Computing a Smallest Multilabeled Phylogenetic Tree from Rooted Triplets

Sylvain Guillemot, Jesper Jansson, and Wing-Kin Sung

Abstract—We investigate the computational complexity of inferring a smallest possible multilabeled phylogenetic tree (MUL tree) which is consistent with each of the rooted triplets in a given set. This problem has not been studied previously in the literature. We prove that even the very restricted case of determining if there exists a MUL tree consistent with the input and having just one leaf duplication is an NP-hard problem. Furthermore, we show that the general minimization problem is difficult to approximate, although a simple polynomial-time approximation algorithm achieves an approximation ratio close to our derived inapproximability bound. Finally, we provide an exact algorithm for the problem running in exponential time and space. As a by-product, we also obtain new, strong inapproximability results for two partitioning problems on directed graphs called Acyclic Partition and Acyclic Tree-Partition.

Index Terms—Phylogenetics; MUL tree; rooted triplet; acyclic tree-partition; inapproximability; dynamic programming.

1 Introduction

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 *singlelabeled phylogenetic tree*; similarly, a phylogenetic tree where each leaf label may occur more than once is called a *multilabeled phylogenetic tree*, or *MUL tree* for short [9], [10], [15], [16], [17], [21].[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 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, $\mathrm{lca}(u, v)$ denotes the lowest common ancestor (lca) of u and v. For convenience, every node is regarded to be an ancestor of itself, and the notation $u \prec v$ means that v is a *proper* ancestor of u, i.e., an ancestor of u which is not u. A phylogenetic tree in which every internal node has exactly two children is called *binary*, and a *rooted triplet* is a binary phylogenetic tree with exactly three distinctly labeled leaves. The unique rooted triplet on a leaf label set $\{x, y, z\}$ satisfying $\mathrm{lca}(\ell_x, \ell_y) \prec \mathrm{lca}(\ell_x, \ell_z) = \mathrm{lca}(\ell_y, \ell_z)$, where ℓ_x, ℓ_y, and ℓ_z are the three leaves labeled by x, y, and z, respectively, is denoted by $xy|z$. If $xy|z$ is an embedded subtree of a MUL tree M in the sense that there exist three leaves ℓ_x, ℓ_y, ℓ_z in M labeled by x, y, z such that $\mathrm{lca}(\ell_x, \ell_y) \prec \mathrm{lca}(\ell_x, \ell_z) = \mathrm{lca}(\ell_y, \ell_z)$ 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 new algorithmic 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 $d(M)$.[2] Note that for any given instance of SMRT, there is always at least one optimal solution which is binary. We also consider the corresponding 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 $d(M) \leq d$?

From here on, we define $k = |\mathcal{R}|$ and $n = |L|$ for any given instance of SMRT or d-SMRT. 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 $\mathcal{R}$.

1.1 Motivation

The problem of determining whether there exists a *singlelabeled* phylogenetic 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. [2].[3] 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) [3], [12], [18]), or build a *phylogenetic network* (an extension of a phylogenetic tree in which internal nodes may have more than a single parent) which contains all of the rooted triplets. See [5] for a recent survey of related results and many references. In this paper, we introduce 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 in a tree. For example, a gene tree can contain several leaves labeled by the same species due to gene duplication events [9], [15], [16], [17], [21]. 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 and are widely used in Biogeography to infer clues about ecological processes and events that affect the geographic distribution of organisms (see, e.g., [4], [10], [15], [17]). MUL trees are also often employed to study host-parasite cospeciation [15], [17], [20]. In short, MUL trees are not only a natural, but also a versatile and useful generalization of singlelabeled phylogenetic trees.

Our motivation for developing new algorithms for constructing MUL trees comes from the final discussion in [17] where Huber

1. MUL trees are called *rl-trees* in [9] and *area cladograms* in [10].

2. Here, "smallest" refers to the number of leaf duplications. To infer a singlelabeled phylogenetic tree consistent with a given set of rooted triplets and having as few internal nodes as possible (a so-called "minimally resolved supertree") is a different problem, recently studied in [24].

3. The running time of the original implementation of the algorithm of Aho et al. [2] was $O(nk)$. Henzinger et al. [13] later presented a faster implementation of this algorithm, and replacing the dynamic graph connectivity data structure used by [13] by a more recent one [14] further reduces the complexity of the algorithm to $\min\{O(n + k \log^2 n),\ O(k + n^2 \log n)\}$ time [18].

- *S. Guillemot is with the Institut Gaspard Monge - Université Paris-Est, 5 boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée, France. E-mail: Sylvain.Guillemot@univ-mlv.fr, guillemo@free.fr.*
- *J. Jansson is with Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan. E-mail: Jesper.Jansson@ocha.ac.jp.*
- *W.-K. Sung is with the School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, and Genome Institute of Singapore, 60 Biopolis Street, Genome, Singapore 138672. E-mail: ksung@comp.nus.edu.sg.*

Manuscript received 5 Nov. 2009; revised 21 Jan. 2010; accepted 25 Jan. 2010; published online 20 Aug. 2010.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2009-11-0204. Digital Object Identifier no. 10.1109/TCBB.2010.77.

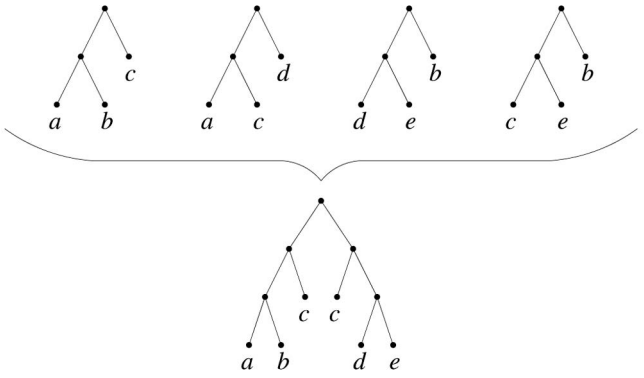

Fig. 1. The set of rooted triplets $\{ab|c,\ ac|d,\ de|b,\ ce|b\}$ is consistent with a MUL tree containing one leaf duplication.

et al. conclude: "More importantly, more work needs to be done concerning the inference of MUL trees from a set of gene trees and, in particular, how to root such trees as the network construction heavily relies on the position of the root." Using rooted triplets as input may be helpful here because computationally expensive techniques such as maximum likelihood-oriented methods can often yield accurate trees in reasonable time for small subsets of the objects being studied (in particular, for subsets of cardinality three [7]).

1.2 Our Results and Organization of the Paper

We present the first negative and positive results regarding the computational complexity and polynomial-time approximability of SMRT. Significantly, even the severely restricted case of determining if there exists a MUL tree consistent with the input and having just one leaf duplication turns out to be an NP-hard problem. (In contrast, when leaf duplications are not allowed, the corresponding problem can be solved in polynomial time by the algorithm of Aho et al. [2], as mentioned in Section 1.1.) Moreover, we show that the general case of SMRT is hard to approximate in polynomial time by proving strong inapproximability bounds for a problem on directed graphs named ACYCLIC TREE-PARTITION (defined in Section 3.1) and then describing a measure-preserving reduction from ACYCLIC TREE-PARTITION to SMRT. To alleviate these negative results, we give a polynomial-time approximation algorithm for SMRT whose performance is very close to the derived inapproximability bound, as well as an exact, exponential-time algorithm based on dynamic programming over pairs of subsets of the leaf labels.

The rest of the paper is organized as follows: Section 2 provides 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 that SMRT cannot be approximated within a ratio of $n^{1-\epsilon}$ for any constant $0 < \epsilon \leq 1$ in polynomial time, unless $\mathrm{P} = \mathrm{NP}$. (Section 3 also gives new inapproximability results for the ACYCLIC PARTITION and ACYCLIC TREE-PARTITION problems.) Next, Section 4 presents an exact algorithm for SMRT which runs in $O^*(7^n)$ time and $O(3^n)$ space. Finally, Section 5 mentions some recent algorithmic results for other related problems involving MUL trees.

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 $\mathcal{R}$.*

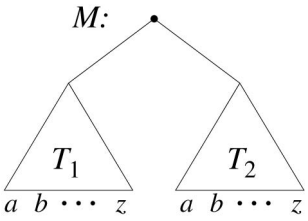

Fig. 2. The MUL tree M is consistent with every rooted triplet over the leaf label set $\{a, b, \ldots, z\}$.

Proof. Let T be an arbitrary singlelabeled 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 their roots to a new parent root node, as illustrated in Fig. 2.

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. [2] (see Section 1.1), determine if there exists a singlelabeled tree consistent with $\mathcal{R}$. If the answer is positive 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. More precisely, we show 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 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

In this section, we define the ACYCLIC PARTITION and ACYCLIC TREE-PARTITION problems and determine their computational complexity.

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 suggest 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 $ap(D)$.[4] Similarly, the ACYCLIC TREE-PARTITION problem seeks an

4. $ap(D)$ is also referred to in the literature as *the dichromatic number of D*. It was introduced by Neumann-Lara in [19].

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 several 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.*

Proof. Follows from Definition 2. □

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

Proof. $atp(D) \leq ap(D) - 1$: Consider any acyclic partition of D into q classes $V_1, \ldots, V_q$ and let T be an arbitrary binary tree with q leaves $l_1, \ldots, l_q$. Let N denote the set of nodes in T and define the partition $P = \{V(x) : x \in N\}$ so that $V(l_i) = V_i$ for every $i \in \{1, \ldots, q\}$ and $V(x) = \emptyset$ for every internal node x of T. Then, (T, P) is an acyclic tree-partition of D with $q - 1$ internal nodes since T is binary.

$ap(D) \leq atp(D) + 1$: Let $(T, \{V(x) : x \in N\})$ be any acyclic tree-partition of D, where N denotes the set of nodes in T. Let q be the number of internal nodes in T; then, T contains $q + 1$ leaves because T is binary. Define a set V'_ℓ for every leaf ℓ of T, where $V'_\ell \subseteq V$, by applying the following procedure:

Initially, let $S := \emptyset$. For each leaf ℓ of T (in arbitrary order): let X_ℓ be the set of all nodes which are ancestors of ℓ and not already in S, define $V'_\ell := \bigcup_{x \in X_\ell} V(x)$, and let $S := S \cup X_\ell$.

Each vertex in V belongs to exactly one of the sets V'_ℓ. Moreover, each V'_ℓ induces an acyclic subgraph of D by Lemma 2. Therefore, $\{V'_\ell : \ell \text{ is a leaf of } T\}$ forms an acyclic partition of D with $q + 1$ classes. □

Next, we derive hardness results for the problems defined above.

In [19], Neumann-Lara noted that the chromatic number of any undirected graph $G = (V, E)$ equals $ap(G^*)$, where $G^* = (V, A)$ is the directed graph obtained by replacing each undirected edge $\{u, v\}$ of E by two arcs $(u, v), (v, u)$. Since GRAPH K-COLORABILITY is NP-hard for any fixed positive integer $K \geq 3$ (see, e.g., [11]), Neumann-Lara's observation immediately implies that r-ACYCLIC PARTITION is NP-hard for any fixed positive integer $r \geq 3$. In Case (i) of Theorem 2, we establish an even tighter NP-hardness result by reducing from a different problem; nevertheless, the simple reduction of Neumann-Lara is still useful as it yields the strong inapproximability bounds for ACYCLIC PARTITION in Case (ii) of Theorem 2.

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 $\mathrm{P} = \mathrm{NP}$, 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 [11]. 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, let D contain 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 $C'_j = \neg x$, let D contain 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: for any 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) For any given undirected graph $G = (V, E)$, the reduction of Neumann-Lara [19] constructs the directed graph $G^* = (V, A)$ by replacing each undirected edge $\{u, v\}$ of E by two arcs $(u, v), (v, u)$. For any $V' \subseteq V$, V' is an independent set of G if and only if V' induces an acyclic subgraph of G^*; thus, colorings of G correspond to acyclic partitions of G^*. It follows that the above reduction is a measure-preserving reduction from CHROMATIC NUMBER to ACYCLIC PARTITION, and therefore known inapproximability results for CHROMATIC NUMBER [8], [23] carry over directly to ACYCLIC PARTITION. □

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 $\mathrm{P} = \mathrm{NP}$, where n is the number of vertices in the input graph.

Proof. Use the same reductions from NOT-ALL-EQUAL 3SAT and CHROMATIC NUMBER as in the proofs of Cases (i) and (ii) of Theorem 2, and apply Lemma 3. □

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 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 $U \subseteq L$. The labels belonging to U are called* unique labels. *An MUL tree M is* consistent *with the pair $(\mathcal{R}, U)$ if: 1) M is consistent with $\mathcal{R}$; and 2) 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 minimum number of leaf duplications. We have:

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 an instance $(\mathcal{R}, U)$ of C-SMRT with leaf 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) ⇒ 2): Suppose D has an acyclic tree-partition consisting of a binary 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).

Let us describe the construction of M formally. First, introduce the following additional notation: given a MUL tree M and a sequence of labels $s = x_1, \ldots, x_n$, let $R(M, s)$ be the tree obtained by starting with a caterpillar with $n + 1$ leaves $l_0, \ldots, l_n$ (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: 1) if x is a leaf then M_x consists of a single leaf labeled by z; 2) if x is an internal node with two children y, y' then M_x is the MUL tree obtained by joining M'_y and $M'_{y'}$ to a common parent root node; and 3) 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 define

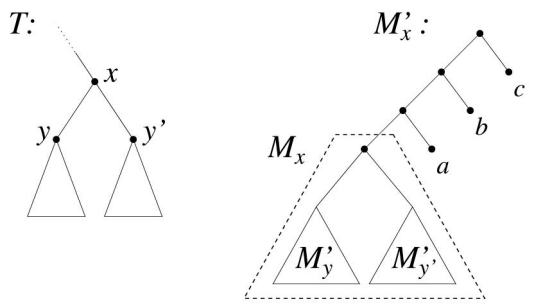

Fig. 3. Suppose that node x has two children y, y' in T, and that $V_x = \{a, b, c\}$ with $a < b < c$ in the topological ordering s_x. Then, M_x is obtained by joining M'_y and $M'_{y'}$ to a new parent node, and $M'_x := R(M_x, s_x)$. Note that both of M'_y and $M'_{y'}$ contain at least one leaf labeled by z.

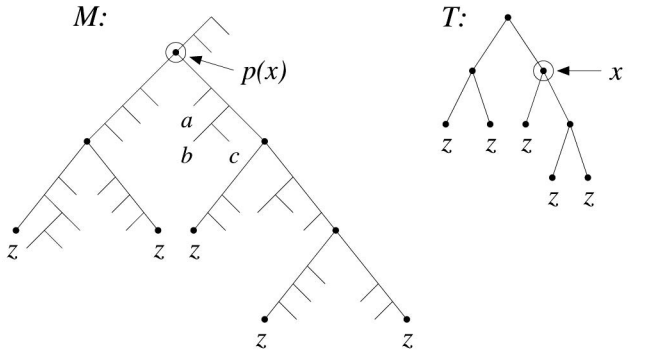

Fig. 4. T is the topological restriction of M to leaves labeled by z. The node $p(x)$ for the marked node x is shown. In this example, $V_x = \{a, b, c\}$.

$M'_x := R(M_x, s_x)$. See Fig. 3 for an illustration of points 2) and 3). Finally, let $M := M'_t$, where t is the root of T.

Now consider the constructed MUL tree M. Clearly, only z is duplicated in M; since $\{V_x : x \in N\}$ is a partition of V, each vertex from V_x will appear only once in M (namely, as the label of a leaf which is directly attached to the path between the root of M'_x and the root of M_x). Also observe that the leaves of M labeled by z correspond to the leaves of T, so their number is $r + 1$; hence M has r duplications. Next, we show by case analysis that M is indeed consistent with $\mathcal{R}$. Consider any $zu|v \in \mathcal{R}$. 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:

- $x = y$: Then both u and v belong to V_x. Consider $M'_x = R(M_x, s_x)$. M_x contains a leaf labeled by z. Furthermore, since $(u, v) \in A$ and s_x is a topological ordering of $D[V_x]$, it follows that u precedes v in s_x. Then, M'_x (and thus M) is consistent with $zu|v$.
- $x \prec y$ in T: Consider $M'_y = R(M_y, s_y)$. The condition $x \prec y$ in T implies that M'_x is a subtree of M_y and so M_y contains leaves labeled by z and u. On the other hand, v appears in s_y. Therefore, M'_y (and thus M) is consistent with $zu|v$.
- $y \prec x$ in T: This is impossible according to Point 2 of Definition 2.
- Both $x \not\prec y$ and $y \not\prec x$ in T: Let $c = \mathrm{lca}(x, y)$ 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 the MUL tree M_c obtained by joining M'_{c_x} and M'_{c_y} to a common parent root node. M'_{c_x} contains leaves labeled by z and u, while M'_{c_y} contains a leaf labeled by v; hence M_c (and M) is consistent with $zu|v$.

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 be the topological restriction of M to leaves labeled by z, i.e., the binary tree obtained from M by deleting all nodes which are not on any path from the root to a leaf labeled by z along with their incident edges, and then contracting every edge between a node having just one child and its child (see Fig. 4). For each node x in T, define $p(x)$ as the node in M corresponding to the parent of x in T; in case x is the root of T then define $p(x)$ to be an imaginary parent node of the root of M. Next, for each node x in T, let V_x consist of every label y from V such that $\mathrm{lca}(x, y)$ in M is different from x and such that $p(x)$ is a proper ancestor of $\mathrm{lca}(x, y)$ in M. (Thus, each V_x contains those labels from V that are attached along the path in M between $p(x)$ and the node corresponding to x.)

Let N be the set of nodes of T. We now show that $(T, \{V_x : x \in N\})$ is an acyclic tree-partition of D with r internal nodes. The leaves of T correspond to the leaves in M labeled by z, and there are $r + 1$ such leaves; therefore, it is clear that T has r internal nodes. Moreover, T is binary and $\{V_x : x \in N\}$ forms a partition of V since each label of V occurs only once in M. It remains to verify Points 1 and 2 of Definition 2:

- Point 1: To show that $D[V_x]$ is acyclic for any $x \in N$, let $T_1, \ldots, T_q$ be the subtrees of M hanging along the path between the node corresponding to x in M and $p(x)$, and numbered according to increasing distance from x. Then, the trees T_i are singlelabeled, have disjoint label sets, and the union of their label sets is V_x. Let s_x be any linear ordering of V_x which ranks the elements of $\mathcal{L}(T_i)$ before the elements of $\mathcal{L}(T_{i+1})$ for each $1 \leq i < q$. We need to prove that s_x is in fact a topological ordering of $D[V_x]$. For this purpose, suppose that (u, v) is an arc of $D[V_x]$. Observe that if u appears in T_i and v appears in T_j with $j \leq i$ then $zu|v$ cannot be consistent with M (since T_i does not contain z). However, $zu|v \in \mathcal{R}$ by the construction of $\mathcal{R}$, and since $zu|v$ is consistent with M, this implies by the above observation that $u \in \mathcal{L}(T_i)$ and $v \in \mathcal{L}(T_j)$ with $i < j$. Hence, u appears before v in s_x, i.e., s_x is a topological ordering of $D[V_x]$, so $D[V_x]$ is acyclic.
- Point 2: Consider any $x, y \in N$ with $x \prec y$, and let P be the path in M joining the node corresponding to x to the root of M (note that P must pass through the node corresponding to y). Now, if $u \in V_x$ and $v \in V_y$, there exist two disjoint singlelabeled trees T_A, T_B attached along P such that T_B contains the only occurrence of u, T_A contains the only occurrence of v, and T_A is above T_B. Suppose by contradiction that D contains an arc (v, u). Then, by the construction of $\mathcal{R}$, we have $zv|u \in \mathcal{R}$. But the only way for M to be consistent with $zv|u$ is if z appears in T_A, which is not the case. We conclude that D cannot contain an arc from V_y to V_x.

□

We next describe a reduction from the constrained to the unconstrained variant of SMRT.

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: 1) for each $x \in U$, labels x_i

($1 \le i \le n+1$); and 2) 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 $r \le n$. 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 singlelabeled binary tree T_u over $\{x_1, \ldots, x_j\}$, where j equals either 1 or $n+1$. Observe that: 1) for each $x \in U$, each label x_i occurs exactly once in M'; and 2) 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 $x_i y_j | z_k \in \mathcal{R}'$, there exist three leaves u, v, w in M labeled by x, y, z, respectively, such that the corresponding $xy|z \in \mathcal{R}$ is consistent with M; by selecting the leaves in M' labeled by x_i, y_j, z_k in subtrees T_u, T_v, T_w, we see that $x_i y_j | z_k$ is consistent with M'. This proves that M' is consistent with every triplet in $\mathcal{R}'$.

($\Leftarrow$): Let M' be a MUL tree with r duplications consistent with $\mathcal{R}'$. For each $x \in L$, define $i_x \in \{1, \ldots, n+1\}$ as follows: if $x \in U$, let i_x be an index such that the leaf label x_{i_x} is not duplicated in M' (since M' has $r < n+1$ duplications, for each $x \in U$ there exists at least one such index); if $x \in L \backslash U$, let $i_x = 1$. Next, let M'' be a subtree of M' such that: 1) for each $x \in U$, M'' contains the unique occurrence of x_{i_x} and no occurrences of x_j for $j \ne i_x$; and 2) for each $x \in L \backslash U$, M'' contains every occurrence of x_1. Finally, for each leaf x_{i_x} in M'', change its label to x, and let M be the resulting MUL tree.

First note that M has $\le r$ duplications and the labels of U are not duplicated. Indeed, for any $x \in U$, M has only one occurrence of x, while 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 $d(M) \le d(M') = r$.

Second, consider any triplet $xy|z \in \mathcal{R}$, and let $i = i_x$, $j = i_y, k = i_z$. Then, $x_i y_j | z_k \in \mathcal{R}'$, and this triplet must be present in M' since M' is consistent with $\mathcal{R}'$. Thus, there exist leaves ℓ_x, ℓ_y, ℓ_z in M' labeled by x_i, y_j, z_k such that $\mathrm{lca}(\ell_x, \ell_y) \prec \mathrm{lca}(\ell_x, \ell_z) = \mathrm{lca}(\ell_y, \ell_z)$. By the definition of i, j, k, these leaves are also present in M with the same relationships, but having the labels x, y, z. We conclude that $xy|z$ is consistent with M.

Therefore, M is a MUL tree with $\le r$ duplications which is consistent with $(\mathcal{R}, U)$. □

Propositions 1 and 2 together with the 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 $0 < \epsilon \le 1$ in polynomial time, unless* P = NP.

We remark that the analogous MINIMUM DUPLICATION SUPERSEQUENCE problem [9] 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 [6]) and approximable within $O(\log n \log\log n)$ in polynomial time (by a result of [22]).

4 An Exact Algorithm for SMRT

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

Let $\mathcal{R}$ be a given set of rooted triplets over a leaf label set L. We use a dynamic programming approach, exploiting the recursive structure of the problem as follows: if a binary MUL tree M is an optimal solution for $\mathcal{R}$, then its two child subtrees M_1, M_2 should be optimal solutions for some subproblems. A first idea would be to use the leaf label sets directly to define suitable subproblems, but difficulties arise with this approach for two main reasons. First, the two child subtrees M_1, M_2 may have overlapping leaf label sets, and it is not clear how to check the consistency of labels in the intersection. Second, it is possible that one M_i has the same leaf label set as M (i.e., not a proper subset), thus we cannot ensure that a given subproblem is broken into strictly smaller subproblems.

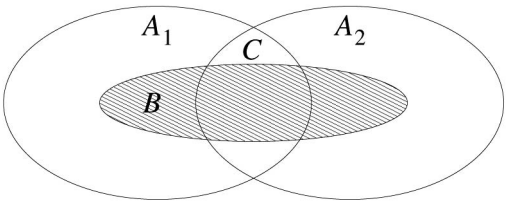

Fig. 5. A Venn diagram illustrating the relationships between the sets A_1, A_2, B, C, where (A_1, A_2) is a split of (A, B) and C is defined by $C = A_1 \cap A_2$. If both A_1 and A_2 are proper subsets of A then (A_1, A_2) is called a proper split according to Case 1 in Definition 7.

To handle these issues, we do dynamic programming on *pairs of subsets*. More precisely, we consider pairs of subsets of L of the form (A, B) such that $B \subseteq A \subseteq L$. For a given pair (A, B), 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.*

Intuitively, by selecting a subset A of L, we focus on rooted triplets in $\mathcal{R}$ involving leaf labels from A only. The specified subset B of A then further allows certain triplets over the leaf label set A to be "ignored"; a MUL tree that complies with (A, B) does not have to be consistent with the triplets from $\mathcal{R}$ of the form $\cdot\cdot|w$ where $w \in B$.

Subproblems in our dynamic programming approach correspond to pairs (A, B) with $B \subseteq A \subseteq L$. For any 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. The base cases are when $|A| \le 2$ or $B = A$, and we 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 introduce a few more definitions.

Definition 5. *Let (A, B) be a pair such that $B \subseteq A \subseteq L$. A* split *of (A, B) is a pair (A_1, A_2) of subsets of A such that $A_1 \cup A_2 = A$.*

Definition 6. *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 $uv|w \in \mathcal{R}$ and $u \in A_i \backslash A_j$, $v \in A_j \backslash A_i$ with $i \ne j$ then $w \in B$.*

Observe that A_1, A_2 in Definition 5 are not necessarily disjoint, and that the definition does not actually depend on B. Also, B may intersect with both A_1 and A_2, as shown in Fig. 5. From here on, we let B_i denote the intersection of B with A_i, and define $C = A_1 \cap A_2$.

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 $|A| \ge 2$, 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 $A_1 = \mathcal{L}(M_1)$,*

$A_2 = \mathcal{L}(M_2)$, $C = A_1 \cap A_2$, *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 $(A_i, B_i \cup C)$.*

Proof. 1) $\Rightarrow$ 2): Suppose that M complies with (A, B). We first show that M_i complies with $(A_i, B_i \cup C)$. Suppose that $uv|w \in \mathcal{R}$ with $u, v, w \in A_i$ and $w \notin B_i \cup C$. 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 $\mathrm{lca}(\ell_u, \ell_v) \prec \mathrm{lca}(\ell_u, \ell_w) = \mathrm{lca}(\ell_v, \ell_w)$. 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 $w \in C$, 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 that $u \in A_i \backslash A_j$, $v \in A_j \backslash A_i$ with $i \neq j$, and $w \notin B$. If $\mathcal{R}$ contained the rooted triplet $uv|w$, then $uv|w$ would be consistent with M since M complies with (A, B) and since $w \notin B$. But this is impossible since u only appears in M_i and v only appears in M_j.

2) $\Rightarrow$ 1): To prove that M complies with (A, B), take 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 $w \notin C$: Then, $w \notin B_i \cup C$. Since M_i complies with $(A_i, B_i \cup C)$, 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 $i \neq j$: For this case, we have the following three possible subcases.
   a. $u, v \notin C$: 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).
   b. $v \in C$: Then, $u, v, w \in A_i$, and we are in Case 1.
   c. $u \in C$: 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.

Definition 7. *A split (A_1, A_2) of (A, B) is a* proper split *if and only if either: 1) both A_1 and A_2 are proper subsets of A; or 2) $A_i = A$ and $A_j \not\subseteq B$, where $i \neq j$.*

The two possible cases of Definition 7 are displayed in Figs. 5 and 6, respectively.[5]

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 $B \subsetneq A$. Given a split $S = (A_1, A_2)$ of (A, B), let $C = A_1 \cap A_2$, $B_i = B \cap A_i$, and define $m(S) = |C| + n(A_1, B_1 \cup C) + n(A_2, B_2 \cup C)$. Then, $n(A, B)$ equals the minimum value of $m(S)$ taken over every nice split S of (A, B) which is proper.*

5. The reason why the case $A_i = A$ and $A_j \subseteq B$ with $i \neq j$ is excluded from the definition is that here $B_i = B \cap A_i = B$ and $C = A_i \cap A_j = A_j$, giving us $(A_i, B_i \cup C) = (A, B \cup A_j) = (A, B)$, which would not take us any closer to the base cases of the recursion.

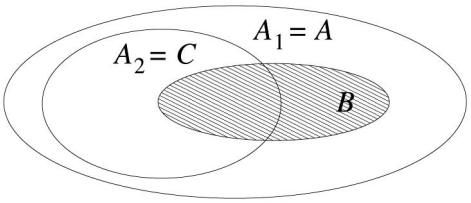

Fig. 6. Illustrating case 2 in Definition 7. When $A_1 = A$ or $A_2 = A$, the split (A_1, A_2) is a proper split if $C \not\subseteq B$.

Proof. The first two points are immediate. Let us prove Point 3. Let $n'(A, B)$ denote the minimum value of $m(S)$ taken over every nice split S of (A, B) which is proper.

We first show that $n(A, B) \leq n'(A, B)$. Consider a nice split $S = (A_1, A_2)$ of (A, B) which is proper, and such that $n'(A, B) = m(S)$. For $i \in \{1, 2\}$, let M_i be a MUL tree complying with $(A_i, B_i \cup C)$ and having a minimum number of leaf duplications, and let (M_1, M_2) be a MUL tree obtained by joining M_1 and M_2 to a common parent root node. By Lemma 4, $M = (M_1, M_2)$ complies with (A, B), thus

$$\begin{aligned} n(A, B) &\leq d(M) = |C| + d(M_1) + d(M_2) \\ &= |C| + n(A_1, B_1 \cup C) + n(A_2, B_2 \cup C) = m(S) = n'(A, B). \end{aligned}$$

We next show that $n'(A, B) \leq n(A, B)$. Let M be a binary MUL tree complying with (A, B), and having a minimum number of nodes. Since $|A| \geq 2$, we have $M = (M_1, M_2)$. Let $A_i = \mathcal{L}(M_i)$ for $i \in \{1, 2\}$. Then, $S = (A_1, A_2)$ is a split of (A, B). By Lemma 4, S is a nice split of (A, B) and M_i complies with $(A_i, B_i \cup C)$, implying that $n(A_i, B_i \cup C) \leq d(M_i)$.

To see that S is proper, suppose on the contrary that $A_i = A$ and $C \subseteq B$. Then, M_i complies with $(A_i, B_i \cup C) = (A, B)$, contradicting the minimality of M.

We conclude that $n'(A, B) \leq m(S) = |C| + n(A_1, B_1 \cup C) + n(A_2, B_2 \cup C) \leq |C| + d(M_1) + d(M_2) = d(M) = n(A, B)$, and the result follows. □

At each level of the recursion, $|A|$ decreases or $|B|$ increases, so Lemma 5 allows us to compute $n(A, B)$ in bottom-up order over all 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 in $O^*(7^n)$ time and $O(3^n)$ space.*

Proof. To prove the correctness of the algorithm, we verify that the definition of the relation $\leq$ on pairs is compatible with the above relations. Indeed, whenever computing $n(A, B)$ according to Point 3 in Lemma 5, we recursively call $n(A_i, B_i \cup C)$. Then either: 1) $A_i \subsetneq A$, in which case we have $(A_i, B_i \cup C) < (A, B)$; or 2) $A_i = A$, then $C \not\subseteq B$ since the split is proper, therefore $B \subsetneq B_i \cup C$ and $(A_i, B_i \cup C) < (A, B)$.

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 each 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 $O(\sum_{p=0}^{n}\binom{n}{p}2^p 3^p n^3) = O(7^n n^3)$. □

5 Related Work

Although the problem of inferring a MUL tree from an input set of singlelabeled phylogenetic trees that minimizes the number of leaf duplications has not been studied before, Huber et al. [15] recently introduced another approach to inferring MUL trees based on *bipartitions of a multiset*, and asked: can a given collection of bipartitions of a multiset be represented by an unrooted MUL tree? They proved the NP-hardness of a restricted case of the problem, and gave a fixed-parameter algorithm for the general problem in terms of a parameter associated to the given multiset that counts the total number of duplications in the multiset. Note that the problem studied in [15] differs from SMRT; for example, a collection of bipartitions of a multiset might not be representable by any MUL tree at all (see Section 1 in [15]), whereas any set of singlelabeled phylogenetic trees can trivially be merged into a consistent MUL tree just by attaching all the trees to a new parent root node.

Also related to this line of research are several recently published combinatorial algorithms for *manipulating already-known MUL trees*:

- Huber et al. [17] 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 (see [16] for the precise mathematical definition of "to exhibit").
- Ganapathy et al. [10] gave algorithms for identifying common patterns in two MUL trees based on maximum agreement subtrees, and presented a simple linear-time algorithm for checking if two input MUL trees are isomorphic which extends the classical tree isomorphism algorithm in [1].
- Scornavacca et al. [21] considered some computational problems involving the extraction of the unambiguous parts of an input MUL tree. More precisely, [21] proposed linear-time algorithms to identify every so-called observed duplication node in a MUL tree, testing if two MUL trees are isomorphic (using a different idea than the algorithm of Ganapathy et al. [10] mentioned above), and computing a largest duplication-free rooted subtree of a MUL tree. They also showed that it is an NP-hard problem to prune all of the MUL trees in a given set at observed duplication nodes to singlelabeled trees in such a way that the obtained set of trees can be merged without conflicts into a singlelabeled tree.

We believe that many interesting combinatorial properties and algorithms for inferring and comparing MUL trees remain to be discovered. In light of the negative results established in this paper, it is probably necessary to consider structurally restricted MUL trees of some kind in order to obtain efficient algorithms for SMRT as well as for other related problems.

Acknowledgments

An extended abstract of this paper appeared in *Proceedings of the 20th Annual International Symposium on Algorithms and Computation (ISAAC 2009)*, volume 5878 of Lecture Notes in Computer Science, pp. 1205-1214, Springer-Verlag, 2009. Jesper Jansson is funded by the Special Coordination Funds for Promoting Science and Technology. He is the corresponding author.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] A.V. Aho, Y. Sagiv, T.G. Szymanski, and J.D. Ullman, "Inferring a Tree from Lowest Common Ancestors with an Application to the Optimization of Relational Expressions," *SIAM J. Computing,* vol. 10, no. 3, pp. 405-421, 1981.

[3] V. Berry and F. Nicolas, "Maximum Agreement and Compatible Supertrees," *J. Discrete Algorithms,* vol. 5, no. 3, pp. 564-591, 2007.

[4] G.K. Brown, G. Nelson, and P.Y. Ladiges, "Historical Biogeography of Rhododendron Section Vireya and the Malesian Archipelago," *J. Biogeography,* vol. 33, no. 11, pp. 1929-1944, 2006.

[5] J. Byrka, S. Guillemot, and J. Jansson, "New Results on Optimizing Rooted Triplets Consistency," *Discrete Applied Math.,* vol. 158, no. 11, pp. 1136-1147, 2010.

[6] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon, "A Fixed-Parameter Algorithm for the Directed Feedback Vertex Set Problem," *J. ACM,* vol. 55, no. 5, article no. 21, 2008.

[7] B. Chor, M. Hendy, and D. Penny, "Analytic Solutions for Three Taxon ML Trees with Variable Rates across Sites," *Discrete Applied Math.,* vol. 155, nos. 6-7, pp. 750-758, 2007.

[8] U. Feige and J. Kilian, "Zero Knowledge and the Chromatic Number," *J. Computer and System Sciences,* vol. 57, no. 2, pp. 187-199, 1998.

[9] M. Fellows, M. Hallett, and U. Stege, "Analogs & Duals of the MAST Problem for Sequences & Trees," *J. Algorithms,* vol. 49, no. 1, pp. 192-216, 2003.

[10] G. Ganapathy, B. Goodson, R. Jansen, H.-S. Le, V. Ramachandran, and T. Warnow, "Pattern Identification in Biogeography," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 3, no. 4, pp. 334-346, Oct.-Dec. 2006.

[11] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[12] S. Guillemot and V. Berry, "Fixed-Parameter Tractability of the Maximum Agreement Supertree Problem," *Proc. 18th Ann. Symp. Combinatorial Pattern Matching (CPM '07),* pp. 274-285, 2007.

[13] M.R. Henzinger, V. King, and T. Warnow, "Constructing a Tree from Homeomorphic Subtrees, with Applications to Computational Evolutionary Biology," *Algorithmica,* vol. 24, no. 1, pp. 1-13, 1999.

[14] J. Holm, K. de Lichtenberg, and M. Thorup, "Poly-logarithmic Deterministic Fully-Dynamic Algorithms for Connectivity, Minimum Spanning Tree, 2-Edge, and Biconnectivity," *J. ACM,* vol. 48, no. 4, pp. 723-760, 2001.

[15] K.T. Huber, M. Lott, V. Moulton, and A. Spillner, "The Complexity of Deriving Multi-Labeled Trees from Bipartitions," *J. Computational Biology,* vol. 15, no. 6, pp. 639-651, 2008.

[16] K.T. Huber and V. Moulton, "Phylogenetic Networks from Multi-Labelled Trees," *J. Math. Biology,* vol. 52, no. 5, pp. 613-632, 2006.

[17] K.T. Huber, B. Oxelman, M. Lott, and V. Moulton, "Reconstructing the Evolutionary History of Polyploids from Multilabeled Trees," *Molecular Biology and Evolution,* vol. 23, no. 9, pp. 1784-1791, 2006.

[18] J. Jansson, J.H.-K. Ng, K. Sadakane, and W.-K. Sung, "Rooted Maximum Agreement Supertrees," *Algorithmica,* vol. 43, no. 4, pp. 293-307, 2005.

[19] V. Neumann-Lara, "The Dichromatic Number of a Digraph," *J. Combinatorial Theory, Series B,* vol. 33, no. 3, pp. 265-270, 1982.

[20] R.D.M. Page and M.A. Charleston, "Trees within Trees: Phylogeny and Historical Associations," *Trends in Ecology & Evolution,* vol. 13, no. 9, pp. 356-359, 1998.

[21] C. Scornavacca, V. Berry, and V. Ranwez, "From Gene Trees to Species Trees through a Supertree Approach," *Proc. Third Int'l Conf. Language and Automata Theory and Applications (LATA '09),* pp. 702-714, 2009.

[22] P.D. Seymour, "Packing Directed Circuits Fractionally," *Combinatorica,* vol. 15, no. 2, pp. 281-288, 1995.

[23] D. Zuckerman, "Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number," *Theory of Computing,* vol. 3, no. 1, pp. 103-128, 2007.

[24] J. Jansson, R.S. Lemence, and A. Lingas, "The Complexity of Inferring a Minimally Resolved Phylogenetic Supertree," *Proc. 10th Int'l Workshop Algorithms in Bioinformatics (WABI '10),* pp. 262-273, 2010.