MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Counting Minimum Weight Arborescences

Koyo HAYASHI and Satoru IWATA

METR 2017–07 March 2017

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Counting Minimum Weight Arborescences *

Koyo Hayashi † Satoru Iwata ‡

March 2017

Abstract

In a directed graph $D = (V, A)$ with a specified vertex $r \in V$, an arc subset $B \subseteq A$ is called an *r-arborescence* if B has no arc entering r and there is a unique path from r to v in (V, B) for each $v \in V \backslash \{r\}$. The problem for finding a minimum weight r-arborescence in a weighted digraph has been studied for decades starting with Chu and Liu (1965), Edmonds (1967) and Bock (1971). In this paper, we focus on the number of minimum weight arborescences. We present an algorithm for counting minimum weight r-arborescences in $O(n^{\omega})$ time, where n is the number of vertices of an input digraph and ω is the matrix multiplication exponent.

1 Introduction

In a directed graph $D = (V, A)$ with a specified vertex $r \in V$, an arc subset $B \subseteq A$ is called an *r-arborescence* (or an *arborescence rooted at r*) if B has no arc entering r and there is a unique path from r to v in (V, B) for each $v \in V \backslash \{r\}$. As easily checked, a digraph D contains an r-arborescence if and only if each vertex in D is reachable from r. If D is a weighted digraph, a *minimum weight r-arborescence* is an r-arborescence whose total arc weight is minimum. Polynomial-time algorithms for finding a minimum weight r-arborescence were discovered independently by Chu and Liu [3], Edmonds [4] and Bock [1]. The best known bound for this problem has been obtained by Gabow et al. [6]. Their algorithm runs in $O(m + n \log n)$ time, where n and m are the numbers of vertices and arcs of an input digraph, respectively.

The above algorithms, however, find at most one minimum weight r-arborescence, while a digraph might contain more than one. In this paper, we focus on the multiplicity of optimal solutions. More specifically, we consider the following problem:

> Given a directed graph $D = (V, A)$ with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$, find the number of minimum weight r-arborescences in D, (1)

where $\mathbb{R}_+$ is the set of nonnegative real numbers. If w is a uniform weight, in particular, this problem is easy. All we have to do in this case is to compute the number of r-arborecences in D. This can be done by applying the following theorem, which is commonly known as the Matrix Tree Theorem. See, e.g., [7, Problem 4.16] for its proof.

Theorem 1.1 (Matrix Tree Theorem) *Let $D = (V, A)$ be a directed graph. Let a_{ij} denote the number of arcs leaving i and entering j for any two distinct vertices $i, j \in V$. Define the $V \times V$ matrix $L = (l_{ij})$ by*

$$l_{ij} := \begin{cases} \sum_{k \neq j} a_{kj} & (i = j), \\ -a_{ij} & (\textit{otherwise}). \end{cases} \tag{2}$$

Then, for each vertex $i \in V$, the number of arborescences in D rooted at i is equal to $\det L_i$, where L_i is the submatrix obtained by deleting the i-th row and column from L. ■

*This work is supported by CREST, JST.

†Department of Mathematical Informatics, University of Tokyo, Tokyo 113-8656, Japan. E-mail: hayashi-koyo928@g.ecc.u-tokyo.ac.jp

‡Department of Mathematical Informatics, University of Tokyo, Tokyo 113-8656, Japan. E-mail: iwata@mist.i.u-tokyo.ac.jp

By Theorem 1.1, one can compute the number of minimum weight r-arborescences in a uniformly weighted digraph in $O(n^\omega)$ time, where ω is the matrix multiplication exponent ($2 < \omega < 3$), i.e., the number of elementary operations needed to multiply two $n \times n$ matrices is $O(n^\omega)$. Although problem (1) is not so simple for an arbitrary w, we show that one can solve it within the same asymptotic running time based on the method of Fulkerson [5].

The problem for finding a minimum weight r-arborescence can be formulated as an integer program, which can be relaxed to the following linear program:

$$\begin{aligned} \text{Minimize} \quad & \sum_{a \in A} w(a)x(a) \\ \text{subject to} \quad & \sum_{a \in \delta^-(U)} x(a) \geq 1 \qquad (U \subseteq V \setminus \{r\}), \\ & x(a) \geq 0 \qquad (a \in A), \end{aligned} \tag{LP}$$

where $\delta^-(U)$ denotes a set of arcs entering U. The dual of (LP) can be described as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{U \subseteq V \setminus \{r\}} y(U) \\ \text{subject to} \quad & \sum_{\substack{U \subseteq V \setminus \{r\}: \\ a \in \delta^-(U)}} y(U) \leq w(a) \qquad (a \in A), \\ & y(U) \geq 0 \qquad (U \subseteq V \setminus \{r\}). \end{aligned} \tag{DP}$$

Fulkerson [5] gave an algorithm for finding an optimal solution of (DP). This algorithm yields as a byproduct an arc subset $A^\circ \subseteq A$ and a collection $\mathcal{F} \subseteq 2^{V \setminus \{r\}}$ such that an r-arborescence B in D is of minimum weight if and only if

$$B \subseteq A^\circ \text{ and } |B \cap \delta^-(U)| = 1 \text{ for each } U \in \mathcal{F}. \tag{3}$$

This condition comes from the complementary slackness between (LP) and (DP).

To solve our problem (1), it suffices to count r-arborescences that satisfy (3). Actually, such counting can be done in $O(n^\omega)$ time. A key observation that leads to this bound is that:

> Given an unweighted strongly connected digraph $D = (V, A)$, one can determine the numbers of arborescences in D rooted at v for all $v \in V$ simultaneously in $O(n^\omega)$ time. (4)

We also give an efficient implementation of Fulkerson's algorithm that runs in $O(n^2 + m \log n)$ time, which enables us to solve our problem (1) in $O(n^\omega)$ time as a whole.

A similar problem for spanning trees in an undirected graph has already been considered by Broder and Mayr [2]. They devised an algorithm for counting minimum weight spanning trees in an undirected graph in $O(n^\omega)$ time.

The rest of this paper is organized as follows. In Section 2, we explain Fulkerson's algorithm. In Section 3, we give an $O(n^2 + m \log n)$-time implementation of Fulkerson's algorithm. Section 4 is devoted to proving (4). In Section 5, we give an algorithm for counting r-arborescences that satisfy (3) in $O(n^\omega)$ time to conclude that one can solve our problem (1) in $O(n^\omega)$ time.

2 Fulkerson's algorithm

Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$. We assume that every vertex in D is reachable from r and that no arc of A enters r. In this section, we explain Fulkerson's algorithm [5] for finding an optimal solution y of (DP). An arc $a \in A$ is said to be *tight* for a feasible solution y of (DP) if

$$\sum_{U \subseteq V \setminus \{r\}:\, a \in \delta^-(U)} y(U) = w(a). \tag{5}$$

Fulkerson's algorithm can be described as follows.

Algorithm Fulkerson

⟨**1**⟩ Set $A^\circ := \{a \in A \mid w(a) = 0\}$ and $y := 0$.

⟨**2**⟩ Iterate the following until every vertex is reachable in $D^\circ = (V, A^\circ)$ from r.

⟨**2-1**⟩ Find a strong component U of the digraph D° with $r \notin U$ and $A^\circ \cap \delta^-(U) = \emptyset$.

⟨**2-2**⟩ Increase $y(U)$ as much as possible until some arc $a \in \delta^-(U)$ gets tight for y.

⟨**2-3**⟩ Set $A^\circ := A^\circ \cup \{a \in \delta^-(U) \mid a \text{ is tight for } y\}$.

Let y and A° be those obtained at the end of the above algorithm and set $D^\circ := (V, A^\circ)$. Note that an arc a belongs to A° if and only if a is tight for y. Let $\mathcal{F}$ be a collection of vertex sets $U \subseteq V \backslash \{r\}$ with $y(U) > 0$.

For a vertex subset $U \subseteq V$, we denote by $D^\circ[U]$ the subgraph of D° induced by U. It is easy to see that $D^\circ[U]$ is strongly connected for each $U \in \mathcal{F}$, and that $\mathcal{F}$ is laminar, i.e., $U \subseteq W$, $W \subseteq U$ or $U \cap W = \emptyset$ for all $U, W \in \mathcal{F}$. Then we can take an r-arborescence B in D such that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$, by taking $D = D^\circ$ in the following lemma.

Lemma 2.1 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and let $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ be a laminar family. Suppose that every vertex is reachable in D from r and $D[U]$ is strongly connected for each $U \in \mathcal{F}$. Then there exists an r-arborescence B in D such that $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.*

Proof. By induction on $|\mathcal{F}|$. We may assume that $\mathcal{F}$ contains no singleton. The case $\mathcal{F} = \emptyset$ being trivial, suppose that $|\mathcal{F}| \geq 1$. Let U be an inclusion-wise minimal set in $\mathcal{F}$. Shrink U to a single vertex u, obtaining a new digraph D'. Similarly, set $\mathcal{F}' := \{(W \backslash U) \cup \{u\} \mid U \subsetneq W \in \mathcal{F}\} \cup \{W \mid W \cap U = \emptyset,\ W \in \mathcal{F}\}$. Since $|\mathcal{F}'| = |\mathcal{F}| - 1$, induction gives an r-arborescence B' in D' such that $|B' \cap \delta^-_{D'}(W)| = 1$ for each $W \in \mathcal{F}'$. Expanding u to the original vertex set U, extend B' to an r-arborescence B in D (such an r-arborescence exists since $D[U]$ is strongly connected). Then B satisfies that $|B \cap \delta^-_D(W)| = 1$ for each $W \in \mathcal{F}$. ■

Choose any r-arborescence B in D such that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$. Now let us show that y is an optimal solution of (DP) and B is a minimum weight r-arborescence. To prove this, let χ^B be the incidence vector of B: namely we let $\chi^B(a)$ to be 1 for $a \in B$ and 0 otherwise. Then χ^B and y are optimal solutions of (LP) and (DP), respectively, by the complementary slackness. Indeed, if $\chi^B(a) > 0$ for some $a \in A$, then a is tight for y (as $a \in B \subseteq A^\circ$); If $y(U) > 0$ for some $U \subseteq V \backslash \{r\}$, then we have $\chi^B(\delta^-(U)) = 1$ (as $U \in \mathcal{F}$ and $|B \cap \delta^-(U)| = 1$). Hence y is optimal and B is of minimum weight.

Conversely, any minimum weight r-arborescence B in D satisfies that $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$. To see this, let B^* be a minimum weight r-arborescence in D and let χ^{B^*} be the incidence vector of B^*. Then χ^{B^*} is an optimal solution of (LP). Since y is optimal as well, we have the following by the complementary slackness: For each $a \in B^*$, a is tight for y (as $\chi^{B^*}(a) > 0$); For each $U \in \mathcal{F}$, $\chi^{B^*}(\delta^-(U))$ is equal to 1 (as $y(U) > 0$). Hence we have $B^* \subseteq A^\circ$ and $|B^* \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.

As a consequence we have the following well-known proposition.

Proposition 2.2 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$ such that every vertex in D is reachable from r. Then there exists an arc set $A^\circ \subseteq A$ and a laminar family $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ such that an r-arorescence B in D is of minimum weight if and only if $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.* ■

3 An efficient implementation of Fulkerson's algorithm

In this section, we give an implementation of Fulkerson's algorithm that runs in $O(n^2 + m \log n)$ time. Fulkerson's algorithm can be redescribed as follows.

Algorithm Fulkerson*

⟨**1**⟩ Set $A^\circ := \emptyset$ and $y := 0$.
 Set $z(v) := 0$ for each $v \in V \backslash \{r\}$.

⟨**2**⟩ Iterate the following until every vertex is reachable in $D^\circ = (V, A^\circ)$ from r.

 ⟨**2-1**⟩ Find a strong component U of the digraph D° with $r \notin U$ and $A^\circ \cap \delta^-(U) = \emptyset$.

 ⟨**2-2**⟩ Set $\mu := \min\{w(a) - z(v) \mid a = (u, v) \in \delta^-(U)\}$.
 Set $y(U) := \mu$.
 Set $z(v) := z(v) + \mu$ for each $v \in U$.
 Set $A^\circ := A^\circ \cup \{a \in \delta^-(U) \mid w(a) - z(v) = 0\}$.

Throughout the iterations, $z(v)$ is equal to the sum of all positive $y(W)$ with $v \in W \subseteq V \backslash \{r\}$. Note that the case $\mu = 0$ may occur in ⟨**2-2**⟩ if U is a singleton (as we set $A^\circ := \emptyset$ initially).

Let us consider the running time bound. First note that there are at most $2n$ iterations. This comes from that the collection of vertex sets U chosen in ⟨**2-1**⟩ is laminar.

Next we consider the running time of ⟨**2-2**⟩. This part can be done in $O(n^2 + m)$ time throughout all iterations, if we initially sort arcs of $\delta^-(v)$ for each $v \in V \backslash \{r\}$ such that they are in increasing order with respect to w. (Indeed, A° never decreases throughout the iterations.) This sorting can be done in $O(m \log n)$ time. Hence we can perform ⟨**2-1**⟩ (including sorting) in $O(n^2 + m \log n)$ time throughout all iterations.

Now let us consider the running time of ⟨**2-1**⟩. A naive way for this part could require $O(m + n)$ time at each iteration, which amounts to $O(nm)$ time as a whole. Actually, one can do ⟨**2-1**⟩ in $O(n)$ time at each iteration. To show this, we introduce a certain concept for reachability.

Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$. We say that D is *r-harmonious* if there exist functions $\phi : V \to \{0, 1, 2\}$ and $\theta : V \to \mathbb{Z}_+$ such that:

(i) a vertex $v \in V$ is reachable from r if and only if $\phi(v) = 0$;
(ii) if $\phi(v) = 2$, then $\delta^-(v) = \emptyset$;
(iii) for any $u, v \in V$ with $\phi(u) \geq 1$ and $\phi(v) = 1$, v is reachable from u if and only if $\theta(u) \geq \theta(v)$; (6)
(iv) if D contains no r-arborescence, then there exists a strong component K of D such that $\delta^-(K) = \emptyset$ and $\phi(v) = 1$ for all $v \in K$.

If such functions ϕ and θ exist, we say that ϕ is a *color* for D and θ is a *label* for D. See Figure 1 for an example of an r-harmonious digraph.

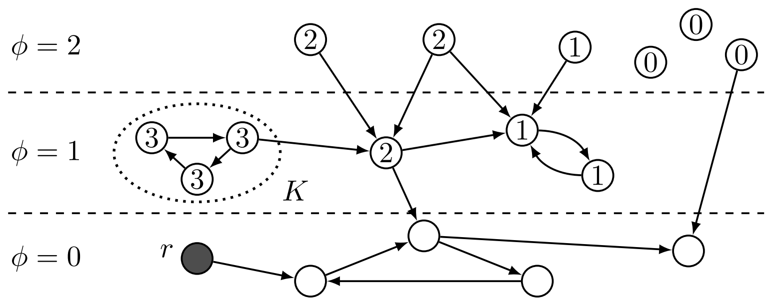

Figure 1: An r-harmonious digraph. Numbers assigned to vertices represent their values of the label.

Let us check some facts on an r-harmonious digraph $D = (V, A)$ with a color ϕ and a label θ. Suppose that D contains no r-arborescence, and let K be a strong component of D such that $\delta^-(K) = \emptyset$ and $\phi(v) = 1$ for all $v \in K$. Then each vertex v with $\phi(v) = 1$ is reachable from any vertex of K. (Indeed, if $\theta(u) < \theta(v)$ for some $u \in K$ and some vertex v with $\phi(v) = 1$, then u is reachable from v but v is not reachable from u, which contradicts that K is a strong component of D with $\delta^-(K) = \emptyset$.) This also implies the uniqueness of K. So we let $K(D)$ denote the strong component K of D for any r-harmonious digraph D that contains no r-arborescence.

Now we are ready to prove the following lemma, which will be used to implement **(2-1)** efficiently.

Lemma 3.1 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$. Suppose that D is r-harmonious with a color ϕ and a label θ and that D contains no r-arborescence. When adding to D some arcs entering $K(D)$, one can find a color and a label for the new digraph in $O(n)$ time. In particular, the new digraph is r-harmonious.*

Proof. Let F be the set of arcs entering $K(D)$ that have been added to D, and set $T := \{u \mid (u, v) \in F\}$. Let D' denote the new digraph. Clearly, (6)(ii) is maintained. We consider two cases.

Case 1: $T \cap \phi^{-1}(0) \neq \emptyset$. In this case, every vertex v with $\phi(v) = 1$ becomes reachable in D' from r (as $K(D)$ is reachable in D' from r). Set $\phi(v) := 0$ for each vertex v with $\phi(v) = 1$, and set $\theta(v) := 0$ for all $v \in V$. This maintains condition (6)(i), (ii) and (iii). If D' contains no r-arborescence, choose one vertex s with $\phi(s) = 2$, and set $\phi(s) := 1$ and $\theta(s) := 1$. This maintains (6). (In fact, $K(D') = \{s\}$.)

Case 2: $T \cap \phi^{-1}(0) = \emptyset$. Clearly, (6)(i) is maintained. Let k be the value of θ on $K(D)$. (Note that θ takes the same value over $K(D)$.) Let j be the minimum value of θ over $K(D) \cup \{v \in T \mid \phi(v) = 1\}$. To restore (6)(iii), we do the following:

$$\text{Set } \theta(v) := j \text{ for all } v \in \{u \in V \mid \phi(u) \geq 1,\, j \leq \theta(u) \leq k\} \cup \{u \in T \mid \phi(u) = 2\}. \tag{7}$$

This maintains (6)(iii), since the set of vertices u with $j \leq \theta(u) \leq k$ and $\phi(u) = 1$ is a strong component of D'.

If $\theta(v)$ is less than j for any vertex v with $\phi(v) = 2$ after doing (7), then (6)(iv) is also maintained. (In fact, $K(D') = \{u \in V \mid \phi(u) = 1,\, \theta(u) = j\}$.) If there is a vertex s with $\phi(s) = 2$ and $\theta(s) = j$ after doing (7), we set $\phi(s) := 1$ and $\theta(s) := j + 1$. This maintains (6). (In fact, $K(D') = \{s\}$.)

It is not difficult to see that these operations can be done in $O(n)$ time. ■

Lemma 3.1 implies that ⟨**2-1**⟩ can be performed efficiently by keeping $D^\circ = (V, A^\circ)$ r-harmonious throughout the iterations. So we have the following result.

Theorem 3.2 *Fulkerson's algorithm can be implemented to run in $O(n^2 + m \log n)$ time.*

Proof. Since we have already observed that ⟨**2-2**⟩ can be done in $O(n^2 + m \log n)$ time throughout all iterations, it suffices to show that one can perform ⟨**2-1**⟩ in $O(n)$ time at each iteration.

At the start of the algorithm, do the following: Choose one vertex $s \in V \backslash \{r\}$ arbitrarily, and set $\phi(r) := 0$, $\phi(s) := 1$ and $\phi(v) := 2$ for all $v \in V \backslash \{r, s\}$; Set $\theta(s) := 1$ and set $\theta(v) := 0$ for all $v \in V \backslash \{s\}$. Then $D^\circ = (V, A^\circ)$ is initially an r-harmonious digraph with a color ϕ and a label θ (as $A^\circ = \emptyset$). If we choose $K(D^\circ)$ in ⟨**2-1**⟩ at each iteration, we can restore ϕ and θ in $O(n)$ time for the new digraph D° obtained after ⟨**2-2**⟩ at its iteration by Lemma 3.1. This implies that one can perform ⟨**2-1**⟩ in $O(n)$ time at each iteration. ■

Directly from Theorem 3.2 and discussion in Section 2, we have the following corollary.

Corollary 3.3 *Let $D = (V, A)$ be a digraph with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$ such that every vertex in D is reachable from r. Then one can find in $O(n^2 + m \log n)$ time an arc set $A^\circ \subseteq A$ and a laminar family $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ such that an r-arorescence B in D is of minimum weight if and only if $B \subseteq A^\circ$ and $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.* ■

4 Simultaneous counting of rooted arborescences

Let $D = (V, A)$ be an unweighted digraph with n vertices. For each $v \in V$, let $\tau(v)$ denote the number of arborescences in D rooted at v. Theorem 1.1 tells us that one can find $\tau(v)$ in $O(n^\omega)$ time for each vertex v. To find $\tau(v)$ for all $v \in V$, we must determine all the diagonal cofactors of the matrix L defined by (2). If L were nonsingular, this could be done by computing its inverse in $O(n^\omega)$ time. Unfortunately, however, L is in fact singular, and a naive way requires $O(n^\omega \cdot n)$ time. In this section, we show that one can determine $\tau(v)$ for all $v \in V$ in $O(n^\omega)$ time if D is strongly connected.

Let $H = (h_{ij})$ be an $n \times n$ matrix satisfying the following condition:

$$\sum_{i=1}^{n} h_{ij} = 0 \quad \text{for } j = 1, 2, \ldots, n. \tag{8}$$

So the sum of each column of H is equal to zero. Let H_i denote the submatrix obtained by deleting the i-th row and column from H for $i = 1, 2, \ldots, n$. Then H can be partitioned as

$$H = \begin{pmatrix} \alpha & \beta \\ \gamma & \eta \end{pmatrix}, \tag{9}$$

where $\alpha = H_n$. If α is nonsingular, then each $\det H_i$ can be written as follows.

Lemma 4.1 *If H satisfies (8) and $\alpha = H_n$ is nonsingular, then*

$$\det H_i = -(\alpha^{-1}\beta)_i \cdot \det \alpha \tag{10}$$

for $i = 1, 2, \ldots n - 1$. Here $(\alpha^{-1}\beta)_i$ means the i-th component of the vector $\alpha^{-1}\beta$.

Proof. Let i be an integer from 1 to $n - 1$. Define

$$P := \begin{pmatrix} \alpha^{-1} & 0 \\ \lambda & 1 \end{pmatrix}, \tag{11}$$

where λ is a row vector of dimension $n - 1$ with all entries equal to 1. Then we have

$$PH = \begin{pmatrix} I & \alpha^{-1}\beta \\ 0 & 0 \end{pmatrix}, \tag{12}$$

where I is the identity matrix of dimension $n-1$. Let e_i denote the i-th unit vector, and let G_i denote a matrix arising from H by replacing the i-th column of H with e_i. It follows from (12) that

$$\det(PG_i) = \det \begin{pmatrix} (Pe_i)_i & (\alpha^{-1}\beta)_i \\ (Pe_i)_n & 0 \end{pmatrix} = -(\alpha^{-1}\beta)_i. \tag{13}$$

This implies the lemma, since $\det(PG_i) = \det P \det G_i = \det H_i / \det \alpha$. ■

Since both $\det \alpha$ and the vector $\alpha^{-1}\beta$ can be determined in $O(n^\omega)$ time, Lemma 4.1 implies that all the diagonal cofactors of the matrix H can be computed in $O(n^\omega)$ time. This immediately yields the following result.

Theorem 4.2 *Given a strongly connected digraph $D = (V, A)$, one can determine $\tau(v)$ for all $v \in V$ in $O(n^\omega)$ time, where $\tau(v)$ is the number of arborescences in D rooted at v.*

Proof. Define the matrix L by (2). Note that the sum of each column of L is equal to zero. Since D is strongly connected, $\det L_i$ is positive for each $i \in V$ by Theorem 1.1. This implies, in particular, that L_i is nonsingular for any $i \in V$. Hence, directly from Lemma 4.1 and Theorem 1.1, we obtain the theorem. ■

5 Counting minimum weight arborescences

Proposition 2.2 reduces our problem (1) to a problem for counting arborescences satisfying certain conditions in an unweighted digraph. Recall that $D[U]$ denotes the subgraph of D induced by U. We now consider the following problem:

Given: a digraph $D = (V, A)$ with a specified vertex $r \in V$ and a laminar family $\mathcal{F} \subseteq 2^{V \setminus \{r\}}$ such that every vertex is reachable in D from r and $D[U]$ is strongly connected for each $U \in \mathcal{F}$; (14)
Find: the number of r-arborescences B in D such that $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$.

In this section, we show that one can solve the above problem (14) in $O(n^\omega)$ time. With Corollary 3.3, this implies that our problem (1) can be solved in $O(n^\omega)$ time.

5.1 Outline

We may assume that $\mathcal{F}$ contains no singleton, since any r-arborescence in D enters each vertex $v \neq r$ exactly once. We say that an r-arborescence B in D is *$\mathcal{F}$-tight* if $|B \cap \delta^-(U)| = 1$ for each $U \in \mathcal{F}$. A key observation for counting $\mathcal{F}$-tight r-arborescences in D is that:

$$\text{For any } \mathcal{F}\text{-tight } r\text{-arborescence } B \text{ in } D \text{ and for any } U \in \mathcal{F},\ B[U] \text{ is an arborescence in } D[U], \tag{15}$$

where $B[U]$ denotes a set of arcs of B spanned by U.

Now we give an useful idea that yields an efficient method for counting $\mathcal{F}$-tight r-arborescences in D. Let U be an inclusion-wise minimal set in $\mathcal{F}$. For each $v \in U$, let $\tau_U(v)$ denote the number of arborescences in $D[U]$ rooted at v. Since $D[U]$ is strongly connected, $\tau_U(v)$ is positive for each $v \in U$. Let D_U be a digraph arising from D by doing the following operations:

$$\text{For each arc } a = (s, v) \in \delta^-(U), \text{ replace } a \text{ by } \tau_U(v) \text{ parallel arcs; Shrink } U \text{ to a new vertex } u. \tag{16}$$

Similarly, let $\mathcal{F}_U$ be a collection obtained from $\mathcal{F}$ by shrinking U. More precisely, set $\mathcal{F}_U := \{(W \backslash U) \cup \{u\} \mid U \subsetneq W \in \mathcal{F}\} \cup \{W \mid W \cap U = \emptyset,\ W \in \mathcal{F}\}$. Then we have the following.

Claim 5.1 *The number of $\mathcal{F}$-tight r-arborescences in D is equal to that of $\mathcal{F}_U$-tight r-arborescences in D_U.*

Proof. Let D' be a digraph obtained from D by shrinking U to one new vertex u (without replicating arcs). To avoid complication, let $\delta^-(U)$ and $\delta^-(u)$ denote the arc sets $\delta^-_D(U)$ and $\delta^-_{D'}(u)$, respectively, and identify them as the same set. For each $a \in \delta^-(U)$, let $\partial^- a$ denote the head of a in D. So $\partial^- a \in U$.

For each arc $a \in \delta^-(u)$, let $\sigma(a)$ denote the number of $\mathcal{F}_U$-tight r-arborescences in D' containing a. Then the number of $\mathcal{F}$-tight r-arborescences in D containing a is equal to $\sigma(a) \cdot \tau_U(\partial^- a)$ for each arc $a \in \delta^-(U)$, by (15) and the minimality of U. Hence the total number of $\mathcal{F}$-tight r-arborescences in D is equal to $\sum_{a \in \delta^-(U)} \sigma(a) \cdot \tau_U(\partial^- a)$, which implies the claim. ■

We can derive from Claim 5.1 an efficient method for solving problem (14). Note that $|\mathcal{F}_U| = |\mathcal{F}| - 1$. Resetting $D := D_U$ and $\mathcal{F} := \mathcal{F}_U$ and iterating the series of the operations, we will get $\mathcal{F} = \emptyset$ at some point. Claim 5.1 implies that throughout the iterations the number of $\mathcal{F}$-tight r-arborescences in D does not change. If $\mathcal{F}$ is empty, the number of $\mathcal{F}$-tight r-arborescences in D is nothing but that of r-arborescences in D, which can be determined by just applying Theorem 1.1.

5.2 Algorithm description and complexity

Now we develop an algorithm for problem (14). Let $D = (V, A)$ be a digraph with vertex set $V = \{1, 2, \ldots, n\}$ that contains a specified vertex $r \in V$, and let $\mathcal{F} \subseteq 2^{V \backslash \{r\}}$ be a laminar family. Suppose that D contains an r-arborescence and that $D[U]$ is strongly connected for each $U \in \mathcal{F}$. We may assume that $\mathcal{F}$ contains no singleton. Moreover, we may assume that the laminar family $\mathcal{F} = \{U_k\}_{k=1}^t$ satisfies that $U_i \subsetneq U_j$ or $U_i \cap U_j = \emptyset$ for any $1 \leq i < j \leq t$, since the members of $\mathcal{F}$ can be found in such an order by Fulkerson's algorithm. Let a_{ij} denote the number of arcs leaving i and entering j for any two distinct vertices $i, j \in V$. The counting algorithm can be described as follows.

ALGORITHM COUNTING

⟨1⟩ Set $\psi(v) := v$ for each vertex $v \in V = \{1, 2, \ldots, n\}$. Set $q := n + 1$.

⟨2⟩ For $k = 1, 2, \ldots, t$, do the following.

⟨2-1⟩ Set $I := \{\psi(v) \mid v \in U_k\}$ and $J := \{\psi(v) \mid v \in V \backslash U_k\}$. Define the $I \times I$ matrix $L = (l_{ij})$ by

$$l_{ij} := \begin{cases} \sum_{p \in I \backslash \{i\}} a_{pj} & (i = j), \\ -a_{ij} & \text{(otherwise)}. \end{cases} \tag{17}$$

Determine $\tau(i) := \det L_i$ for each $i \in I$, where L_i is the submatrix obtained by deleting i-th row and column from L.

⟨2-2⟩ Set $a_{jq} := \sum_{i \in I} a_{ji} \cdot \tau(i)$ and $a_{qj} := \sum_{i \in I} a_{ij}$ for each $j \in J$.

⟨**2-3**⟩ Set $\psi(v) := q$ for each $v \in U_k$. Set $q := q + 1$.

⟨**3**⟩ Set $I := \{\psi(v) \mid v \in V\}$. Define the $I \times I$ matrix $L = (l_{ij})$ by (17). Return $\det L_{\psi(r)}$.

Let us consider the running time bound. For $k = 1, 2, \ldots, t$, let d_k and n_k be the sizes of I and $I \cup J$ in ⟨**2-1**⟩ at k-th iteration, respectively. So $n_1 = n$. Note that each d_k is larger than 1 (since we suppose that $\mathcal{F}$ contains no singleton). It is easy to see that $n_{k+1} = n_k - d_k + 1$ for $k = 1, 2, \ldots, t-1$. This gives that $t \leq n$ and $\sum_{k=1}^{t} d_k \leq 2n$. Since we can do ⟨**2-1**⟩ in $O({d_k}^\omega)$ time at k-th iteration by Theorem 4.2, we can perform ⟨**2-1**⟩, ⟨**2-2**⟩ and ⟨**2-3**⟩ at k-th iteration in time

$$O({d_k}^\omega + d_k(n_k - d_k)) \leq O({d_k}^\omega + nd_k). \tag{18}$$

Hence, throughout all iterations, we can perform ⟨**2**⟩ in time

$$O\left(\sum_{k=1}^{t}({d_k}^\omega + nd_k)\right) \leq O\left(\left(\sum_{k=1}^{t} d_k\right)^\omega + n^2\right) \leq O(n^\omega). \tag{19}$$

Also we can do ⟨**3**⟩ in $O(n^\omega)$ time. Therefore, problem (14) can be solved in $O(n^\omega)$ time, which together with Corollary 3.3, implies the following theorem.

Theorem 5.2 *Given a directed graph $D = (V, A)$ with a specified vertex $r \in V$ and a weight function $w : A \to \mathbb{R}_+$, one can find the number of minimum weight r-arborescences in D in $O(n^\omega)$ time.* ■

References

[1] F. Bock: An algorithm to construct a minimum directed spanning tree in a directed network. In *Developments in Operations Research*, Gordon and Breach, New York, 1971, 29–44.

[2] A. Z. Broder and E. W. Mayr: Counting minimum weight spanning trees. *Journal of Algorithms*, **24** (1997), 171–176.

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

[4] J. Edmonds: Optimal branchings. *Journal of Research of the National Bureau of Standards*, **71B** (1967), 233–240.

[5] D. R. Fulkerson: Packing rooted directed cuts in a weighted directed graph. *Mathematical Programming*, **6** (1974), 1–13.

[6] H. N. Gabow, Z. Galil, T. Spencer and R. E. Tarjan: Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatrica*, **6** (1986), 109–122.

[7] L. Lovász: *Combinatorial Problems and Exercises: Second edition*, AMS Chelsea Publishing, Providence, 2007.