Rooted Maximum Agreement Supertrees

Jesper Jansson[1], Joseph H.-K. Ng[1], Kunihiko Sadakane[2], and Wing-Kin Sung[1]

[1] School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543. {jansson,nghonkeo,ksung}@comp.nus.edu.sg

[2] Department of Computer Science and Communication Engineering, Kyushu University, Japan. sada@csce.kyushu-u.ac.jp

Abstract. Given a set $\mathcal{T}$ of rooted, unordered trees, where each $T_i \in \mathcal{T}$ is distinctly leaf-labeled by a set $\Lambda(T_i)$ and where the sets $\Lambda(T_i)$ may overlap, the maximum agreement supertree problem (MASP) is to construct a distinctly leaf-labeled tree Q with leaf set $\Lambda(Q) \subseteq \bigcup_{T_i \in \mathcal{T}} \Lambda(T_i)$ such that $|\Lambda(Q)|$ is maximized and for each $T_i \in \mathcal{T}$, the topological restriction of T_i to $\Lambda(Q)$ is isomorphic to the topological restriction of Q to $\Lambda(T_i)$. Let $n = \left|\bigcup_{T_i \in \mathcal{T}} \Lambda(T_i)\right|$, $k = |\mathcal{T}|$, and $D = \max_{T_i \in \mathcal{T}}\left\{\deg(T_i)\right\}$. We first show that MASP with $k = 2$ can be solved in $O\left(\sqrt{D}\, n \log(2n/D)\right)$ time, which is $O(n \log n)$ when $D = O(1)$ and $O(n^{1.5})$ when D is unrestricted. We then present an algorithm for MASP with $D = 2$ whose running time is polynomial if $k = O(1)$. On the other hand, we prove that MASP is NP-hard for any fixed $k \geq 3$ when D is unrestricted, and also NP-hard for any fixed $D \geq 2$ when k is unrestricted even if each input tree is required to contain at most three leaves. Finally, we describe a polynomial-time $(n/\log n)$-approximation algorithm for MASP.

1 Introduction

An important objective in phylogenetics is to develop methods for merging a collection of phylogenetic trees on overlapping sets of taxa into a single supertree so that no (or as little as possible) branching information is lost. Ideally, the resulting supertree can then be used to deduce evolutionary relationships between taxa which do not occur together in any one of the input trees. Supertree methods are useful because most individual studies investigate relatively few taxa [22] and because sample bias leads to certain taxa being studied much more frequently than others [4]. Also, supertree methods can combine trees constructed for different types of data or under different models of evolution. Furthermore, although computationally expensive methods for constructing reliable phylogenetic trees are infeasible for large sets of taxa, they can be applied to obtain highly accurate trees for smaller, overlapping subsets of the taxa that may then be merged using less computationally intense, supertree-based techniques (see, e.g., [7,16,20]).

Since the set of trees which is to be combined may in practice contain contradictory branching structure (for example, if the trees have been constructed from data originating from different genes or if the experimental data contains errors), a supertree method needs to specify how to resolve conflicts. In this paper, we consider *maximum agreement supertrees.* The intuitive idea is to identify

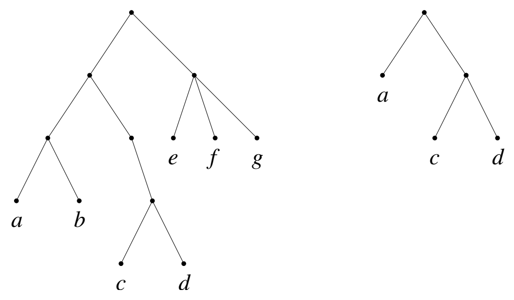

Fig. 1. Let T be the tree on the left. Then $T \mid \{a, c, d, h\}$ is the tree shown on the right.

and remove a smallest possible subset of the taxa so that the remaining taxa can be combined without conflicts. In this way, one would get an indication of which ancestral relationships can be regarded as resolved and which taxa need to be subjected to further experiments. We formalize the above as a computational problem called *the maximum agreement supertree problem* (MASP).

Further motivation for studying maximum agreement supertrees comes from the relation to a well-studied problem known as *the maximum agreement subtree problem* (MAST) in which the input is a set of leaf-labeled trees and the goal is to compute a tree contained in all of the input trees with as many labeled leaves as possible. Our results in this paper complement those previously known for MAST. The computational complexity of MAST has been closely investigated (see Section 1.2), motivated by the practical usefulness of maximum agreement subtrees. For example, maximum agreement subtrees can be used not only to identify small problematic subsets of taxa during phylogenetic reconstruction, but also to measure the similarity of a given set of trees [9,11,19] or to estimate a classification's stability to small changes in the data [11]. Moreover, MAST-based algorithms have been used to prepare and improve bilingual context-using dictionaries for automated language translation systems [8,21].

1.1 Problem Definitions

Let T be a tree whose leaves are labeled by a set S. We say that T is *distinctly leaf-labeled by* S if no two leaves in T have the same label. Below, each leaf in such a tree is identified with its corresponding label in S. Given a rooted, unordered, distinctly leaf-labeled tree T and a set S', *the topological restriction of* T *to* S' (denoted by $T \mid S'$) is the tree obtained by deleting from T all nodes which are not on any path from the root to a leaf in S' along with their incident edges, and then contracting every edge between a node having just one child and its child (see Fig. 1). For any tree T, denote its set of leaves by $\Lambda(T)$.

Let $\mathcal{T} = \{T_1, T_2, ..., T_k\}$ be a set of rooted, unordered trees, where each T_i is distinctly leaf-labeled and where the sets $\Lambda(T_i)$ may overlap. A *total agreement supertree of* $\mathcal{T}$ is a tree Q such that Q is distinctly leaf-labeled by $\bigcup_{T_i \in \mathcal{T}} \Lambda(T_i)$ and $Q \mid \Lambda(T_i)$ is isomorphic to T_i for every $T_i \in \mathcal{T}$. Note that two or more trees in $\mathcal{T}$ may contain conflicting branching information, in which case a total agreement supertree of $\mathcal{T}$ does not exist. *The total agreement supertree problem* (TASP) is: Given a set $\mathcal{T}$ of distinctly leaf-labeled, rooted, unordered trees, output a total agreement supertree of $\mathcal{T}$ if one exists, otherwise output null.

When $\mathcal{T} = \{T_1, T_2, ..., T_k\}$ is specified, we write $S = \bigcup_{T_i \in \mathcal{T}} \Lambda(T_i)$ and call S *the leaf set of* $\mathcal{T}$. For any $S' \subseteq S$, we let $\mathcal{T} \,|\, S'$ denote the set $\{T_1 \,|\, S', T_2 \,|\, S', ..., T_k \,|\, S'\}$. If there exists a total agreement supertree Q of $\mathcal{T} \,|\, S'$ then we say that S' is *consistent with* $\mathcal{T}$ and call Q an *agreement supertree of* $\mathcal{T}$. A *maximum agreement supertree of* $\mathcal{T}$ is an agreement supertree of $\mathcal{T}$ with as many leaves as possible. *The maximum agreement supertree problem* (MASP) is: Given a set $\mathcal{T}$ of distinctly leaf-labeled, rooted, unordered trees, output a maximum agreement supertree of $\mathcal{T}$. An *agreement subtree of* $\mathcal{T}$ is a tree U such that for some $S' \subseteq S$ it holds that U is distinctly leaf-labeled by S' and $T_i \,|\, S'$ is isomorphic to U for every $T_i \in \mathcal{T}$. A *maximum agreement subtree of* $\mathcal{T}$ is an agreement subtree of $\mathcal{T}$ with the maximum possible number of leaves. *The maximum agreement subtree problem* (MAST), also referred to in the literature as *the maximum homeomorphic subtree problem* (MHT), is to find a maximum agreement subtree of $\mathcal{T}$.

Throughout this paper, we let n denote the cardinality of the leaf set and k the number of input trees, i.e., $n = \left|\bigcup_{T_i \in \mathcal{T}} \Lambda(T_i)\right|$ and $k = |\mathcal{T}|$ in the problem definitions above. We let $D = \max_{T_i \in \mathcal{T}}\{\deg(T_i)\}$, where $\deg(T_i)$ is the degree[1] of T_i. We assume that none of the trees in $\mathcal{T}$ have a node with degree 1, so that each tree contains $O(n)$ nodes. Note that if we are given a subset S' of S which is consistent with $\mathcal{T}$, then we can efficiently construct a total agreement supertree of $\mathcal{T} \,|\, S'$ using the algorithm for TASP by Henzinger *et al.* [16] (see also Lemma 7 in Section 5). Hence, we focus on the subproblem of MASP of computing a maximum cardinality subset S' of S such that S' is consistent with $\mathcal{T}$.

A *rooted triplet* is a distinctly leaf-labeled, binary, rooted, unordered tree with three leaves. The unique rooted triplet on $\{a, b, c\}$ in which the lowest common ancestor of a and b is a proper descendant of the lowest common ancestor of a and c (or equivalently, where the lowest common ancestor of a and b is a proper descendant of the lowest common ancestor of b and c) is denoted by $(\{a, b\}, c)$.

1.2 Previous Results

Comprehensive surveys of existing methods for constructing supertrees can be found in [4,20,22]. Below, we mention some known results related to MASP.

Aho, Sagiv, Szymanski, and Ullman [1] presented an algorithm which can be used to solve TASP in $O(kn)$ time when all trees in $\mathcal{T}$ are rooted triplets. Several years later, Henzinger, King, and Warnow [16] showed how to modify the algorithm to solve TASP for any $\mathcal{T}$ in $\min\{O(Nn^{0.5}), O(N + n^2 \log n)\}$ time, where $N = \sum_{T_i \in \mathcal{T}} |T_i|$ is the total number of nodes in $\mathcal{T}$. In contrast, the analog of TASP for *unrooted* trees is NP-hard, even if all of the input trees are *quartets* (distinctly leaf-labeled, unrooted trees each having four leaves and no nodes with precisely two neighbors) [23]. A polynomial-time algorithm for

[1] The *degree of a node* u in a rooted tree is the number of children of u. The *degree of a rooted tree* T is the maximum degree of all nodes in T.

computing an unrooted total agreement supertree if one exists when all k input trees are binary and $k = O(1)$ was given by Bryant in [6].

The computational complexity of MAST has been studied extensively (e.g., [3,5,8,9,10,11,14,15,19,24]). Today, the fastest known algorithm for MAST for two trees, invented by Kao, Lam, Sung, and Ting [19], runs in $O\big(\sqrt{D}\, n \log(2n/D)\big)$ time, which is $O(n \log n)$ when $D = O(1)$ and $O(n^{1.5})$ when D is unrestricted.

Amir and Keselman [3] considered the case of $k \geq 3$ input trees. They proved that MAST is NP-hard for three trees with unrestricted degrees, but solvable in polynomial time for three or more trees if the degree of at least one of the trees is bounded by a constant. For the latter case, Farach, Przytycka, and Thorup [9] gave an algorithm with improved efficiency running in $O(kn^3 + n^d)$ time, where d is an upper bound on at least one of the input trees' degrees; Bryant [5] proposed a conceptually different algorithm with the same running time.

Hein, Jiang, Wang, and Zhang [15] proved the following inapproximability result: MAST for three trees with unrestricted degrees cannot be approximated within a factor of $2^{\log^\delta n}$ in polynomial time for any constant $\delta < 1$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}[2^{\mathrm{polylog}\, n}]$. Gąsieniec, Jansson, Lingas, and Östlin [14] proved that MAST cannot be approximated within a factor of n^ε for any constant ε where $0 \leq \varepsilon < \frac{1}{9}$ in polynomial time unless $\mathrm{P} = \mathrm{NP}$, even for instances containing only trees of height 2, and showed that if the number of trees is bounded by a constant and all the input trees' heights are bounded by a constant then MAST can be approximated within a constant factor in $O(n \log n)$ time.

A problem related to MASP and MAST is *the maximum refinement subtree problem* (MRST). Its goal is to construct a tree W with $\Lambda(W) \subseteq S$ which maximizes $|\Lambda(W)|$ such that for each $T_i \in \mathcal{T}$, $T_i \,|\, \Lambda(W)$ can be obtained from W by applying a series of edge contractions. MRST is NP-hard for $k = 2$ if D is unrestricted [15] but solvable in polynomial time if $k = O(1)$ and $D = O(1)$ [12]. Another related problem is *the maximum compatible subset of rooted triplets problem* (MCSR) in which the input is a set $\mathcal{T}$ of rooted triplets and the objective is to find a $\mathcal{T}' \subseteq \mathcal{T}$ of maximum cardinality such that there exists a total agreement supertree of $\mathcal{T}'$. MCSR is NP-hard [5,18]; two polynomial-time approximation algorithms for MCSR were given in [14].

1.3 Our Results and Organization of Paper

In Section 2, we make use of known positive and negative results for MAST to obtain an efficient algorithm for MASP restricted to $k = 2$ and an NP-hardness proof for MASP restricted to any fixed $k \geq 3$, respectively. The algorithm for $k = 2$ runs in $O\big(\sqrt{D}\, n \log(2n/D)\big)$ time, which is $O(n \log n)$ when $D = O(1)$ and $O(n^{1.5})$ when D is unrestricted. Then, in Section 3, we present a more complex MAST-based algorithm for solving MASP with $D = 2$. It runs in $O(k(2n)^{3k^2})$ time, which is polynomial when $k = O(1)$. In Section 4, we prove that MASP is NP-hard even if all of the input trees are required to be rooted triplets (i.e., $D = 2$ and k is unrestricted). Finally, in Section 5, we describe a simple polynomial-time approximation algorithm for MASP which is guaranteed to find an approximate solution with at least $\frac{\log n}{n}$ times the number of leaves in an optimal solution.

2 Preliminaries

We first investigate the close relationship between MASP and MAST.

Lemma 1. *For any set $\mathcal{T} = \{T_1, T_2, ..., T_k\}$ of distinctly leaf-labeled, rooted, unordered trees such that $\Lambda(T_1) = \Lambda(T_2) = ... = \Lambda(T_k)$, an optimal solution to MASP for $\mathcal{T}$ is an optimal solution to MAST for $\mathcal{T}$ and vice versa.*

Proof. Write $S = \Lambda(T_1) = \Lambda(T_2) = ... = \Lambda(T_k)$, let Q be any agreement supertree of $\mathcal{T}$, and let $S' = \Lambda(Q)$. Then, by definition, $Q \,|\, \Lambda(T_i \,|\, S') = T_i \,|\, S'$ for every $T_i \in \mathcal{T}$. Now, $\Lambda(T_i \,|\, S') = S \cap S' = S'$, so $T_i \,|\, S' = Q \,|\, S' = Q$ for every $T_i \in \mathcal{T}$, which means that Q is an agreement subtree of $\mathcal{T}$. Conversely, let U be an agreement subtree of $\mathcal{T}$ whose leaves are distinctly labeled by some set S'. For every $T_i \in \mathcal{T}$, we have $T_i \,|\, S' = U$. Then $U \,|\, \Lambda(T_i \,|\, S') = (T_i \,|\, S') \,|\, \Lambda(T_i \,|\, S') = T_i \,|\, S'$ for every $T_i \in \mathcal{T}$, i.e., U is an agreement supertree of $\mathcal{T}$. □

Theorem 1. *MASP with $k = 2$ can be solved in $O\big(\sqrt{D}\, n \log(2n/D)\big)$ time.*

Proof. Given an instance $\mathcal{T} = \{T_1, T_2\}$ of MASP with $k = 2$, let $L = \Lambda(T_1) \cap \Lambda(T_2)$ and run the algorithm of Kao, Lam, Sung, and Ting [19] on the instance $\mathcal{T} \,|\, L$ to obtain a maximum agreement subtree U of $\mathcal{T} \,|\, L$. This takes $O\big(\sqrt{D}\, n \log(2n/D)\big)$ time. By Lemma 1, U is also a maximum agreement supertree of $\mathcal{T} \,|\, L$. Next, for every leaf which appears in exactly one of T_1 and T_2, insert it into U according to its position in T_1 or T_2. More precisely, let $X = L \backslash \Lambda(U)$ and first compute $T_1' = T_1 \,|\, (\Lambda(T_1) \backslash X)$ and $T_2' = T_2 \,|\, (\Lambda(T_2) \backslash X)$ in $O(n)$ time. For any node $u \in U$, let $T_1'(u)$ and $T_2'(u)$ be the node in T_1' and T_2' respectively corresponding to u. Construct a tree Q as follows: initially, set $Q = T_1'$, then for each edge (u, v) of U, where we assume u is the parent of v, replace the edge in Q between $T_1'(v)$ and its parent with the path in T_2' between $T_2'(v)$ and $T_2'(u)$. Q can be constructed using a total of $O(n)$ time. It is straightforward to show that Q is a maximum agreement supertree of $\mathcal{T}$. □

The running time given in Theorem 1 is $O(n \log n)$ for two trees whose degrees are bounded by a constant and $O(n^{1.5})$ for two trees with unrestricted degrees.

The NP-hardness of MAST for any fixed $k \geq 3$ when D is unrestricted [3] together with Lemma 1 yield the following theorem (in fact, Lemma 1 can be used to show that the inapproximability results of [14] and [15] for MAST mentioned in Section 1.2 hold for MASP as well).

Theorem 2. *For any fixed $k \geq 3$, MASP with unrestricted D is NP-hard.*

3 A Polynomial-Time Algorithm for $D = 2$, $k = O(1)$

In this section, we show how MASP restricted to $D = 2$ can be reduced to MAST for a set of k distinctly leaf-labeled binary trees having $O((2n)^{k^2})$ leaves.[2] Hence, we can solve MASP with $D = 2$ in polynomial time if $k = O(1)$.

[2] The proofs and figures in this section have been omitted due to space constraints. They can be found in the full-length version of our paper.

Without loss of generality, assume that every $a \in S$ appears in at least two trees in $\mathcal{T}$. (If a appears in exactly one tree in $\mathcal{T}$, we can obtain a maximum agreement supertree of $\mathcal{T}$ as follows: (1) Remove a from $\mathcal{T}$; (2) compute a maximum agreement supertree T' for the modified $\mathcal{T}$; and (3) insert a into T' according to its position in the original $\mathcal{T}$, as described in the proof of Theorem 1 above.)

MASP is first transformed to MAST for non-distinctly leaf-labeled trees; then, the latter problem is transformed to MAST. Here, by an agreement subtree of a set $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ of *non-distinctly* leaf-labeled trees, we mean a distinctly leaf-labeled tree which is a homeomorphic subtree of every $R_i \in \mathcal{R}$.

We now describe our transformation from MASP to MAST for a set $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$ of non-distinctly leaf-labeled binary trees. To obtain each R_i:

1. Set $R_{i,0} = T_i$.
2. For $j = 1$ to k, do
   a) Let $L = \Lambda(T_j) \setminus \bigcup_{j' \in \{1,\ldots,j-1\} \cup \{i\}} \Lambda(T_{j'})$ and let $U = T_j | L$.
   b) Initially, set $R_{i,j} = R_{i,j-1}$. Generate $|R_{i,j-1}| - 1$ copies of U and attach one to every edge of $R_{i,j}$. Let r be a new node having the current $R_{i,j}$ and another copy of U as its two subtrees, and make r the root of $R_{i,j}$.
3. Set $R_i = R_{i,k}$.

Based on the above construction, for every i, any label in $\Lambda(T_i)$ appears exactly once in R_i, and T_i is a homeomorphic subtree of R_i. Also, $\mathcal{R}$ satisfies:

Lemma 2. *For every $R_i \in \mathcal{R}$, the number of leaves in R_i is at most $(2n)^k$ and the height of R_i is at most $2^k n$.*

Lemma 3. *For any tree X which is distinctly leaf-labeled by some $S' \subseteq S$, X is an agreement supertree of $\mathcal{T}$ if and only if X is an agreement subtree of $\mathcal{R}$.*

Next, we transform MAST for the set $\mathcal{R}$ of non-distinctly leaf-labeled binary trees to MAST for a set $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ of binary trees which are distinctly leaf-labeled by $\{a^1_{b_1,b_2,\ldots,b_k}, a^2_{b_1,b_2,\ldots,b_k} \mid a \in S,\ 1 \le i \le k,\ 1 \le b_i \le \gamma_a[i]\}$, where $\gamma_a[i]$ is the number of occurrences of leaf label a in R_i.

To describe the transformation, we need some additional notation. For every $a \in S$, define $a([b_1..d_1], [b_2..d_2], \ldots, [b_k..d_k])$, where $b_i \le d_i$ for all $1 \le i \le k$, to be a rooted caterpillar with $\prod_{i=1}^{k}(2(d_i - b_i + 1))$ leaves labeled (in order of non-decreasing distance from the root) by $a^1_{b_1,b_2,\ldots,b_k}$, $a^2_{b_1,b_2,\ldots,b_k}$, $a^1_{b_1,b_2,\ldots,b_k+1}$, $a^2_{b_1,b_2,\ldots,b_k+1}$, $\ldots$, $a^1_{d_1,d_2,\ldots,d_k}$, $a^2_{d_1,d_2,\ldots,d_k}$. Define $\bar{a}([b_1..d_1], [b_2..d_2], \ldots, [b_k..d_k])$ as the reversed caterpillar of $a([b_1..d_1], [b_2..d_2], \ldots, [b_k..d_k])$. For every leaf in R_i labeled by a, such a leaf is called *the jth occurrence of a in R_i* if, according to pre-order traversal of R_i, it is the jth visited leaf which is labeled by a.

For $i = 1, 2, \ldots, k$, the tree P_i is constructed from R_i by replacing, for every $a \in S$, the leaf labeled by a with a caterpillar tree $a()$ or $\bar{a}()$ as follows.

1. Set $P_i = R_i$.
2. For every $a \in S$,
   – if T_i is the first tree containing a among $T_1, T_2, \ldots, T_i$, then (in this case, P_i contains exactly one a, that is, $\gamma_a[i] = 1$) replace a in P_i by the caterpillar $\bar{a}([1..\gamma_a[1]], \ldots, [1..\gamma_a[i-1]], [1..1], [1..\gamma_a[i+1]], \ldots, [1..\gamma_a[k]])$.

- else for $j = 1, 2, \ldots, \gamma_a[i]$, replace the jth occurrence of a in P_i by the caterpillar $a([1..\gamma_a[1]], \ldots, [1..\gamma_a[i-1]], [j..j], [1..\gamma_a[i+1]], \ldots, [1..\gamma_a[k]])$.

It is easy to check that each P_i is distinctly labeled by $\{a^1_{b_1,b_2,\ldots,b_k}, a^2_{b_1,b_2,\ldots,b_k} \mid a \in S,\ 1 \le i \le k,\ 1 \le b_i \le \gamma_a[i]\}$. In addition, for every label $a \in S$, there exists exactly one tree P_i which contains the caterpillar $\bar{a}()$ while the rest of the trees in $\mathcal{P}$ contain caterpillars of the form $a()$. Below, more properties of $\mathcal{P}$ are described.

Lemma 4. *For every P_i, $|\Lambda(P_i)| = O((2n)^{k^2})$.*

Lemma 5. *For any $a \in S$, a MAST of $\mathcal{P}$ has ≤ 2 leaves of the form $a^\ell_{b_1,b_2,\ldots,b_k}$.*

Lemma 6. *For any integer x, the size of the MAST of $\mathcal{R}$ is $\ge x$ if and only if the size of the MAST of $\mathcal{P}$ is $\ge 2x$.*

A MASP of $\mathcal{T}$ can now be computed by applying the algorithm of Bryant [5] or Farach *et al.* [9] (see Section 1.2) to $\mathcal{P}$. Since the number of leaves in $\mathcal{P}$ is less than $(2n)^{k^2}$ and all trees are binary, we obtain the main theorem of this section.

Theorem 3. *Given a set of k binary trees $\mathcal{T}$ which are labeled by n distinct labels, their maximum agreement supertree can be computed in $O(k(2n)^{3k^2})$ time.*

4 MASP with $D = 2$ Is NP-Hard

Theorem 2 states that MASP is an NP-hard problem for any fixed $k \ge 3$ when D is unrestricted. We now show that MASP remains NP-hard if restricted to instances with $D = 2$ but where k is left unrestricted. In fact, we prove that MASP is NP-hard even if all of the input trees are required to be rooted triplets. Our NP-hardness proof consists of a polynomial-time reduction from the independent set problem which is known to be NP-hard (see, e.g., [13]).

The independent set problem
Instance: An undirected graph $G = (V, E)$ and a positive integer I.
Question: Is there a subset V' of V with $|V'| = I$ such that V' is an independent set, i.e., such that no two vertices in V' are joined by an edge in E?

The maximum agreement supertree problem restricted to rooted triplets, decision problem version (MASPR-d)
Instance: A set $\mathcal{T}$ of rooted triplets with leaf set S and a positive integer K.
Question: Is there a subset S' of S with $|S'| = K$ which is consistent with $\mathcal{T}$?

Theorem 4. *MASP is NP-hard even if restricted to rooted triplets.*

Proof. Given an arbitrary instance (G, I) of the independent set problem, construct an instance of MASPR-d as follows. Let $S = V \cup \{z_e \mid e \in E\}$ and set $K = I + |E|$. For each edge e in E, include the two rooted triplets $(\{a, z_e\}, b)$ and $(\{b, z_e\}, a)$ in $\mathcal{T}$, where $e = \{a, b\}$. Claim: G has an independent set of size I if and only if there exists a subset S' of S of size K which is consistent with $\mathcal{T}$.

Proof of claim: Suppose there exists an independent set W in G of size I. Then $S' = W \cup \{z_e \mid e \in E\}$ with $|S'| = I + |E|$ is consistent with $\mathcal{T}$ since $\mathcal{T} \mid S'$ contains no rooted triplets (if $\mathcal{T} \mid S'$ had a rooted triplet $(\{x, z_{\{x,y\}}\}, y)$ then x and y would be joined by an edge in E and thus could not both belong to W).

Conversely, suppose there exists a consistent subset S' of S of size K. For each $\{x, y\} \in E$, if $z_{\{x,y\}} \notin S'$ but at least one of x and y belongs to S' then replace x or y in S' by $z_{\{x,y\}}$, and if none of x, y, and $z_{\{x,y\}}$ are contained in S' then replace any element in S' belonging to V by $z_{\{x,y\}}$ (such an element always exists because $K > |E|$). The resulting set S'' will have the form $W \cup \{z_e \mid e \in E\}$ with $W \subseteq V$ and $|S''| = K$, and will still be consistent with $\mathcal{T}$. Next, observe that by the construction of $\mathcal{T}$, for each $\{x, y\} \in E$ at most two of x, y, and $z_{\{x,y\}}$ can be included in any subset of S which is consistent with $\mathcal{T}$. Therefore, for each $\{x, y\} \in E$, since $z_{\{x,y\}} \in S''$ it holds that S'' cannot contain both x and y. Thus, W is an independent set and $|W| = K - |E| = I$.

Hence, MASPR-d is NP-hard and the theorem follows. □

5 A Polynomial-Time $(n/\log n)$-Approximation Algorithm

By the comments preceding Theorem 2, it is highly unlikely that MASP in its general form can be solved exactly or even approximated efficiently (say, within a constant factor) in polynomial time. However, we can adapt one of Akutsu and Halldórsson's [2] algorithms for the largest common subtree problem to obtain the following polynomial-time $(n/\log n)$-approximation algorithm for MASP:

> Arbitrarily partition S into $\lfloor n/\log n \rfloor$ sets $S_1, S_2, ..., S_{\lfloor n/\log n \rfloor}$, each of size at most $\lceil \log n \rceil + 1$. Then, check every subset S_i' of every set S_i to see if S_i' is consistent with $\mathcal{T}$, and let Z be one such subset of maximum cardinality. Return Z.

To see that this algorithm always returns a solution with at least $\frac{\log n}{n}$ times the number of leaves in an optimal solution, let S^* be a maximum consistent leaf subset. Because of the pigeonhole principle, at least one of $S_1, S_2, ..., S_{\lfloor n/\log n \rfloor}$ contains $\geq \frac{1}{\lfloor n/\log n \rfloor}$ of the elements in S^*; thus, $|Z| \geq \frac{|S^*|}{\lfloor n/\log n \rfloor} \geq \frac{|S^*|}{n/\log n}$.

To implement the algorithm efficiently, we first note that the deterministic algorithm for dynamic graph connectivity employed in the algorithm for TASP of Henzinger *et al.* [16] can be replaced with a more recent one due to Holm *et al.* [17] to yield the following improvement. We then obtain Theorem 5 below.

Lemma 7. *TASP is solvable in* $\min\{O(N \log^2 n),\ O(N + n^2 \log n)\}$ *time, where* $N = \sum_{T_i \in \mathcal{T}} |T_i|$ *is the total number of nodes in* $\mathcal{T}$.

Theorem 5. *MASP can be approximated within a factor of* $\frac{n}{\log n}$ *in* $O(n^2 \cdot \log\log n) \cdot \min\{O(k \log\log n),\ O(k + \log n)\}$ *time. MASP restricted to rooted triplets can be approximated within a factor of* $\frac{n}{\log n}$ *in* $O(k + n^2 \log^2 n)$ *time.*

Finally, we remark that MAST can be approximated within a factor of $\frac{n}{\log n}$ in $O(kn^2)$ time using the same technique.

6 Concluding Remarks

Below, we summarize our results on how restricting the parameters D and k affects the computational complexity of MASP. Arrows indicate when a result follows directly from another by generalization (for example, MASP with $D = 2$ and unrestricted k is NP-hard, so the more general case $D = O(1)$ and unrestricted k cannot be any easier) or by specialization (e.g., the algorithm for $D = O(1)$ and $k = 2$ still works for the more restricted case $D = 2$ and $k = 2$).

MASP	$k = 2$	$k = O(1)$	k unrestricted
$D = 2$	$O(n \log n)$ ($\downarrow$)	$O(k(2n)^{3k^2})$ (Theorem 3)	NP-hard (Theorem 4)
$D = O(1)$	$O(n \log n)$ (Theorem 1)	Open	NP-hard ($\uparrow$)
D unrestricted	$O(n^{1.5})$ (Theorem 1)	NP-hard (Theorem 2)	NP-hard ($\leftarrow$ or $\uparrow$)

We have also described a polynomial-time $(n/\log n)$-approximation algorithm for MASP (Theorem 5).

It is interesting to note that MASP with $D = 2$ and unrestricted k is NP-hard while on the other hand, MAST with $D = 2$ and unrestricted k can be solved in $O(kn^3)$ time, i.e., in polynomial time, using the algorithm of Bryant [5] or Farach *et al.* [9] (see Section 1.2). This means that for certain restrictions on the parameters D and k, MASP and MAST cannot have the same computational complexity unless P = NP. Furthermore, although our results indicate that MASP is computationally harder than MAST, the maximum refinement subtree problem (see Section 1.2) does not seem any easier than MASP since it is NP-hard already for $k = 2$ when D is unrestricted [15].

An open problem is to determine the computational complexity of MASP with $D = O(1)$ and $k = O(1)$. We believe that this case is solvable in polynomial time. We would also like to know if the running time of our algorithm for the case $D = 2$ and $k = O(1)$ can be improved.

References

1. 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 Journal on Computing*, 10(3):405–421, 1981.
2. T. Akutsu and M. M. Halldórsson. On the approximation of largest common subtrees and largest common point sets. *Theoretical Computer Science*, 233(1–2):33–50, 2000.
3. A. Amir and D. Keselman. Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms. *SIAM Journal on Computing*, 26(6):1656–1669, 1997.
4. O. Bininda-Emonds, J. Gittleman, and M. Steel. The (super)tree of life: Procedures, problems, and prospects. *Annual Review of Ecology and Systematics*, 33:265–289, 2002.

5. D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees: Theory and Methods in Phylogenetic Analysis.* PhD thesis, Univ. of Canterbury, N.Z., 1997.
6. D. Bryant. Optimal agreement supertrees. In *Proc. of the 1st International Conference on Biology, Informatics, and Mathematics* (JOBIM 2000), volume 2066 of *LNCS*, pages 24–31. Springer, 2001.
7. B. Chor, M. Hendy, and D. Penny. Analytic solutions for three-taxon ML_{MC} trees with variable rates across sites. In *Proc. of the 1st Workshop on Algorithms in Bioinformatics* (WABI 2001), volume 2149 of *LNCS*, pages 204–213. Springer, 2001.
8. R. Cole, M. Farach-Colton, R. Hariharan, T. Przytycka, and M. Thorup. An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM Journal on Computing*, 30(5):1385–1404, 2000.
9. M. Farach, T. Przytycka, and M. Thorup. On the agreement of many trees. *Information Processing Letters*, 55:297–301, 1995.
10. M. Farach and M. Thorup. Sparse dynamic programming for evolutionary-tree comparison. *SIAM Journal on Computing*, 26(1):210–230, 1997.
11. C. R. Finden and A. D. Gordon. Obtaining common pruned trees. *Journal of Classification*, 2:255–276, 1985.
12. G. Ganapathysaravanabavan and T. Warnow. Finding a maximum compatible tree for a bounded number of trees with bounded degree is solvable in polynomial time. In *Proc. of the 1st Workshop on Algorithms in Bioinformatics* (WABI 2001), volume 2149 of *LNCS*, pages 156–163. Springer, 2001.
13. M. Garey and D. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.
14. L. Gąsieniec, J. Jansson, A. Lingas, and A. Östlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, 3:183–197, 1999.
15. J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, 71:153–169, 1996.
16. M. R. Henzinger, V. King, and T. Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24(1):1–13, 1999.
17. J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48(4):723–760, 2001.
18. J. Jansson. On the complexity of inferring rooted evolutionary trees. In *Proc. of the Brazilian Symp. on Graphs, Algorithms, and Combinatorics* (GRACO'01), volume 7 of *Electronic Notes in Discrete Mathematics*, pages 121–125. Elsevier, 2001.
19. M.-Y. Kao, T.-W. Lam, W.-K. Sung, and H.-F. Ting. An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings. *Journal of Algorithms*, 40(2):212–233, 2001.
20. P. Kearney. Phylogenetics and the quartet method. In T. Jiang, Y. Xu, and M. Q. Zhang, editors, *Current Topics in Computational Molecular Biology*, pages 111–133. The MIT Press, Massachusetts, 2002.
21. A. Meyers, R. Yangarber, and R. Grishman. Alignment of shared forests for bilingual corpora. In *Proc. of the 16th International Conference on Computational Linguistics* (COLING-96), pages 460–465, 1996.
22. M. J. Sanderson, A. Purvis, and C. Henze. Phylogenetic supertrees: assembling the trees of life. *TRENDS in Ecology & Evolution*, 13(3):105–109, 1998.
23. M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9(1):91–116, 1992.
24. M. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, 48:77–82, 1993.