH in digraphs [1]. This reduction is a linear-size preserving reduction where the number q of colors in the resulting instance of CC-ARB is the number n of vertices in the input instance of HAMILTONIAN PATH. Since HAMILTONIAN PATH in digraphs does not admit a $2^{o(n)}$ -time algorithm under the Exponential Time Hypothesis (ETH) [13], we can conclude that CC-ARB does not admit an $n^{o(\frac{n}{\log n})}$ -time algorithm under ETH. This shows that the running time given by Theorem 10 cannot be significantly improved.

4 Algorithms for Weighted Digraphs

In this section, we bring weighted q -colored digraphs to limelight and search for a min-weight arborescence. In particular, we study the following problem.

MINIMUM WEIGHT COLOR-CONSTRAINED ARBORESCENCE (MIN CC-ARB)
Input: A q -colored multidigraph G with a weight function $w : E(G) \rightarrow \mathbb{Z}^+$, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$.
Output: A min-weight α -colored s -arborescence of G .

Note that we can handle negative integral weights by adding a constant bias to all the edges and we restrict the range of weights to \mathbb{Z}^+ for easier reading. We also assume that the weights are polynomially bounded. Since our interest is limited to finding a min-weight solution, we can get rid of duplicate edges in G . Indeed, if we have parallel edges (i, j) of the same color, we need to retain only a min-weight edge among them in G . We will henceforth assume that our input graph does not have duplicate edges. In the terminology of the previous section, this would amount to saying $d_{ijc} \in \{0, 1\}$. We define the weighted symbolic version of the Laplacian as follows.

Definition 11. For a q -colored multidigraph G on the vertex-set $[n]$ without duplicate edges and a weight function $w : E(G) \rightarrow \mathbb{Z}^+$, the weighted symbolic out-degree Laplacian

matrix $L_{G,w}$ and the weighted symbolic in-degree Laplacian matrix $L'_{G,w}$ are

$$L_{G,w}[i, j] = \begin{cases} -\sum_{c=1}^q w_{ijc} x_c, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n \sum_{c=1}^q w_{ikc} x_c, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (11)$$

and

$$L'_{G,w}[i, j] = \begin{cases} -\sum_{c=1}^q w_{jic} x_c, & \text{if } i \neq j, 1 \leq i, j \leq n \\ \sum_{k=1}^n \sum_{c=1}^q w_{kic} x_c, & \text{if } i = j, 1 \leq i \leq n \end{cases}, \quad (12)$$

where x_1, \dots, x_{q-1} are indeterminates, $x_q = 1$ and w_{ijc} is the weight of the c -colored edge from vertex i to vertex j if one exists and 0 otherwise.

For any subgraph H of G , $w^\times(H)$ will denote $\prod_{e \in E(H)} w(e)$. One can figure out every coefficient of $\det(L_{G,w})$ by following the proof of Lemma 4. In the earlier case, each α -colored spanning functional subgraph F of G with only singleton cycles contributed a one to the coefficient of x^α . In the present case, F will contribute $w^\times(F)$ to the coefficient of x^α . Hence the weighted version of Lemma 4 is the following.

Lemma 12. *Given a q -colored multidigraph G on the vertex set $[n]$ without duplicate edges, a weight function $w : E(G) \rightarrow \mathbb{Z}^+$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, the coefficient of the monomial x^α in the determinant polynomial of the weighted symbolic out-degree Laplacian $L_{G,w}$ is $\sum_{F \in \mathcal{F}_\alpha} w^\times(F)$, where \mathcal{F}_α is the set of α -colored spanning functional subgraphs F of G such that each cycle of F has length one.*

From Lemma 12 one can prove this weighted variant of Theorem 2 by following the same steps as in the previous section.

Theorem 13. *Given a q -colored multidigraph G on the vertex-set $[n]$ without self-loops and duplicate edges, a weight function $w : E(G) \rightarrow \mathbb{Z}^+$, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, the coefficient of the monomial x^α in the determinant polynomial of the submatrix obtained by deleting the s -th row and column of the weighted symbolic in-degree Laplacian $L'_{G,w}$ is $\sum_{T \in \mathcal{T}_\alpha} w^\times(T)$, where \mathcal{T}_α is the set of α -colored s -arborescences of G .*

Now we describe an attempt to find the minimum weight of an α -colored s -arborescence in the given digraph. For any prime r , let w_r denote a new weight function defined by $w_r(e) = r^{w(e)}$ for all $e \in E(G)$. Let $c_{\alpha,r}$ denote the coefficient of x^α in the determinant of the submatrix of L'_{G,w_r} obtained by deleting the s -th row and column. Theorem 13 shows that $c_{\alpha,r} = \sum_{T \in \mathcal{T}_\alpha} r^{w(T)}$, where \mathcal{T}_α is the set of α -colored s -arborescences in G and $w(T) = \sum_{e \in E(T)} w(e)$ is the total weight of T under the original weight function w . This idea is already used in [17] to find a matching of a given weight and in [2] to find an arborescence of a given weight.

If there is a unique min-weight arborescence T^* in \mathcal{T}_α , then we can extract $w(T^*)$ from $c_{\alpha,r}$. In this case, $c_{\alpha,r}$ will be the sum of $r^{w(T^*)}$ and some strictly larger powers of r . Hence $w(T^*)$ is the largest k for which r^k is a factor of $c_{\alpha,r}$. But this method will fail if

the number of min-weight arborescences in \mathcal{T}_α is a multiple of r . One way to solve this is by scaling up the original weights by a factor of $2mn$ and then perturbing the resulting weights by adding a number chosen uniformly at random from $[2m]$. The Isolation Lemma [17] guarantees that with probability at least $1/2$, the min-weight solution is unique. And hence we can extract $w(T^*)$ from $c_{\alpha,2}$. But this results in a randomized algorithm. Hence we propose an alternate method to tackle the non-uniqueness.

Our method relies on the observation that we do not need uniqueness of the min-weight solution for the extraction of the minimum weight from $c_{\alpha,r}$ to work. It is sufficient that the number of min-weight solutions is not a multiple of r . Let w_{min} be the minimum weight of a solution and t be the number of min-weight solutions in \mathcal{T}_α . One can verify that $\max\{k : r^k \mid c_{\alpha,r}\} \geq w_{min}$ and the equality holds if and only if t is not a multiple of r . Let r_1, \dots, r_n be n distinct single-precision primes larger than m . Since $\prod_{i=1}^n r_i > m^n$, no integer $t \in [m^n]$ is a multiple of all the primes in this list. We compute c_{α,r_i} and then $f(r_i) = \max\{k : r_i^k \mid c_{\alpha,r_i}\}$ for each $i \in [n]$. Let r_j be a prime in the list r_1, \dots, r_n that does not divide t . Hence $f(r_j) = w_{min}$. For every other prime r_i in the list, $f(r_i) \geq w_{min}$. Hence, even though we do not know r_j in advance, we can extract w_{min} as $\min\{f(r_i) : i \in [n]\}$. Once we know w_{min} , we can find a min-weight solution by iterating through all the edges e in $E(G)$ and deleting e from G if the minimum weight of a solution in $G \setminus e$ is also w_{min} . This procedure will terminate with a min-weight solution. The next lemma is the analogue of Lemma 9 for the weighted case.

Lemma 14. *Given a q -colored multidigraph G on n vertices and m edges, a weight function $w : E(G) \rightarrow [W]$ with $W \in n^{\mathcal{O}(1)}$, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, we can find the minimum weight of an α -colored s -arborescence in G using $\tilde{\mathcal{O}}(2^q \cdot n^{q+3})$ multi-precision operations. Moreover, we can find a min-weight solution using $\tilde{\mathcal{O}}(m \cdot 2^q \cdot n^{q+3})$ multi-precision operations.*

The time to construct L'_{G,w_r} is $\mathcal{O}(n^2 q \log W)$ since r^k can be computed in $\mathcal{O}(\log k)$ multi-precision operations. Let M be the sub-matrix of L'_{G,w_r} obtained by deleting the s -th row and column. Each entry of M has degree at most 1 in each of the $q-1$ variables and Theorem 8 guarantees that the Evaluate-Interpolate method takes $\tilde{\mathcal{O}}(2^q \cdot n^{q+2})$ multi-precision operations. Once the determinant is obtained, a linear scan over all monomials gives us the coefficient $c_{\alpha,r}$ of x^α . Note that $c_{\alpha,r} \leq m^n r^{nW}$. Therefore, computing $f(r)$ takes $\mathcal{O}(\log(nW))$ divisions. Since we compute L'_{G,w_r} and $f(r)$ for each of the n primes, the total number of multi-precision operations to find w_{min} is $\tilde{\mathcal{O}}(n(2^q \cdot n^{q+2} + \log(nW)))$ which is claimed in Lemma 14 since $W \in n^{\mathcal{O}(1)}$. The time complexity of finding a min-weight solution is m times the above. This completes the time complexity analysis of Lemma 14.

Now, we will analyze the time complexity in single-precision operations using modular arithmetic and Chinese Remainder algorithm. Notice that now we have to work with much larger coefficients than in the unweighted case. Each entry of L'_{G,w_r} has coefficients upper bounded by r^W and the coefficients in the resulting determinant polynomial are at most $m^n r^{nW}$. This time, we pick nW primes p_1, p_2, \dots, p_{nW} , each larger than $\max\{2n, mr\}$, repeat the determinant computations over each p_i . This blows up the running time by

a factor of nW . The final reconstruction of the coefficient of x^α takes $\mathcal{O}(n^2 \cdot W^2)$ time. Hence each determinant computation for each r takes $\tilde{\mathcal{O}}(n \cdot W \cdot 2^q \cdot n^{q+2} + n^2 \cdot W^2)$ single-precision operations. The final running time is hence n times this.

Theorem 15. *Given a q -colored multidigraph G on n vertices and m edges, a weight function $w : E(G) \rightarrow [W]$ with $W \in n^{\mathcal{O}(1)}$, a vertex $s \in V(G)$ and a color-constraint $\alpha \in \mathbb{N}^{q-1}$, we can find the minimum weight of an α -colored s -arborescence in G in $\tilde{\mathcal{O}}(2^q \cdot n^{q+4} \cdot W + n^3 \cdot W^2)$ time. Moreover, we can find a minimum weight solution in $\tilde{\mathcal{O}}(m(2^q \cdot n^{q+4} \cdot W + n^3 \cdot W^2))$ time.*

5 Concluding Remarks

Though CC-ARB and EXACT PERFECT MATCHING in bipartite graphs reduce to 3-MATROID INTERSECTION, there seems to be a sharp contrast in their complexities. The former has a deterministic polynomial-time algorithm for every fixed q while the only polynomial-time algorithm known for the latter, even for $q = 2$, is randomized. CC-ARB is a special case of 3-MATROID INTERSECTION where two of the three matroids are partition matroids and the third one is a graphic matroid. EXACT PERFECT MATCHING restricted to bipartite graphs is also a special case of 3-MATROID INTERSECTION where all the three matroids are partition matroids. It is intriguing whether the hardness of the problems is dependent on this difference. A deep dive into this is tempting.

We also hope that our colored extensions of the classic matrix-tree theorems of Tutte and Kirchhoff will open another bridge between algebra and algorithms. In fact, an algorithm for CC-ARB with exponential dependence only on q can give practical algorithms to compute the symbolic determinant of a subclass of large matrices of linear polynomials when the number of indeterminates is small. From a parameterized complexity perspective, q is indeed the most natural parameter for CC-ARB and the fixed-parameter tractability of the problem is an interesting future direction.

Acknowledgements

This work is supported by SERB CRG grant number CRG/2022/007751. We are grateful to K. Murali Krishnan and Piyush P. Kurur for their comments and suggestions. We also thank the anonymous reviewer for bringing the work of Hörsch et al. [10] to our attention.

References

- [1] P. S. Ardra, Jasine Babu, Kritika Kashyap, R. Krithika, Sreejith K. Pallathumadam, and Deepak Rajendraprasad. Arborescences and shortest path trees when colors matter. *Theor. Comput. Sci.*, 1055:115486, 2025.
- [2] Francisco Barahona and William R. Pulleyblank. Exact arborescences, matchings and cycles. *Discret. Appl. Math.*, 16(2):91–99, 1987.

- [3] Carl Wilhelm Borchardt. Ueber eine der interpolation entsprechende darstellung der eliminations-resultante. *Journal für die reine und angewandte Mathematik*, pages 111–121, 1860.
- [4] Seth Chaiken and Daniel J Kleitman. Matrix tree theorems. *Journal of combinatorial theory, Series A*, 24(3):377–381, 1978.
- [5] Patrick De Leenheer. An elementary proof of a matrix tree theorem for directed graphs. *SIAM Review*, 62(3):716–726, 2020.
- [6] Reinhard Diestel. Graph theory. *Springer*, 2017.
- [7] Harold N. Gabow and Robert Endre Tarjan. Efficient Algorithms for a Family of Matroid Intersection Problems. *J. Algorithms*, 5(1):80–131, 1984.
- [8] Dan Gusfield. Matroid optimization with the interleaving of two ordered sets. *Discret. Appl. Math.*, 8(1):41–50, 1984.
- [9] Ellis Horowitz and Sartaj Sahni. On computing the exact determinant of matrices with polynomial entries. *J. ACM*, 22(1):38–50, 1975.
- [10] Florian Hörsch, András Imolay, Ryuhei Mizutani, Taihei Oki and Tamás Schwarcz. Problems on Group-Labeled Matroid Bases. In *51st International Colloquium on Automata, Languages, and Programming*, pages 86:1–86:20, 2024.
- [11] Alon Itai and Michael Rodeh. Some matching problems. In *4th Colloquium on Automata, Languages and Programming*, volume 52 of *Lecture Notes in Computer Science*, pages 258–268, 1977.
- [12] Alon Itai, Michael Rodeh, and Steven L Tanimoto. Some matching problems for bipartite graphs. *J. ACM*, 25(4):517–525, 1978.
- [13] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bull. EATCS*, 105:41–72, 2011.
- [14] Ana Marco and José-Javier Martínez. Parallel computation of determinants of matrices with polynomial entries. *Journal of Symbolic Computation*, 37(6):749–760, 2004.
- [15] John W. Moon. Some determinant expansions and the matrix-tree theorem. *Discret. Math.*, 124(1-3):163–171, 1994.
- [16] Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *19th annual ACM symposium on Theory of computing*, pages 345–354, 1987.
- [17] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987.
- [18] Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *J. ACM*, 29(2):285–309, 1982.
- [19] Franz Rendl and Matthias Leclerc. A multiply constrained matroid optimization problem. *Discret. Math.*, 73(1-2):207–212, 1989.

- [20] Alexander Schrijver. Combinatorial Optimization - Polyhedra and Efficiency. *Springer Berlin, Heidelberg*, 2003.
- [21] William Thomas Tutte. The dissection of equilateral triangles into equilateral triangles. *Math. Proc. Cambridge Phil. Soc.*, 44(4):463–482, 1948.
- [22] Jan van den Brand, Li Chen, Richard Peng, Rasmus Kyng, Yang P. Liu, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford. A Deterministic Almost-Linear Time Algorithm for Minimum-Cost Flow. In *64th IEEE Annual Symposium on Foundations of Computer Science*, pages 503–514. IEEE, 2023.
- [23] Joachim von zur Gathen and Jürgen Gerhard. Modern computer algebra. *Cambridge University Press*, 2013.