RANKING TOP-\(k\) TREES IN TREE-BASED PHYLOGENETIC NETWORKS

MOMOKO HAYAMIZU\(^1,2\) AND KAZUHISA MAKINO\(^3\)

ABSTRACT. 'Tree-based' phylogenetic networks proposed by Francis and Steel have attracted much attention of theoretical biologists in the last few years. At the heart of the definitions of tree-based phylogenetic networks is the notion of 'support trees', about which there are numerous algorithmic problems that are important for evolutionary data analysis. Recently, Hayamizu [arXiv:1811.05849 [math.CO]] proved a structure theorem for tree-based phylogenetic networks and obtained linear-time and linear-delay algorithms for many basic problems on support trees, such as counting, optimisation, and enumeration. In the present paper, we consider the following fundamental problem in statistical data analysis: given a tree-based phylogenetic network \(N\) whose arcs are associated with probability, create the top-\(k\) support tree ranking for \(N\) by their likelihood values. We provide a linear-delay (and hence optimal) algorithm for the problem and thus reveal the interesting property of tree-based phylogenetic networks that ranking top-\(k\) support trees is as computationally easy as picking \(k\) arbitrary support trees.

I. INTRODUCTION

Although phylogenetic trees have been used as the standard model of evolution, phylogenetic networks have become popular amongst biologists as a tool to describe conflicting signals in data or uncertainty in evolutionary histories [4][5][9]. Therefore, when we wish to reconstruct the phylogenetic tree \(T\) on a set \(X\) of species from non-tree-like data, a natural idea would be to describe the data using a phylogenetic network \(N = (V, A)\) on \(X\) and then remove extra arcs to discover an embedding \(\tau = (V, S)\) of \(T\) inside \(N\), where \(\tau\) is called a 'support tree' of \(N\) [6].

However, the above strategy only makes sense when \(N\) is 'tree-based', namely, \(N\) is merely a tree with additional arc [6], which is not always the case [12]. In [6], Francis and Steel provided a linear-time algorithm for finding a support tree of \(N\) if \(N\) is tree-based and reporting that it does not exist otherwise. Another linear-time algorithm for this decision problem was obtained by Zhang in [13].

While Francis and Steel’s work was followed by many studies (e.g., [1][2][4][5][7][11][13]), Hayamizu’s recent work [8] significantly advanced our understanding of how tree-based networks could be useful in contemporary phylogenetic analysis. In fact, Hayamizu’s structure theorem has derived a series of linear-time and linear-delay algorithms for many basic problems (e.g., counting, enumeration and optimisation) on support trees, and has thus enabled various data analysis using tree-based phylogenetic networks (see [8] for details).

In the present paper, we consider a so-called 'top-\(k\) ranking problem', with the aim to further facilitate the application of tree-based phylogenetic networks. The problem is as follows: given a tree-based phylogenetic network \(N\) where each arc \(a\) exists in the true evolutionary lineage with probability \(w(a) > 0\), list top-\(k\) support trees of \(N\) in non-increasing order by their likelihood values. We note that this problem is an important generalisation of the top-1 ranking problem, which asks for a maximum likelihood support tree of \(N\) and can be solved in linear time [6], since nearly optimal support trees can provide more biological insights than the maximum likelihood one.

At first glance, ranking top-\(k\) support trees may seem more difficult than picking \(k\) arbitrary support trees, the latter of which is possible with linear delay [8]; however, in this paper, we provide a linear-delay (i.e., optimal) algorithm for the top-\(k\) ranking problem and thus reveal that the above two problems have the same time complexity, which is an interesting property of tree-based phylogenetic networks.
2. Preliminaries

Throughout this paper, $X$ represents a non-empty finite set of present-day species. All graphs considered here are finite, simple, directed acyclic graphs. For a graph $G$, $V(G)$ and $A(G)$ denote the sets of vertices and arcs of $G$, respectively. A graph $G$ is called a subgraph of a graph $H$ if both $V(G) \subseteq V(H)$ and $A(G) \subseteq A(H)$ hold, in which case we write $G \subseteq H$. When $G \subseteq H$ but $G \not\subseteq H$, then $G$ is called a proper subgraph of $H$. When $G \subseteq H$ and $V(G) = V(H)$, $G$ is a spanning subgraph of $H$. Given a graph $G$ and a non-empty subset $A'$ of $A(G)$, $A'$ is said to induce the subgraph $G[A']$ of $G$, that is, the one whose arc-set is $A'$ and whose vertex-set consists of all ends of arcs in $A'$. For a graph $G$ with $|A(G)| \geq 1$ and a partition $\{A_1, \ldots, A_d\}$ of $A(G)$, the collection $\{G[A_1], \ldots, G[A_d]\}$ of arc-induced subgraphs of $G$ is called a decomposition of $G$. For an arc $a = (u, v) \in A(G)$, $u$ and $v$ are called the tail and head of $a$ and are denoted by $\text{tail}(a)$ and $\text{head}(a)$, respectively. For a vertex $v$ of a graph $G$, the in-degree of $v$ in $G$, denoted by $\text{deg}^+_G(v)$, is defined to be the cardinality of the set $\{a \in A(G) \mid \text{head}(a) = v\}$. The out-degree of $v$ in $G$, denoted by $\text{deg}^-_G(v)$, is defined in a similar manner. For any graph $G$, a vertex $v \in V(G)$ with $(\text{deg}^-_G(v), \text{deg}^+_G(v)) = (1, 0)$ is called a leaf of $G$.

Definition 2.1. A rooted binary phylogenetic $X$-network is defined to be a finite simple directed acyclic graph $N$ with the following properties:

1. $N$ has a unique vertex $\rho$ with $\text{deg}^-_N(\rho) = 0$ and $\text{deg}^+_N(\rho) \in \{1, 2\}$;
2. $X$ is the set of leaves of $N$;
3. for any $v \in V(N) \setminus \{\rho\}$, $(\text{deg}^-_N(v), \text{deg}^+_N(v)) = (1, 2)$ holds.

In Definition 2.1, the vertex $\rho$ is called the root of $N$, and a vertex $v \in V(N)$ with $(\text{deg}^-_N(v), \text{deg}^+_N(v)) = (2, 1)$ is called a reticulation vertex of $N$. When $N$ has no reticulation vertex, $N$ is called a rooted binary phylogenetic $X$-tree.

Definition 2.2 ([6]). If a rooted binary phylogenetic $X$-network $N$ that has a spanning tree $T$ can be obtained by inserting zero or more vertices into each arc of a rooted binary phylogenetic $X$-tree $T$, then $N$ is said to be tree-based and $T$ is called a support tree of $N$.

Theorem 2.3 ([8]). Let $N$ be a rooted binary phylogenetic $X$-network and let $S$ be a subset of $A(N)$. Then, the subgraph $N[S]$ of $N$ is a support tree of $N$ if and only if $S$ satisfies the following three conditions, in which case $S$ is called an admissible arc-set of $N$. Moreover, there exists a one-to-one correspondence between support trees of $N$ and admissible arc-sets of $N$:

1. $S$ contains all $(u, v) \in A(N)$ with $\text{deg}^-_N(v) = 1$ or $\text{deg}^+_N(u) = 1$.
2. for any $a_1, a_2 \in A(N)$ with $\text{head}(a_1) = \text{head}(a_2)$, exactly one of $\{a_1, a_2\}$ is in $S$.
3. for any $a_1, a_2 \in A(N)$ with $\text{tail}(a_1) = \text{tail}(a_2)$, at least one of $\{a_1, a_2\}$ is in $S$.

In this paper, as the conditions in Theorem 2.3 still make sense for any subgraph of $N$, we consider admissible arc-sets of subgraphs of $N$.

3. Known results: the structure of support trees

Here, we summarise without proofs the relevant material in [8]. A connected subgraph $Z$ of a tree-based phylogenetic $X$-network $N$ with $|A(Z)| \geq 1$ is called a zig-zag trail (in $N$) if there exists a permutation $(a_1, \ldots, a_m)$ of $A(Z)$ such that for each $i \in [1, m - 1]$, either $\text{head}(a_i) = \text{head}(a_{i+1})$ or $\text{tail}(a_i) = \text{tail}(a_{i+1})$ holds. Then, any zig-zag trail $Z$ in $N$ is specified by an alternating sequence of (not necessarily distinct) vertices and distinct arcs of $N$, such as $(v_0, (v_0, v_1), v_1, (v_2, v_1), v_2, (v_2, v_3), \ldots, (v_m, v_{m-1}), v_m)$, which can be more concisely expressed as $v_0 < v_1 < v_2 < v_3 < \cdots < v_{m-1} < v_m$ or in reverse order. A zig-zag trail $Z$ in $N$ is said to be maximal if $N$ contains no zig-zag trail $Z'$ such that $Z$ is a proper subgraph of $Z'$. A maximal zig-zag trail $Z$ with even $m := |A(Z)| \geq 4$ is called a crown if $Z$ can be written in the cyclic form $v_0 < v_1 < v_2 < v_3 < \cdots < v_{m-1} < v_m > v_0$ and is called a fence otherwise. Furthermore, a fence $Z$ with odd $|A(Z)|$ is called an $N$-fence, in which case $Z$ can be expressed as $v_0 < v_1 < v_2 < v_3 < \cdots < v_{m-1} < v_m > v_0$. A fence $Z$ with even $|A(Z)|$ is called an $M$-fence if it can be written in the form $v_0 < v_1 < v_2 < v_3 < \cdots < v_{m-2} < v_{m-1} > v_m$, rather than $v_0 > v_1 > v_2 > v_3 > \cdots > v_{m-2} > v_{m-1} > v_m$.

From now on, we represent a maximal zig-zag trail $Z$ by a sequence $\langle a_1, \ldots, a_{|A(Z)|}\rangle$ of the elements of $A(Z)$ that form the zig-zag trail in this order, assuming that no confusion arises. Then, we can encode an arbitrary arc-induced subgraph of $Z$ by an $|A(Z)|$-dimensional vector. For example, for an $N$-fence $Z = \langle a_1, a_2, a_3, a_4, a_5\rangle$ of the graph $A(Z)$, the information $Z$ is represented by the vector $(1 \ 0 \ 1 \ 0 \ 1) = (10101^2)$. With this notation, we can state Hayamizu’s structure theorem for tree-based
phylogenetic networks, which gives an explicit characterisation of the family $\Omega$ of all admissible arc-sets of $N$ as follows.

**Theorem 3.11** (8). Any tree-based phylogenetic $X$-network $N$ is uniquely decomposed into maximal zig-zag trails $Z_1, \ldots, Z_d$, each of which is a crown, M-fence or N-fence. Moreover, a subgraph $G$ of $N$ is a support tree of $N$ if and only if $A(G) \cap A(Z_i)$ is an admissible arc-set of $Z_i$ for any $i \in [1,d]$. Furthermore, the collection $\Omega$ of support trees of $N$ is characterised by a direct product of families $\Omega_1, \ldots, \Omega_d$ of the admissible arc-sets of $Z_1, \ldots, Z_d$, namely, we have $\Omega = \prod_{i=1}^d \Omega_i$ with

$$\Omega_i := \begin{cases} \{(01)^{|A(Z_i)|/2}, (10)^{|A(Z_i)|/2}\} & \text{if } Z_i \text{ is a crown;} \\ \{(10)(|A(Z_i)|-1)/2\} & \text{if } Z_i \text{ is an M-fence;} \\ \{(101)^p(10)^q1\} & p, q \in \mathbb{Z}_{\geq 0}, p + q = (|A(Z_i)| - 2)/2 \end{cases}$$ if $Z_i$ is an N-fence.

$4$. **Top-$k$ Support Tree Ranking Problem**

Given a tree-based phylogenetic $X$-network $N$ where each arc $a$ is chosen with probability $w(a) \in (0,1)$, we can assign a ranking number to each support tree $\tau \in \Omega$ of $N$ by the likelihood value $f(\tau) := \prod_{a \in A(\tau)} w(a)$. In principle, the top-$k$ support tree ranking problem for $N$ asks for an ordered set $(\tau^{(1)}, \ldots, \tau^{(k)})$ of $k$ support trees of $N$ such that $f(\tau^{(1)}) \geq \cdots \geq f(\tau^{(k)}) \geq f(\tau)$ holds for any support tree $\tau$ of $N$ other than $\tau^{(i)}$ ($i = 1, \ldots, k$). However, such a ranking is not unique in general, since there can be ‘ties’ in the collection $\Omega$ of support trees of $N$ as well as in the family $\Omega_i$ of admissible arc-sets of each maximal zig-zag trail $Z_i$ in $N$. For convenience, we ensure the uniqueness of the ranking by using the lexicographical order $\leq_{\text{lex}}$ on vectors as follows.

Assume that $N$ is a tree-based phylogenetic $X$-network with $\Omega = \prod_{i=1}^d \Omega_i$ as in Theorem 3.1 and that $Z_i$ is any maximal zig-zag trail in $N$. We define the local ranking for $Z_i$ to be a totally ordered set $(\Omega_i, \leq^*)$ such that for any $x, y \in \Omega_i$, $x \leq^* y$ holds if either $f(x) > f(y)$ or $f(x) = f(y)$ and $x \leq_{\text{lex}} y$ holds. Note that the elements of $\Omega_i$ are $|A(Z_i)|$-dimensional vectors and any two of them are comparable lexicographically.

From now, we identify the $j$-th element in $(\Omega_i, \leq^*)$ with its local ranking number $j \in [1, \ldots, |\Omega_i|]$ in order to write $\Omega = \prod_{i=1}^d \{1, \ldots, |\Omega_i|\}$. Then, the elements of $\Omega$ are vectors having the same dimension again and so we can break ties by using $\leq_{\text{lex}}$ as before. Abusing the notation $\leq^*$ slightly, we call the totally ordered set $(\Omega, \leq^*)$ the support tree ranking (for $N$). For any $k \in \mathbb{N}$ with $k \leq |\Omega|$, the top-$k$ support tree ranking (for $N$) is defined to be a unique subsequence of the first $k$ elements of $(\Omega, \leq^*)$. Note that for any $k \in \mathbb{N}$, one can determine in $O(|A(N)|)$ time whether or not $k \leq |\Omega|$ holds [8].

**Problem 4.1.** **Top-$k$ support tree ranking problem**

**Input:** A tree-based phylogenetic $X$-network $N$ with associated probability $w : A(N) \rightarrow (0,1]$ and $k \in \mathbb{N}$ not exceeding the number $|\Omega|$ of support trees of $N$.

**Output:** The top-$k$ support tree ranking $(\tau^{(1)}, \ldots, \tau^{(k)})$ for $N$.

5. **Results**

As a preliminary step, we prove the following proposition about the local ranking.

**Proposition 5.1.** For any maximal zig-zag trail $Z_i$ in a tree-based phylogenetic $X$-network $N$ with associated probability $w : A(N) \rightarrow (0,1]$, the first element in the local ranking $(\Omega_i, \leq^*)$ can be found in $O(|A(Z_i)|)$ time. Moreover, given the $j$-th element in $(\Omega_i, \leq^*)$, one can find the $(j+1)$-th element in $O(|A(Z_i)|)$ time.

**Proof.** One can check in $O(|A(Z_i)|)$ time whether $Z_i$ is a crown, N-fence or M-fence. In the case where $Z_i$ is a crown or N-fence, the local ranking for $Z_i$ is trivial to compute as $|\Omega_i| \leq 2$ holds by Theorem 3.1. Assume that $Z_i$ is an M-fence $(a_1, \ldots, a_{2m})$ with $|A(Z_i)| = 2m$. Also, let $x_{p+1} := (101)^p(10)^q1$ for each $p \in [0, m-1]$ and let $\Delta_p := w(a_{2p+1}) - w(a_{2p})$ for each $p \in [1, m-1]$. Then, $f(x_{p+1}) = f(x_p) + \Delta_p$ holds for each $p \in [1, m-1]$. As one can obtain both $f(x_1)$ and $(\Delta_1, \ldots, \Delta_{m-1})$ in $O(|A(Z_i)|)$ time, computing the likelihood values $f(x)$ for all $x \in \Omega_i$ requires $O(|A(Z_i)|)$ time. This completes the proof.

We define $I_0 := \emptyset$ and $I_j := \{\tau^{(0)}, \ldots, \tau^{(j)}\}$ for each $j \in [1,k]$. Recalling $\Omega = \prod_{i=1}^d \{1, \ldots, |\Omega_i|\}$, we see that $(\Omega, \leq^*)$ is a linear extension of the partially ordered set $(\Omega, \leq)$ (i.e., $x \leq y$ implies $x \leq^* y$), where $\leq$ is the usual component-wise order on vectors (e.g., $(x_1, x_2) \leq (y_1, y_2)$ if and only if $x_1 \leq y_1$ and $x_2 \leq y_2$). We also note that this requires each $I_j$ to be an order ideal of $(\Omega, \leq)$ (i.e., for any $x \leq y$, $y \in I_j$ implies $x \in I_j$). These arguments lead to the following proposition.
Proposition 5.2. Let \((τ^{(1)}, \ldots, τ^{(k)})\) be the top-k support tree ranking for a tree-based phylogenetic X-network \(N\) with associated probability \(w : A(N) → (0, 1)\) and let \(I_j\) be as defined above. Then, \(I_1 = \{(1 \ldots 1)\}\) holds, and for each \(j \in [1, k−1]\), there exists \(τ \in I_j\) with \(∥τ^{(j+1)} − τ^{(1)}∥ = 1\).

Let \(e_i\) be the unit vector such that \(i\)-th component is one and the others are all zeros. Also, for each \(τ \in \Omega \setminus \{τ^{(1)}\}\), let \(id(τ)\) be the first index such that the \(i\)-th component of \(υ\) is strictly greater than one and let \(e(τ) := e_{id(τ)}\). For example, \(τ = (1 1 1 5 8)\) gives \(e(τ) = (0 0 0 1 0)\). Then, we have the next lemma, which is illustrated in Figure 1.

Lemma 5.3. Let \((Ω, ≤^*)\) be the support tree ranking for a tree-based phylogenetic X-network \(N\) with associated probability \(w : A(N) → (0, 1)\) and let \(Γ\) be a graph with \(V(Γ) = Ω\) and \(A(Γ) = \{(τ, τ') \in \Omega × (Ω \setminus \{τ^{(1)}\}) | τ = τ' − e(τ')\}\). Then, \(Γ\) is a spanning tree of the Hasse diagram of \((Ω, ≤)\) such that \(τ^{(1)}\) is the root of \(Γ\) and \((τ, τ') \in A(Γ)\) implies \(τ ≤^* τ'\).

Proof. It is clear that \((τ, τ') \in A(Γ)\) implies \(τ ≤ τ'\) (and hence \(τ ≤^* τ'\)). By construction, \(Γ\) is a tree rooted at \(τ^{(1)}\) because \(deg^-_Γ(τ^{(1)}) = 0\) holds and for each \(τ' \in V(Γ) \setminus \{τ^{(1)}\}\), there exists a unique element \(τ \in V(Γ)\) with \((τ, τ') \in A(Γ)\). This completes the proof. \(\square\)

Figure 1. An illustration of Lemma 5.3. The top left is a tree-based phylogenetic X-network \(N\) whose arcs are associated with probability. The top right shows the maximal zig-zag trails \(Z_i\) in \(N\) with \(|Ω_i| ≥ 2\) and the likelihood of each element of \((Ω_i, ≤^*)\) \((i = 1, 2, 3)\). On the bottom is the spanning tree \(Γ\) (shown in bold) of the Hasse diagram of \((Ω, ≤)\).

In what follows, for any \(Ω' ⊆ Ω\), we write \(least(Ω')\) to mean the least element of \((Ω', ≤^*)\). For any \(τ \in Ω\), let \(Child(τ) := \{τ' \in V(Γ) | (τ, τ') \in A(Γ)\}\) and \(child^*(τ) := least(Child(τ))\). Also, for any \(τ' \in Ω \setminus \{τ^{(1)}\}\), let \(sibling^*(τ') := least(τ ∈ Child(parent(τ')) | τ' ≤^* τ \land τ \neq τ')\), where \(parent(τ')\) represents a unique...
element \( r \in V(\Gamma) \) with \( (r, r') \in A(\Gamma) \). We note that both \( \text{child}^*(r') = \emptyset \) and \( \text{sibling}^*(r') = \emptyset \) are possible to occur.

**Lemma 5.4.** Let \( \Gamma \) be the graph as in Lemma 5.3, and let \( Q_j \) be a subset of \( V(\Gamma) \) that is recursively defined by

\[
Q_j := \begin{cases} 
  (Q'_{j-1} \setminus \{r^{(j-1)}\}) \cup \{\text{child}^*(r^{(j-1)}), \text{sibling}^*(r^{(j-1)})\}, & j \in [2, k] \\
  \{r^{(1)}\}, & j = 1.
\end{cases}
\]

Then, for each \( j \in [1, k] \), we have \( r^{(j)} \in Q_j \) and \( r^{(\ell)} \not\in Q_j \) for all \( \ell < j \).

**Proof.** Let \( P_1 = Q_1 \) and \( P_j = \{\text{least}(\text{Child}(r) \setminus I_{j-1}) \mid r \in I_{j-1}\} \) for \( j \in [2, k] \). We will show that \( P_j = Q_j \) holds for any \( j \in [1, k] \), which completes the proof, since \( r^{(j)} \in P_j \) and \( r^{(\ell)} \not\in P_j \) for all \( \ell < j \).

For \( j \in [2, k] \), we have

\[
P_j = \{\text{least}(\text{Child}(r) \setminus I_{j-1}) \mid r \in I_{j-1}\} \cup \{\text{least}(\text{Child}(r \setminus I_{j-1}))\} = \{\text{least}(\text{Child}(r) \setminus I_{j-1}) \mid r \in I_{j-2}\} \cup \{\text{child}^*(r^{(j-1)}), \text{sibling}^*(r^{(j-1)})\},
\]

where we assume that \( I_0 = \emptyset \). Note that \( I_{j-2} \) contains \( p := \text{parent}(r^{(j-1)}) \). This implies

\[
P_j = \{\text{least}(\text{Child}(r) \setminus I_{j-1}) \mid r \in I_{j-2}\} \cup \{\text{child}^*(r^{(j-1)}), \text{sibling}^*(r^{(j-1)})\}.
\]

For any \( r \in I_{j-2} \), we have \( \text{least}(\text{Child}(r) \setminus I_{j-1}) = \text{least}(\text{Child}(r) \setminus I_{j-2}) \) because \( r^{(j-1)} \not\in \text{Child}(r) \) holds. We thus obtain

\[
P_j = \{\text{least}(\text{Child}(r) \setminus I_{j-1}) \mid r \in I_{j-2}\} \cup \{\text{least}(\text{Child}(p) \setminus I_{j-2})\} \cup \{\text{child}^*(r^{(j-1)}), \text{sibling}^*(r^{(j-1)})\} = \{P_{j-1} \setminus \{r^{(j-1)}\} \cup \{\text{child}^*(r^{(j-1)}), \text{sibling}^*(r^{(j-1)})\}
\]

From Equation 1 and \( P_1 = Q_1 \), the desired conclusion follows.

We are in a position to give an algorithm for Problem 4.1. As illustrated in Table 1, the algorithm starts by setting \( j := 1 \) and \( Q_1 := \{r^{(1)}\} \) and then returns \( r^{(j)} = \text{least}(Q_j) \) for each \( j \in [1, k] \), where \( Q_j \) is iteratively updated using Equation 1.

| \( j \) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| \( Q_j \) | \{1 1\} | \{1 1 2\}, \{2 1 2\} | \{1 1 2\} | \{1 1 2\} | \{1 1 1\}, \{1 2 2\}, \{1 2 1\}, \{2 1 2\}, \{2 1 1\}, \{1 2 2\} | \{1 2 2\}, \{2 2 1\}, \{2 2 2\} | \{2 2 2\} |
| \( r^{(1)} \) | \{1 1\} | \{1 1 2\}, \{2 1 2\} | \{1 1 2\} | \{1 1 2\} | \{1 1 1\}, \{1 2 2\}, \{1 2 1\}, \{2 1 2\}, \{2 1 1\}, \{1 2 2\} | \{1 2 2\}, \{2 2 1\}, \{2 2 2\} | \{2 2 2\} |
| \text{child}^*(r^{(1)}) | \{1 2\} | \{1 2 2\} | \{1 2\} | \{1 2\} | \{1 2\}, \{2 1\}, \{2 1\} | \{2 2\}, \{2 2\} | \{2 2\} |
| \text{sibling}^*(r^{(1)}) | \{2 1\} | \{2 1\} | \{2 1\} | \{2 1\} | \{2 1\} | \{2 1\} | \{2 1\} |

Table 1: Application of the proposed algorithm to the input \( N \) in Figure 1 (\( k = 8 \)).

In order to analyse the running time of the above algorithm, let us review some basics of a priority queue, which is a data structure for maintaining objects that are prioritised by their associated values. In its most basic form, a priority queue supports the operations called INSERT and DELETE-MIN, where the former refers to adding a new object, and the latter to detecting and deleting the one with the highest-priority [2]. Implemented with a binary heap, each of these operations can be performed in \( O(\log n) \) time, where \( n \) denotes the number of elements in the priority queue [2].

**Theorem 5.5.** The top-\( k \) support tree ranking problem (Problem 4.1) can be solved with linear delay, and hence in \( O(\log|A(N)|) \) time.

**Proof.** As Equation 1 implies that \( |Q_{j+1} - Q_j| \leq 1 \) holds for any \( j \in [1, k - 1] \), \( |Q_j| \leq k \) holds for any \( j \in [1, k] \). Then, if we keep the elements of each \( Q_j \) in a priority queue, \( O(\log k) \) time suffices to return \( r^{(j)} \) and to delete \( r^{(j)} \) from \( Q_j \). Also, once \( \text{child}^*(r^{(j)}) \) and \( \text{sibling}^*(r^{(j)}) \) have been obtained, inserting the two elements requires \( O(\log k) \) time. We note that \( O(\log k) \leq O(|A(N)|) \) follows from \( k \leq 2^{|A(N)|} \). By Proposition 5.1 for each \( j \in [1, k - 1] \), one can compute \( \{\text{child}^*(r^{(j)}), \text{sibling}^*(r^{(j)})\} \) in \( \sum_{i=1}^{k} O(|A(Z_i)|) \) time, which equals \( O(|A(N)|) \) time as \( Z_1, \ldots, Z_{k} \) is a decomposition of \( N \). Hence, our algorithm can return \( r^{(1)}, \ldots, r^{(k)} \) one after the other in such a way that the delay between two consecutive outputs is \( O(|A(N)|) \) time. This completes the proof. \( \square \)
Finally, we make two remarks. First, \( \Omega(k|A(N)|) \) time is required to output \( k \) distinct support trees of \( N \) as each support tree has size \( \Omega(|A(N)|) \). Therefore, the running time of our algorithm (as well as that of the enumeration algorithm in [8]) is \( \Theta(k|A(N)|) \), which guarantees the optimality of those algorithms. Second, as commonly in the literature (e.g., [10]), it would be natural to wonder about the time complexity of an analogue of Problem 4.1 that only asks for outputting a sequence of the differences between \( \tau(j-1) \) and \( \tau(j) \); however, we note that this problem still requires \( \Omega(k|A(N)|) \) time because the size of each difference is \( \Omega(|A(N)|) \). To illustrate this, consider a tree-based phylogenetic \( X \)-network \( N \) that is decomposed into maximal fences, each of which has only one admissible arc-set, and \( c \) crowns, each of which has size \( \Omega(|A(N)|/c) \). The difference between any two support trees has size \( \Omega(|A(N)|/c) \), which equals \( \Omega(|A(N)|) \) if \( c \) is a constant.

REFERENCES

1. M. Anaya, O. Anipchenko-Ulaj, A. Ashfaq, J. Chiu, M. Kaiser, M. S. Ohsawa, M. Owen, E. Pavlechko, K. St. John, S. Suleria, K. Thompson, and C. Yap, On determining if tree-based networks contain fixed trees, Bulletin of Mathematical Biology 78 (2016), no. 5, 961–969.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms, MIT press, 2009.
3. M. Fischer, M. Galla, L. Herbst, Y. Long, and K. Wicke, Non-binary treebased unrooted phylogenetic networks and their relations to binary and rooted ones, arXiv:1810.06853 [q-bio.PE] (2018).
4. A. Francis, K. T. Huber, and V. Moulton, Tree-based unrooted phylogenetic networks, Bulletin of mathematical biology 80 (2018), no. 2, 404–416.
5. A. Francis, C. Semple, and M. Steel, New characterisations of tree-based networks and proximity measures, Advances in Applied Mathematics 93 (2018), 93–107.
6. A. R. Francis and M. Steel, Which phylogenetic networks are merely trees with additional arcs?, Systematic Biology 64 (2015), no. 5, 768–777.
7. M. Hayamizu, On the existence of infinitely many universal tree-based networks, Journal of Theoretical Biology 396 (2016), 204–206.
8. ____., A structure theorem for tree-based phylogenetic networks, arXiv:1811.05849 [math.CO] (2018).
9. D. H. Huson, R. Rupp, and C. Scornavacca, Phylogenetic networks: concepts, algorithms and applications, Cambridge University Press, 2010.
10. S. Kapoor and H. Ramesh, Algorithms for enumerating all spanning trees of undirected and weighted graphs, SIAM Journal on Computing 24 (1995), no. 2, 247–265.
11. J. C. Pons, C. Semple, and M. Steel, Tree-based networks: characterisations, metrics, and support trees, Journal of Mathematical Biology 78 (2019), no. 4, 809–918.
12. L. van Iersel, Different topological restrictions of rooted phylogenetic networks. Which make biological sense?, http://phylonetworks.blogspot.nl/2013/03/different-topological-restrictions-of.html, 2013, Accessed: 2019-03-16.
13. L. Zhang, On tree-based phylogenetic networks, Journal of Computational Biology 23 (2016), no. 7, 553–565.

ACKNOWLEDGEMENT

The first author acknowledges support from JST PRESTO Grant Number JPMJPR16EB.

THE INSTITUTE OF STATISTICAL MATHEMATICS, TOKYO, JAPAN
E-mail address: hayamizu@ism.ac.jp

RESEARCH INSTITUTE FOR MATHEMATICAL SCIENCES, KYOTO UNIVERSITY, KYOTO, JAPAN
E-mail address: makino@kurims.kyoto-u.ac.jp