Computing a Smallest Multi-labeled Phylogenetic Tree from Rooted Triplets

Sylvain Guillemot[1], Jesper Jansson[2,⋆,⋆⋆], and Wing-Kin Sung[3,4]

[1] Institut Gaspard Monge - Université Paris-Est, 5 boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée, France
Sylvain.Guillemot@univ-mlv.fr
[2] Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan
Jesper.Jansson@ocha.ac.jp
[3] School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543
ksung@comp.nus.edu.sg
[4] Genome Institute of Singapore, 60 Biopolis Street, Genome, Singapore 138672

Abstract. We investigate the computational complexity of a new combinatorial problem of inferring a smallest possible multi-labeled phylogenetic tree (MUL tree) which is consistent with each of the rooted triplets in a given set. We prove that even the restricted case of determining if there exists a MUL tree consistent with the input and having just one leaf duplication is NP-hard. Furthermore, we show that the general minimization problem is NP-hard to approximate within a ratio of $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$, where n denotes the number of distinct leaf labels in the input set, although a simple polynomial-time approximation algorithm achieves the approximation ratio n. We also provide an exact algorithm for the problem running in $O^*(7^n)$ time and $O^*(3^n)$ space.

1 Introduction

1.1 Problem Definitions

A *phylogenetic tree* is a rooted, unordered tree in which every internal node has at least two children and where each leaf is labeled by an element from a set of leaf labels. A phylogenetic tree where each leaf label occurs at most once is called a *single-labeled phylogenetic tree*; a phylogenetic tree where each leaf label may occur more than once is called a *multi-labeled phylogenetic tree*, or *MUL tree* for short [6,8,11].[1] For any MUL tree M, denote the set of all leaf labels that occur in M by $\mathcal{L}(M)$. For any leaf label $x \in \mathcal{L}(M)$, the number of *duplications of* x is equal to the number of occurrences of x in M minus 1. The number of *leaf duplications in* M, denoted by $d(M)$, is the total number of duplications

⋆ Funded by the Special Coordination Funds for Promoting Science and Technology.
⋆⋆ Corresponding author.
[1] MUL trees are called *rl-trees* in [6].

Y. Dong, D.-Z. Du, and O. Ibarra (Eds.): ISAAC 2009, LNCS 5878, pp. 1205–1214, 2009.

of all leaf labels in $\mathcal{L}(M)$. Define $m(M)$ as the number of leaves in M. Then, $d(M) = m(M) - |\mathcal{L}(M)|$.

For any two nodes u, v in a rooted tree, the notation $u \prec v$ means that u is a proper descendant of v, and $\mathsf{lca}(u, v)$ denotes the lowest common ancestor (lca) of u and v. (For convenience, any node is considered to be an ancestor of itself.) A *rooted triplet* is a binary phylogenetic tree with exactly three distinctly labeled leaves. The unique rooted triplet on leaf label set $\{x, y, z\}$ where $\mathsf{lca}(x, y) \prec \mathsf{lca}(x, z) = \mathsf{lca}(y, z)$ is denoted by $xy|z$. If $xy|z$ is an embedded subtree of a MUL tree M, i.e., if there exist three leaves ℓ_x, ℓ_y, ℓ_z in M labeled by x, y, and z, respectively, such that $\mathsf{lca}(\ell_x, \ell_y) \prec \mathsf{lca}(\ell_x, \ell_z) = \mathsf{lca}(\ell_y, \ell_z)$ then $xy|z$ and M are said to be *consistent* with each other; otherwise, $xy|z$ and M are *inconsistent*. A set $\mathcal{R}$ of rooted triplets and a MUL tree M are *consistent* with each other if every $xy|z \in \mathcal{R}$ is consistent with M. See Fig. 1 for an example.

In this paper, we consider the following problem, named *the smallest MUL tree from rooted triplets problem* (SMRT): Given a set $\mathcal{R}$ of rooted triplets over a leaf label set L, output a MUL tree M with $\mathcal{L}(M) = L$ which is consistent with $\mathcal{R}$ and which minimizes $d(M)$. We also consider the following decision problem for any positive integer d, termed d-SMRT: Given a set $\mathcal{R}$ of rooted triplets over a leaf label set L, does there exist a MUL tree M with $\mathcal{L}(M) = L$ which is consistent with $\mathcal{R}$ and which satisfies $d(M) \leq d$? In the rest of this paper, we define $k = |\mathcal{R}|$ and $n = |L|$ for any given instance of SMRT or d-SMRT.

1.2 Motivation and Previous Work

The problem of determining whether there exists a *single-labeled* tree consistent with all of the rooted triplets in a given set, and if so, constructing such a tree, can be solved efficiently by a classical algorithm of Aho *et al.* [1]. When no such tree exists because of conflicts in the branching information, one may try to select a largest possible subset of the triplets which is consistent with some tree (*the maximum rooted triplets consistency problem* (MRTC)), find a largest possible subset of the leaves such that the restriction of the input triplets to those leaves is consistent with some tree (*the maximum agreement supertree problem* (MASP)), or build a *phylogenetic network* (a generalization of a phylogenetic tree in which internal nodes may have more than a single parent) which contains all of the rooted triplets. See [3] for a recent survey of related results and many references. In this paper, we consider a new approach: Allow leaf labels to be repeated, but try to minimize the number of such repetitions.

The main application of phylogenetic trees is to describe tree-like evolution for a set of objects; leaves represent the objects while internal nodes correspond to their common ancestors. In the study of evolutionary history, MUL trees arise from the modeling of biological processes where it is necessary to use certain leaf labels more than once. For example, a gene tree can contain several leaves labeled by the same species due to gene duplication events [6,8,11]. As another example, area cladograms, where the names of geographical areas are used to label the leaves, may apply the same label to more than a single leaf (see, e.g., [2,8]). MUL trees can also be useful for studying host-parasite cospeciation [8,10].

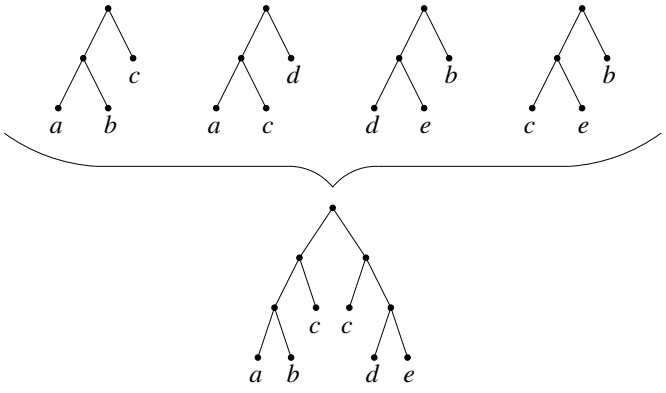

Fig. 1. The set of rooted triplets $\{ab|c,\ ac|d,\ de|b,\ ce|b\}$ is consistent with a MUL tree containing one leaf duplication

Although the problem of inferring a MUL tree from an input set of single-labeled phylogenetic trees that minimizes the number of leaf duplications has not been studied before, several algorithms for manipulating already known MUL trees have been published in the literature. Huber *et al.* [8] presented a method for constructing a phylogenetic network from an input MUL tree. The network output by their method is binary and has the fewest possible reticulation nodes among all binary networks which exhibit the structural information of the input MUL tree. Scornavacca *et al.* [11] considered some computational problems involving the extraction of the unambiguous parts of an input MUL tree. More precisely, [11] proposed linear-time algorithms to identify every so-called observed duplication node (odn) in a MUL tree, testing if two MUL trees are isomorphic, and computing a largest duplication-free rooted subtree of a MUL tree. They also showed that it is an NP-hard problem to prune each of the MUL trees in a given set to a single-labeled tree at odns in such a way that the obtained set of trees can be merged without conflicts into a single-labeled tree.

1.3 Our Results and Organization of the Paper

We present several negative and positive results regarding the computational complexity and polynomial-time approximability of SMRT. Below, we say that an algorithm $\mathcal{A}$ for SMRT is an *α-approximation algorithm* (and that the *approximation ratio of* $\mathcal{A}$ is at most α) if, for every input $\mathcal{R}$, the MUL tree output by $\mathcal{A}$ is consistent with $\mathcal{R}$ and contains at most $\alpha \cdot d(M^*)$ leaf duplications, where M^* is an optimal MUL tree (i.e., having the fewest possible number of leaf duplications) consistent with $\mathcal{R}$.

The rest of the paper is organized as follows. Section 2 presents a simple polynomial-time n-approximation algorithm for SMRT. On the negative side, Section 3 proves that d-SMRT is NP-hard even if $d = 1$, and also that SMRT cannot be approximated within a ratio of $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$ in

polynomial time, unless $\mathsf{P} = \mathsf{NP}$. Finally, Section 4 presents an exact algorithm for SMRT which runs in $O^*(7^n)$ time and $O^*(3^n)$ space.

2 Straightforward n-Approximation of SMRT

We start with the following simple observation.

Lemma 1. *For any set $\mathcal{R}$ of rooted triplets over a leaf label set L with $|L| = n$, there exists a MUL tree with $2n$ leaves which is consistent with $\mathcal{R}$.*

Proof. Let T be an arbitrary single-labeled phylogenetic tree with n leaves bijectively labeled by L. Let M be the MUL tree obtained by taking two copies T_1, T_2 of T and joining the roots of T_1 and T_2 to a new parent root node. Clearly, M has $2n$ leaves and any rooted triplet $xy|z$ over L is consistent with M since T_1 contains leaves labeled by x, y and T_2 contains a leaf labeled by z. □

Consequently, SMRT admits a trivial polynomial-time n-approximation algorithm: Using the algorithm of Aho *et al.* [1], determine if there exists a single-labeled tree consistent with $\mathcal{R}$. If the answer is yes then output this tree, otherwise output the MUL tree from Lemma 1 which has exactly n leaf duplications.

Theorem 1. SMRT *can be approximated within a ratio of n in polynomial time.*

3 Hardness Results for SMRT

This section demonstrates that SMRT is computationally intractable. It is shown that d-SMRT is NP-hard already for $d = 1$ and that SMRT is NP-hard to approximate within a ratio of $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$. (Recall that n denotes the number of distinct leaf labels in the input set $\mathcal{R}$.) To obtain our hardness results, we first prove strong inapproximability bounds for a problem on directed graphs named Acyclic Tree-Partition (defined below) and then give a measure-preserving reduction from Acyclic Tree-Partition to SMRT.

3.1 Hardness of Acyclic Partition and Acyclic Tree-Partition

Definition 1. *Let $D = (V, A)$ be a directed graph. An* acyclic partition *of D is a partition of V into subsets $V_1, \ldots, V_r$ called* classes *such that each class induces an acyclic subgraph of D.*

Definition 2. *Let $D = (V, A)$ be a directed graph. An* acyclic tree-partition *of D consists of a binary rooted tree T with a node set N along with a partition $\{V(x) : x \in N\}$ of V (i.e., a subset $V(x)$ of V is associated to each node x of the tree T) such that:*

1. *for every $x \in N$, $V(x)$ induces an acyclic subgraph of D,*
2. *for any $x, y \in N$ with $x \prec y$, D has no arc from $V(y)$ to $V(x)$.*

Definitions 1 and 2 lead to the following natural problems. The Acyclic Partition problem takes as input a directed graph D and seeks an acyclic partition of D with the smallest possible number of classes; this number is denoted by $ap(D)$.[2] Similarly, the Acyclic Tree-Partition problem seeks an acyclic tree-partition of an input directed graph D with the minimum number of internal nodes, denoted by $atp(D)$. For any positive integer r, the two decision problems r-Acyclic Partition and r-Acyclic Tree-Partition ask if an input directed graph D satisfies $ap(D) \leq r$ and $atp(D) \leq r$, respectively.

Acyclic partitions and acyclic tree-partitions have some useful properties:

Lemma 2. *Let D be a directed graph and let $(T, \{V(x) : x \in N\})$ be an acyclic tree-partition of D. For any set X of ancestors of a leaf in T, the union $\bigcup_{x \in X} V(x)$ induces an acyclic subgraph of D.*

Lemma 3. *For every directed graph D, $atp(D) = ap(D) - 1$.*

Theorem 2. *(i) r-Acyclic Partition is NP-hard for $r = 2$.*
(ii) Acyclic Partition cannot be approximated within $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$ in polynomial time unless $\mathsf{P} = \mathsf{NP}$, where n is the number of vertices in the input graph.

Proof. (i) Reduce from Not-all-equal 3SAT, which is known to be NP-hard [7]. Let I be a given instance of Not-all-equal 3SAT with m clauses and construct a directed graph D with $3m$ vertices as follows. For each clause C in I, D contains three vertices C_1, C_2, C_3 forming a directed cycle in D that represent the literals of C. In addition, for each pair of conflicting literals $C_i = x$ and $C'_j = \overline{x}$, D contains the two arcs (C_i, C'_j) and (C'_j, C_i). It is easy to see that there is a one-to-one correspondence between the valid truth assignments for I and the acyclic bipartitions of D: given a truth assignment ϕ, define a bipartition V_t, V_f of D by letting V_t (resp. V_f) contain all literals which are assigned the value *true* (resp. *false*) under ϕ.

(ii) Follows by giving a measure-preserving reduction from Chromatic Number and applying known inapproximability results for this problem [5,13]. The reduction maps a given undirected graph $G = (V, E)$ to a directed graph $D = (V, A)$ by replacing each edge $\{u, v\}$ of G by two arcs $(u, v), (v, u)$. Observe that for any $V' \subseteq V$, V' is an independent set of G if and only if V' induces an acyclic subgraph of D. Therefore, colorings of G correspond to acyclic partitions of D. □

Corollary 1. *(i) r-Acyclic Tree-Partition is NP-hard for $r = 1$.*
(ii) Acyclic Tree-Partition cannot be approximated within $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$ in polynomial time unless $\mathsf{P} = \mathsf{NP}$, where n is the number of vertices in the input graph.

3.2 Hardness of SMRT

We first reduce Acyclic Tree-Partition to a *constrained* variant of SMRT that forbids duplications of certain labels (Proposition 1). We then reduce the

[2] $ap(D)$ is also referred to in the literature as *the dichromatic number of D* [9].

constrained variant to the unconstrained SMRT problem (Proposition 2). When combined, these reductions yield the desired hardness results for SMRT, as summarized in Theorem 3. The constrained variant of SMRT is defined as follows.

Definition 3. *Let $\mathcal{R}$ be a set of rooted triplets over a leaf label set L and $U \subseteq L$ a set of* unique labels. *A MUL tree M is* consistent *with the pair $(\mathcal{R}, U)$ if: (i) M is consistent with $\mathcal{R}$; (ii) M has only one occurrence of each label in U.*

The Constrained-SMRT problem (C-SMRT) takes as input a pair $(\mathcal{R}, U)$ and seeks a MUL tree consistent with $(\mathcal{R}, U)$ containing the fewest duplications.

Proposition 1. *There exists a measure-preserving reduction from* Acyclic Tree-Partition *to* C-SMRT.

Proof. Given an instance $D = (V, A)$ of Acyclic Tree-Partition, construct a new instance $(\mathcal{R}, U)$ of C-SMRT with label set $L := V \cup \{z\}$, where z is a new label not belonging to V. The set $\mathcal{R}$ contains exactly the following triplets: for each arc $(u, v) \in A$, let $zu|v \in \mathcal{R}$. The set of unique labels is $U = V$, meaning that only z is allowed to be duplicated. To prove that the reduction is measure-preserving, we show that for every $r \leq |V|$, the following are equivalent:

1. D admits an acyclic tree-partition with r internal nodes;
2. $(\mathcal{R}, U)$ admits a consistent MUL tree with r duplications.

(1) $\Rightarrow$ (2): Suppose D has an acyclic tree-partition consisting of a tree $T = (N, E)$ with r internal nodes and a partition $\{V_x : x \in N\}$ of V. We construct a MUL tree M from T by labeling each leaf by z, and then, above each node x of T, attaching the elements of V_x in the order given by a topological ordering of $D[V_x]$ (where $D[V_x]$ denotes the subgraph of D induced by vertices of V_x).

We introduce the following additional notation: given a MUL tree M, and a sequence of labels $s = x_1 \dots x_n$, let $R(M, s)$ be the tree obtained by starting with a caterpillar with $n + 1$ leaves $l_0, \dots, l_n$ (with l_0, l_1 being farthest from the root), substituting l_0 with M, and labeling each leaf $l_i, i \geq 1$ by x_i. We inductively define two MUL trees M_x, M'_x for each node x of T: (i) if x is a leaf then M_x consists of a single leaf labeled by z; (ii) if x is an internal node with two children y, y' then $M_x := (M'_y, M'_{y'})$; (iii) for any node x of T, let s_x be a topological ordering of $D[V_x]$ (which is acyclic by Point 1 of Definition 2), and let $M'_x := R(M_x, s_x)$. Finally, define $M := M'_t$, where t is the root of T.

We now examine the constructed MUL tree M. Clearly, only z is duplicated in M; since $\{V_x : x \in N\}$ is a partition of V, a label $u \in V_x$ appears only once in M (in the subtree between the root of M_x and the root of M'_x). Moreover, since the leaves of M labeled by z correspond to the leaves of T, their number is $r + 1$, hence M has r duplications. Next, we show by a case analysis that M is consistent with $\mathcal{R}$. Consider $zu|v \in \mathcal{R}$, then $(u, v) \in A$ by the construction of $\mathcal{R}$. Let x, y be the nodes of T such that $u \in V_x, v \in V_y$. Four cases are possible:

- if $x = y$: Since $(u, v) \in A$, and since s_x is a topological ordering of $D[V_x]$, it follows that $u <_{s_x} v$. Consider $M'_x = R(M_x, s_x)$. M_x contains a leaf labeled by z, and u, v appear in s_x with $u <_{s_x} v$, so M'_x (and thus M) contains $zu|v$.

- if $x \prec y$ in T: Consider $M'_y = R(M_y, s_y)$. M_y contains leaves labeled by z, u and v appears in s_y, therefore M'_y (and thus M) contains $zu|v$.
- if $y \prec x$ in T: This is impossible according to Point 2 of Definition 2.
- if both $x \not\prec y$ and $y \not\prec x$ in T: Let $c = \mathsf{lca}(x, y)$ in T and let c_x, c_y be the two (distinct) children of c such that $x \preceq c_x, y \preceq c_y$. Consider $M_c = (M'_{c_x}, M'_{c_y})$, then M'_{c_x} contains leaves labeled by z, u and M'_{c_y} contains a leaf labeled by v, hence M_c (and M) contains $zu|v$.

To conclude, M is a MUL tree with r duplications that is consistent with $(\mathcal{R}, U)$.

(2) $\Rightarrow$ (1): Let M be a MUL tree with r duplications which is consistent with $(\mathcal{R}, U)$. We may assume w.l.o.g. that M is binary. By definition, only the label z is duplicated in M. Let $T = (N, E)$ be the subtree of M which connects the leaves labeled by z. For each node x of T, let P_x be the path in M joining x to its parent node in T (or to the root of M if x is the root of T). Then, define V_x as the set of labels appearing in a subtree along the path P_x. It is straightforward to verify that $(T, \{V_x : x \in N\})$ is an acyclic tree-partition of D with r internal nodes (see the full version of this paper for a complete proof). □

Proposition 2. *There exists a measure-preserving reduction from* C-SMRT *to* SMRT.

Proof. Let $(\mathcal{R}, U)$ be any given instance of C-SMRT, where $\mathcal{R}$ is a triplet set over a set L of n leaf labels and $U \subseteq L$ is a set of unique labels. We construct an instance $\mathcal{R}'$ of SMRT by replacing each element of U by $n+1$ copies. Formally, $\mathcal{R}'$ has a leaf label set L' consisting of: (i) for each $x \in U$, labels x_i $(1 \leq i \leq n+1)$; (ii) for each $x \in L \backslash U$, a single element x_1. The set $\mathcal{R}'$ consists of the following triplets: for each $xy|z \in \mathcal{R}$ and each i, j, k, let $x_i y_j | z_k \in \mathcal{R}'$. Assume w.l.o.g. that $r \leq n$. We show that $(\mathcal{R}, U)$ has a consistent MUL tree M with r duplications if and only if $\mathcal{R}'$ has a consistent MUL tree M' with r duplications.

($\Rightarrow$): Let M be a MUL tree with r duplications consistent with $(\mathcal{R}, U)$. Construct a MUL tree M' from M by substituting each leaf u having label x by an arbitrary single-labeled binary tree T_u over $\{x_1, \ldots, x_i\}$, where i equals either 1 or $n+1$. Observe that: (i) for each $x \in U$, each label x_i occurs exactly once in M'; (ii) for each $x \in L \backslash U$, the number of occurrences of x in M equals the number of occurrences of x_1 in M'. It follows that $d(M') = d(M) = r$. In addition, for any triplet $x_i y_j | z_k \in \mathcal{R}'$, the corresponding $xy|z \in \mathcal{R}$ is obtained from leaves u, v, w of M; hence $x_i y_j | z_k$ is obtained by selecting the corresponding leaves of T_u, T_v, T_w in M'. This proves that M' is consistent with every triplet in $\mathcal{R}'$.

($\Leftarrow$): Omitted due to space constraints. See the full version of this paper for a complete proof. □

Propositions 1 and 2 together with our hardness results for Acyclic Tree-Partition in Corollary 1 give us the next theorem.

Theorem 3. *(i)* d-SMRT *is* NP*-hard for* $d = 1$*;*
(ii) SMRT *cannot be approximated within* $n^{1-\epsilon}$ *for any constant* $0 < \epsilon \leq 1$ *in polynomial time, unless* P = NP.

We remark that the analogous Minimum Duplication Supersequence problem [6] for strings behaves quite differently: it is equivalent to the Directed Feedback Vertex Set problem, and as such it is FPT with respect to r (by a result of [4]) and approximable within $O(\log n \log\log n)$ in polynomial time [12].

4 An Exact Algorithm for SMRT

Here, we present an exact exponential-time algorithm for SMRT.

We use a dynamic programming approach, exploiting the recursive structure of the problem. More precisely, we consider pairs of subsets of L of the form (A, B) such that $B \subseteq A \subseteq L$. Subproblems in our dynamic programming approach will correspond to pairs (A, B). For a given pair, we will restrict our attention to specific MUL trees given by the following definition.

Definition 4. *Let (A, B) be a pair of subsets of L with $B \subseteq A \subseteq L$. A binary MUL tree M leaf-labeled by A* complies with *(A, B) if and only if for each $uv|w \in \mathcal{R}$ with $u, v, w \in A$ and $w \notin B$, it holds that $uv|w$ is consistent with M.*

For a given pair (A, B), let $n(A, B)$ denote the minimum value of $d(M)$ taken over every binary MUL tree M leaf-labeled by A which complies with (A, B). We compute the values $n(A, B)$ by dynamic programming, and obtain $n(L, \emptyset)$ as the desired value at the end of the computation. To compute a value $n(A, B)$, we break the computation into two subproblems of the form $(A_1, \_), (A_2, \_)$, where A_1, A_2 are the label sets of the two child subtrees. In order to explain this in detail, we will need the following definitions. A *split* of (A, B) is a pair (A_1, A_2) of subsets of A such that $A_1 \cup A_2 = A$ (observe here that A_1, A_2 are not necessarily disjoint, and that the definition does not depend on B).

Definition 5. *Let (A_1, A_2) be a split of (A, B). We say that (A_1, A_2) is a* nice split of *(A, B) if and only if the following holds: for each $u, v, w \in A$, if $u \in A_i \backslash A_j$, $v \in A_j \backslash A_i$, $w \notin B$ with $i \neq j$ then $\mathcal{R}$ does not contain the rooted triplet $uv|w$.*

From here on, we will denote by B_i the intersection of B with A_i. Also, we define $B' = A_1 \cap A_2$. The next property describes the recursive structure of the problem, characterizing the fact that M complies with (A, B) by conditions on its child subtrees.

Lemma 4. *Let (A, B) be a pair such that $B \subseteq A \subseteq L$ with $|A| \geq 2$ and let M be a binary MUL tree over A, consisting of two MUL trees M_1, M_2 joined by a parent root node. Write $A_1 = \mathcal{L}(M_1)$ and $A_2 = \mathcal{L}(M_2)$, $B' = A_1 \cap A_2$ and $B_i = B \cap A_i$. Then the following are equivalent:*

1. *M complies with (A, B);*
2. *(A_1, A_2) is a nice split of (A, B), and for $i \in \{1, 2\}$, M_i complies with $(A_i, B_i \cup B')$.*

Proof. (1) ⇒ (2): We first show that M_i complies with $(A_i, B_i \cup B')$. Suppose that $uv|w \in \mathcal{R}$ with $u, v, w \in A_i$ and $w \notin B_i \cup B'$. Then we also have $u, v, w \in A$ and $w \notin B$, which implies that $uv|w$ is consistent with M (since M complies with (A, B)). Therefore, M has leaves ℓ_u, ℓ_v, ℓ_w labeled by u, v, w such that $\mathsf{lca}(\ell_u, \ell_v) \prec \mathsf{lca}(\ell_u, \ell_w) = \mathsf{lca}(\ell_v, \ell_w)$. What we need to show is that these three leaves all appear in M_i. If this was not the case, we would have ℓ_w appearing in M_j $(j \neq i)$, which would imply that $w \in B'$, contradicting the hypothesis. It follows that ℓ_u, ℓ_v, ℓ_w all appear in M_i, thus $uv|w$ is consistent with M_i.

Next, we show that (A_1, A_2) is a nice split of (A, B). Let $u, v, w \in A$. Suppose $u \in A_i \backslash A_j$, $v \in A_j \backslash A_i$, $w \notin B$ with $i \neq j$. If $\mathcal{R}$ contained the rooted triplet $uv|w$, then $uv|w$ would be consistent with M since M complies with (A, B) and $w \notin B$. But this is impossible since u only appears in M_i and v only appears in M_j.

(2) ⇒ (1): To prove that M complies with (A, B), consider any $uv|w \in \mathcal{R}$ with $u, v, w \in A$ and $w \notin B$ and show that $uv|w$ is always consistent with M. There are four (partially overlapping) cases:

1. $u, v, w \in A_i$ and $w \notin B'$: Then $w \notin B_i \cup B'$. Since M_i complies with $(A_i, B_i \cup B')$, we conclude that $uv|w$ is consistent with M_i, and thus with M.
2. $u, v \in A_i, w \in A_j$ with $i \neq j$: Then u, v appear in M_i and w appears in M_j, hence $uv|w$ is consistent with M.
3. $u, w \in A_i, v \in A_j$ with $i \neq j$: We have three mutually exclusive subcases.
   - $u, v \notin B'$: Then $\mathcal{R}$ contains $uv|w$ with $u \in A_i \backslash A_j, v \in A_j \backslash A_i$ and $w \notin B$. This contradicts the assumption that (A_1, A_2) is a nice split of (A, B).
   - $v \in B'$: Then $u, v, w \in A_i$, and we are in Case 1.
   - $u \in B'$: Then $u, v \in A_j, w \in A_i$, and we are in Case 2.
4. $v, w \in A_i, u \in A_j$ with $i \neq j$: This case is symmetric to the previous case. □

Lemma 4 yields recurrence relations for $n(A, B)$ as stated in Lemma 5 below. Say that a split (A_1, A_2) of (A, B) is *proper* if and only if either: (i) A_1, A_2 are proper subsets of A; or (ii) $A_i = A$ and $B' \not\subseteq B$, where $B' = A_1 \cap A_2$.

Lemma 5. *The following recurrence relations for $n(A, B)$ hold:*

1. *Let (A, B) be a pair with $|A| \leq 2$. Then $n(A, B) = 0$.*
2. *Let (A, B) be a pair with $B = A$. Then $n(A, B) = 0$.*
3. *Let (A, B) be a pair with $|A| \geq 3$ and $B \subset A$. Given a split $S = (A_1, A_2)$ of (A, B), let $B' = A_1 \cap A_2, B_i = B \cap A_i$, and define $m(S) = |B'| + n(A_1, B_1 \cup B') + n(A_2, B_2 \cup B')$. Then $n(A, B)$ equals the minimum of the values $m(S)$ over all nice splits S of (A, B) which are proper.*

Lemma 5 allows us to compute $n(A, B)$ by dynamic programming on the pairs ordered by: $(A, B) \leq (A', B')$ if and only if $|A| < |A'|$ or $(|A| = |A'|$ and $|B| \geq |B'|)$. This yields a dynamic programming algorithm for solving SMRT. At the end of the algorithm, $n(L, \emptyset)$ gives the value of an optimal solution, and a corresponding optimal MUL tree can be obtained by performing a traceback.

Theorem 4. SMRT *can be solved using $O^*(7^n)$ time and $O(3^n)$ space.*

Proof. To prove the correctness of the algorithm, we can verify that the definition of the relation $\leq$ on pairs is compatible with the above relations. Indeed, when computing $n(A, B)$ in point 3, we recursively call $n(A_i, B_i \cup B')$. Then: (i) either $A_i \subset A$, in which case we have $(A_i, B_i \cup B') < (A, B)$; (ii) or $A_i = A$, then $B' \not\subseteq B$ since the split is proper, therefore $B \subset B_i \cup B'$ and $(A_i, B_i \cup B') < (A, B)$.

We now analyze the complexity of the algorithm. Fix an integer $p \leq n$. For any $A \subseteq L$ of size p, there are 2^p pairs (A, B), so the number of pairs (A, B) with $|A| = p$ is $\binom{n}{p}2^p$. It follows that the total number of pairs considered is $\sum_{p=0}^{n} \binom{n}{p}2^p = 3^n$, giving the claimed space complexity. Next, for any pair (A, B) with $|A| = p$, there are 3^p splits to consider, and each split is processed in $O(n^3)$ time (i.e., the time required to check that the split is nice and to perform the set operations). Hence, the time complexity is $O(\sum_{p=0}^{n} \binom{n}{p}2^p 3^p n^3) = O(7^n n^3)$. □

References

1. Aho, A.V., Sagiv, Y., Szymanski, T.G., Ullman, J.D.: Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM Journal on Computing 10(3), 405–421 (1981)
2. Brown, G.K., Nelson, G., Ladiges, P.Y.: Historical biogeography of *Rhododendron* section *Vireya* and the Malesian Archipelago. Journal of Biogeography 33, 1929–1944 (2006)
3. Byrka, J., Guillemot, S., Jansson, J.: New results on optimizing rooted triplets consistency. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 484–495. Springer, Heidelberg (2008)
4. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem (Article 21). Journal of the ACM 55(5) (2008)
5. Feige, U., Kilian, J.: Zero knowledge and the chromatic number. Journal of Computer and System Sciences 57(2), 187–199 (1998)
6. Fellows, M., Hallett, M., Stege, U.: Analogs & duals of the MAST problem for sequences & trees. Journal of Algorithms 49(1), 192–216 (2003)
7. Garey, M., Johnson, D.: Computers and Intractability – A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
8. Huber, K.T., Oxelman, B., Lott, M., Moulton, V.: Reconstructing the evolutionary history of polyploids from multilabeled trees. Molecular Biology and Evolution 23(9), 1784–1791 (2006)
9. Neumann-Lara, V.: The dichromatic number of a digraph. Journal of Combinatorial Theory, Series B 33(3), 265–270 (1982)
10. Page, R.D.M., Charleston, M.A.: Trees within trees: phylogeny and historical associations. Trends in Ecology & Evolution 13(9), 356–359 (1998)
11. Scornavacca, C., Berry, V., Ranwez, V.: From gene trees to species trees through a supertree approach. In: Proc. of the 3rd Int. Conference on Language and Automata Theory and Applications (LATA 2009). LNCS, vol. 5457, pp. 702–714. Springer, Heidelberg (2009)
12. Seymour, P.D.: Packing directed circuits fractionally. Combinatorica 15(2), 281–288 (1995)
13. Zuckerman, D.: Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. Theory of Computing 3(1), 103–128 (2007)