Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 335 (2005) 93–107
Theoretical Computer Science
www.elsevier.com/locate/tcs

Computing the maximum agreement of phylogenetic networks☆

Charles Choy[a], Jesper Jansson[a,*], Kunihiko Sadakane[b], Wing-Kin Sung[a]

[a]*School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore*
[b]*Department of Computer Science and Communication Engineering, Kyushu University, Japan*

Abstract

We introduce the maximum agreement phylogenetic subnetwork problem (MASN) for finding branching structure shared by a set of phylogenetic networks. We prove that the problem is NP-hard even if restricted to three phylogenetic networks and give an $O(n^2)$-time algorithm for the special case of two level-1 phylogenetic networks, where n is the number of leaves in the input networks and where N is called a level-f phylogenetic network if every biconnected component in the underlying undirected graph induces a subgraph of N containing at most f nodes with indegree 2. We also show how to extend our technique to yield a polynomial-time algorithm for any two level-f phylogenetic networks N_1, N_2 satisfying $f = O(\log n)$; more precisely, its running time is $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$, where $V(N_i)$ and f_i denote the set of nodes in N_i and the level of N_i, respectively, for $i \in \{1, 2\}$.

Keywords: Phylogenetic network comparison; Maximum agreement subnetwork; Algorithm; Computational complexity

☆ A preliminary version of this article appeared in *Proceedings of Computing: the* 10th *Australasian Theory Symposium* (CATS 2004), Electronic Notes in Theoretical Computer Science, Vol. 91, pp. 134–147, Elsevier B.V., 2004.

* Corresponding author.
E-mail addresses: choychih@comp.nus.edu.sg (C. Choy), jansson@comp.nus.edu.sg (J. Jansson), sada@csce.kyushu-u.ac.jp (K. Sadakane), ksung@comp.nus.edu.sg (W.-K. Sung).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.12.012

1. Introduction

Phylogenetic trees have been used in many fields of science to describe how a set of objects (e.g., biological species, proteins, nucleic acids, languages, chain letters, or medieval manuscripts) produced by an evolutionary process are believed to be related [3,4,6,22,28]. In a phylogenetic tree, the objects are represented by leaves and common ancestors by internal nodes so that the branching structure of the tree reflects the assumed evolutionary relationships. However, certain evolutionary events such as horizontal gene transfer or hybrid speciation cannot be adequately represented in a single tree structure [15,16,23,25–27,30]. Phylogenetic networks were introduced in order to solve this shortcoming by allowing internal nodes to have more than one parent.

Recently, various algorithms for constructing and comparing phylogenetic networks have been proposed (see, e.g., [15,18,19,23,25–27,30]). Here, we consider the following scenario. Suppose a number of phylogenetic networks, each one describing the possible evolution of a fixed set of objects, have been obtained by applying different construction methods or different clustering criteria to some available data. Furthermore, suppose that these networks do not completely agree because of distortions due to assumptions inherent to the methods used or because of measurement errors. It would then be informative to find a subnetwork contained in every one of the input networks with as many labeled leaves as possible since such a subnetwork more likely represents genuine evolutionary structure in the data. In this way, one would get an indication of which ancestral relationships can be regarded as resolved and which objects need to be subjected to further experiments.

We formalize the above as a computational problem called *the maximum agreement phylogenetic subnetwork problem* (MASN). Since the number of leaves in the input phylogenetic networks may be very large, we investigate the computational complexity of MASN and some of its restrictions to determine when the problem can be solved by efficient algorithms.

Further motivation for MASN comes from its relation to a well-studied problem known as *the maximum agreement subtree problem* (MAST). [1] Phylogenetic networks are a natural generalization of rooted binary phylogenetic trees; similarly, MASN generalizes MAST restricted to rooted binary trees. Hence, our results in this paper complement those previously established for MAST. The computational complexity of MAST has been closely studied (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 species during phylogenetic reconstruction, but also to measure the similarity of a given set of trees [9,12,21] or to estimate a classification's stability to small changes in the data [12]. Moreover, MAST-based algorithms have been used to prepare and improve bilingual context-using dictionaries for automated language translation systems [7,24].

[1] In MAST, 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; see, e.g., [2] or [29] for a formal definition. MAST is also referred to as *the maximum homeomorphic subtree problem* (MHT) by some researchers.

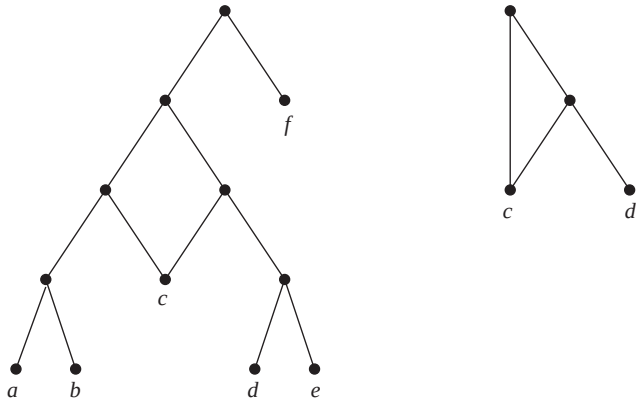

Fig. 1. Let N be the phylogenetic network on the left. Then $N \mid \{c, d\}$ is the phylogenetic network on the right.

1.1. Problem definition

Let L be a finite set. A *phylogenetic network*[2] *for* L is a connected, rooted, simple, directed acyclic graph in which: (1) each node has outdegree at most 2; (2) each node has indegree 1 or 2, except the root node which has indegree 0; (3) no node has both indegree 1 and outdegree 1; and (4) all nodes with outdegree 0 are labeled by elements from L in such a way that no two nodes are assigned the same label. From here on, nodes of outdegree 0 are referred to as *leaves* and identified with their corresponding elements in L.

Given a phylogenetic network N for L and a subset L' of L, *the topological restriction* of N to L', denoted by $N \mid L'$, is defined as the phylogenetic network obtained by first deleting all nodes which are not on any directed path from the root to one of the leaves in L' along with their incident edges, and then, for every node with outdegree 1 and indegree less than 2, contracting its outgoing edge (any resulting set of multiple edges between two nodes is replaced by a single edge). See Fig. 1 for an example.

Given a set $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$ of phylogenetic networks for L, an *agreement subnetwork of* $\mathcal{N}$ is a phylogenetic network A such that for some $L' \subseteq L$ it holds that A is a subgraph of each of $N_1 \mid L'$, $N_2 \mid L'$, $\ldots$, and $N_k \mid L'$. A *maximum agreement subnetwork of* $\mathcal{N}$ is an agreement subnetwork of $\mathcal{N}$ with the maximum possible number of leaves. *The* MASN is: Given a finite set L and a set $\mathcal{N}$ of phylogenetic networks for L, find a maximum agreement subnetwork of $\mathcal{N}$. See Fig. 2. Throughout this paper, n and k represent the cardinalities of L and $\mathcal{N}$, respectively, in the problem definition above.

[2] The existing methods for phylogenetic network reconstruction (see Section 1.2) make various assumptions on the available data and on the structure of the phylogenetic network that is to be constructed. In addition, the exact definition of a phylogenetic network varies somewhat from paper to paper. Therefore, to be able to compare phylogenetic networks produced by different construction methods, the definition that we use here is more general, in the sense that it focuses on the topology of the given networks and does not, for example, require internal nodes to be labeled as in [15,30] or that certain temporal constraints are satisfied as in [23,26] (also note that when some species are missing from a data set, e.g., due to extinction, some construction methods might not produce a phylogenetic network for the observed species which satisfies the temporal constraints stated in [23,26]).

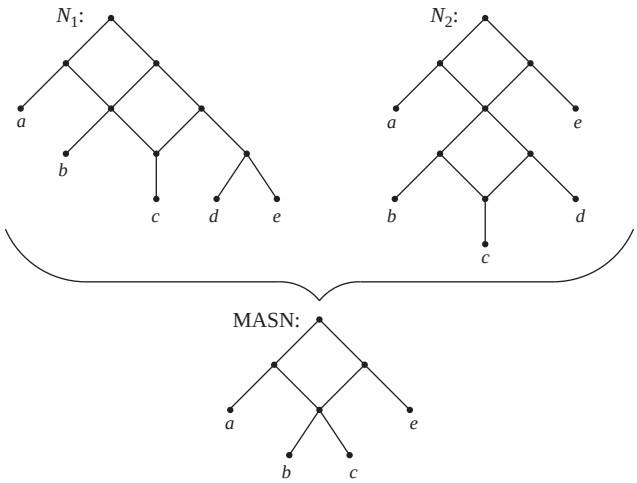

Fig. 2. One of the maximum agreement subnetworks of two given phylogenetic networks N_1 and N_2. This solution is not unique; another maximum agreement subnetwork of N_1 and N_2 has leaf set $\{a, b, c, d\}$.

1.2. Previous results

Several methods for *constructing* phylogenetic networks have been proposed previously; for a survey, refer to [23,27]. See also [15,18,19,26,30] for some recent related results not described in the surveys. In particular, [15,19,26,30] consider problems involving constructing a phylogenetic network with an additional structural constraint which we in this paper refer to as a *level-1* phylogenetic network, to be defined in Section 2. A method for *comparing* two given phylogenetic networks (more precisely, measuring their similarity in order to assess the topological accuracy of different phylogenetic network construction methods) based on the Robinson–Foulds (RF) measure for phylogenetic trees was proposed by Nakhleh et al. [25].

No results for MASN in its general form have appeared in the literature before. On the other hand, the special case of MASN known as the MAST has received a lot of attention in the last ten years. Below, we summarize some of the most important results known for MAST.

Finden and Gordon [12] presented a polynomial-time heuristic (not guaranteed to find an optimal solution) for MAST restricted to instances consisting of two binary trees. A few years later, Steel and Warnow [29] gave the first exact polynomial-time algorithm to solve MAST for two trees with unbounded degrees. Since then, a great number of improvements have been published (e.g., [7,10,11,20,21]). The fastest currently known algorithm for MAST for two trees, invented by Kao et al. [21], runs in $O(\sqrt{D}\, n \log(2n/D))$ time, where n is the number of leaves and D is the maximum degree of the two input trees. Note that this is $O(n \log n)$ for two trees with maximum degree bounded by a constant and $O(n^{1.5})$ for two trees with unbounded degrees.

Amir and Keselman [2] considered the case of $k \geqslant 3$ input trees. They proved that MAST is NP-hard for three trees with unbounded degrees, but solvable in polynomial time for

three or more trees if the degree of at least one of the input trees is bounded by a constant. For the latter case, Farach et al. [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. Bryant's approach led to a recent result in the field of parameterized complexity theory stating that it is possible to determine whether an instance of MAST has an agreement subtree with at least $n - \mu$ leaves for any integer $0 \leqslant \mu \leqslant n$ in $O(kn^3 + 2.270^\mu)$ time [3] (see [1]).

Hein et al. [17] proved the following inapproximability result: MAST with three trees with unbounded degrees cannot be approximated within a factor of $2^{\log^\delta n}$ in polynomial time for any constant $\delta < 1$, unless NP $\subseteq$ DTIME$[2^{\text{polylog}\, n}]$. Gąsieniec et al. [14] proved that MAST is hard to approximate in polynomial time 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.

1.3. Our results and organization of paper

We define the concept of a level-f phylogenetic network in Section 2. Then, in Section 3, we present an algorithm for computing a maximum agreement subnetwork between two level-1 phylogenetic networks in $O(n^2)$ time. We also describe how our algorithm can be extended to solve MASN for a level-f_1 phylogenetic network N_1 and a level-f_2 phylogenetic network N_2 in $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$ time (where $V(N_i)$ denotes the set of nodes of N_i), which is polynomial in the input size when $\max\{f_1, f_2\} = O(\log n)$. Next, in Section 4, we prove that in the general case (i.e., for level-f phylogenetic networks where f is unbounded), the MASN is NP-hard even if restricted to just three networks. Finally, we state some open problems in Section 5.

2. Preliminaries

Let N be a phylogenetic network for a finite set L. Recall that nodes with outdegree 0 are called *leaves*. We refer to nodes with indegree 2 as *hybrid nodes*. (Observe that a leaf can also be a hybrid node.) For any hybrid node h in N, its two incoming edges are called *the hybrid edges of* h and its two parents *the hybrid parents of* h. To distinguish between the hybrid edges of h, we call one of them *the left hybrid edge of* h ($lhe(h)$) and the other one *the right hybrid edge of* h ($rhe(h)$). *The left hybrid parent of* h ($lhp(h)$) and *the right hybrid parent of* h ($rhp(h)$) are defined accordingly. Every ancestor of h from which $lhp(h)$ and $rhp(h)$ can be reached using two disjoint paths is called a *split node of* h. If s is a split node of h then any path starting at a child of s and ending at a parent of h is called a *clipped merge path of* h. If h only has one split node, we denote it by $sn(h)$. See Fig. 3.

Let $\mathcal{U}(N)$ be the undirected graph obtained from N by replacing each directed edge by an undirected edge. For every biconnected component B in $\mathcal{U}(N)$, the *level of* B is defined

[3] Note that $O(kn^3 + 2.270^\mu)$ running time might be preferable to $O(kn^3 + n^d)$ if d is unbounded and the number of leaves we are willing to exclude is small.

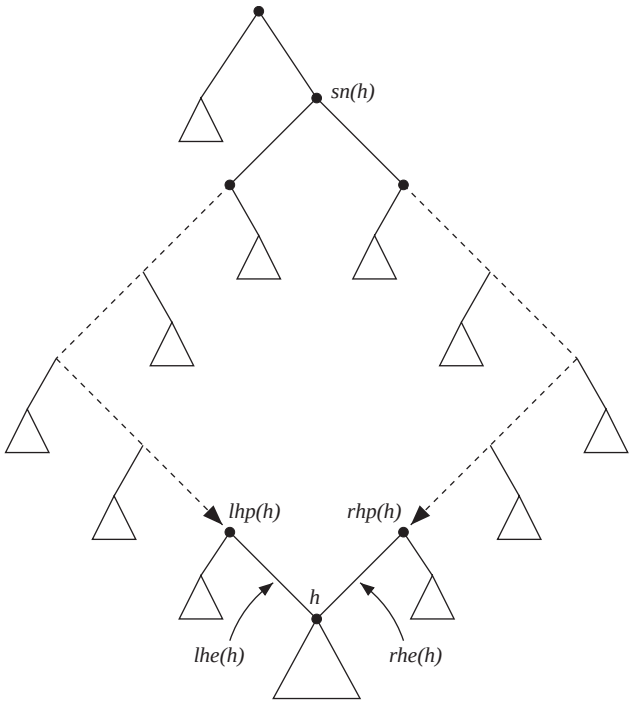

Fig. 3. Here, h is a hybrid node with a unique split node. The figure shows $sn(h)$, $lhp(h)$, $rhp(h)$, $lhe(h)$, $rhe(h)$, and as dashed lines, two clipped merge paths of h.

as the number of nodes with indegree 2 in the subgraph of N induced by the set of nodes in B. N is said to be a *level-f* phylogenetic network if the maximum level of all biconnected components in $\mathcal{U}(N)$ is equal to f. (For example, N_1 and N_2 in Fig. 2 are a level-2 and a level-1 phylogenetic network, respectively.) Note that N is a tree if and only if $f = 0$. If $f = 1$ then every node in N belongs to at most one clipped merge path.[4] Moreover, if $f = 1$ then every hybrid node in N has only one split node and any node in N can be a split node for at most one hybrid node.

3. An algorithm for MASN for two phylogenetic networks

Given two level-0 phylogenetic networks (i.e., trees), MASN can be solved in $O(n \log n)$ time by using the algorithm in [7] or [21]. In this section, we consider how to compute a maximum agreement subnetwork of two level-f phylogenetic networks for $f > 0$.

[4] The biological relevance of level-1 phylogenetic networks (there referred to as *galled-trees*) is discussed in [15].

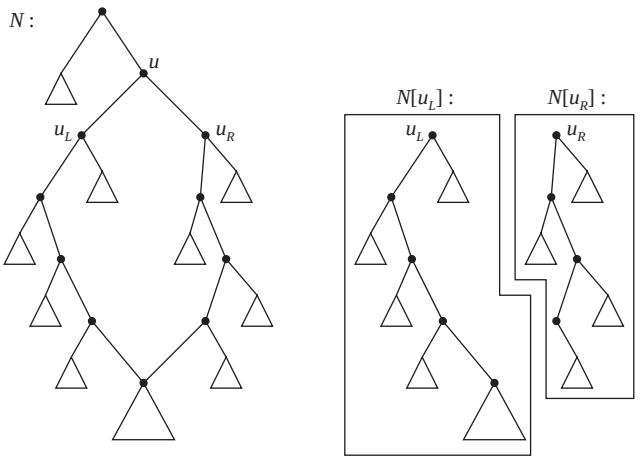

Fig. 4. N is a level-1 phylogenetic network and u is a split node in N. $N[u_{\mathrm{L}}]$ and $N[u_{\mathrm{R}}]'$ are the subgraphs of N shown on the right.

We present an $O(n^2)$-time algorithm for the case $f = 1$ and then show how it can be extended to solve MASN in polynomial time for any f which is upper-bounded by $O(\log n)$.

We first introduce some notation. Let N be a level-1 phylogenetic network. $V(N)$ stands for the set of nodes of N and $\Lambda(N)$ for the set of leaf labels in N. From this point onward, we assume that some arbitrary left-to-right ordering of the children of every node has been fixed. If $u \in V(N)$ has two children then u_{L} and u_{R} denote the left and right child of u, respectively, and if u only has one child c then we set $u_{\mathrm{L}} = c$ and $u_{\mathrm{R}} = \emptyset$. For any $u \in V(N)$, $N[u]$ is the subnetwork of N rooted at u, i.e., the minimal subgraph of N which includes all nodes and directed edges of N reachable from u. $N[\emptyset]$ refers to the empty network with no nodes or edges.

Next, for every $u \in V(N)$, we define a subgraph $N[u]'$ of $N[u]$ as follows. If u belongs to a clipped merge path of some hybrid node h then $N[u]'$ is the subgraph of $N[u]$ where $N[h]$ and h's incoming edge have been removed (since N is a level-1 phylogenetic network, each $u \in V(N)$ can belong to at most one clipped merge path). Otherwise, $N[u]'$ is defined as $N[u]$ if u is not a hybrid node in N, and as $N[\emptyset]$ if u is a hybrid node in N. See Fig. 4 for an example.

For any two level-1 phylogenetic networks N_1 and N_2, define $\mathit{Masn}(N_1, N_2)$ as the number of leaves in a maximum agreement subnetwork of N_1 and N_2. If N_1 or N_2 is an empty network then $\mathit{Masn}(N_1, N_2)$ is equal to 0. Otherwise, $\mathit{Masn}(N_1, N_2)$ can be expressed recursively using the following lemma which is a straightforward generalization of the main lemma in [29] for MAST (the only difference is the case $\mathit{Match}(N_1[u], N_2[v])$; here, when trying to match the two subnetworks rooted at u_{L} and u_{R} to the two subnetworks rooted at v_{L} and v_{R}, we ensure that the set of nodes in the intersection of $V(N_1[u_{\mathrm{L}}])$ and $V(N_1[u_{\mathrm{R}}])$ is matched to only one of the two subnetworks rooted at v_{L} and v_{R}, and conversely).

Algorithm *ComputeMasn*
Input: Two level-1 phylogenetic networks N_1 and N_2.
Output: The number of leaves in a maximum agreement subnetwork of $\{N_1, N_2\}$.
1 Let $\mathcal{O}$ be the lexicographic ordering of $V(N_1) \times V(N_2)$, where the nodes in each $V(N_i)$ are ordered according to postorder.
2 **for** each $(u, v) \in V(N_1) \times V(N_2)$ in increasing order in $\mathcal{O}$ **do**
Compute $Masn(N_1[u], N_2[v])$, $Masn(N_1[u]', N_2[v])$,
$Masn(N_1[u], N_2[v]')$, and $Masn(N_1[u]', N_2[v]')$ by using the expression in Lemma 1.
endfor
3 **return** $Masn(N_1[r_1], N_2[r_2])$, where r_i is the root of N_i for $i \in \{1, 2\}$.
End *ComputeMasn*

Fig. 5. A dynamic programming algorithm for computing all values of *Masn*.

Lemma 1. *Let N_1 and N_2 be two level-1 phylogenetic networks. For every $(u, v) \in V(N_1) \times V(N_2)$,*

$$Masn(N_1[u], N_2[v]) = \begin{cases} |\Lambda(N_1[u]) \cap \Lambda(N_2[v])|, & \text{if at least one of } u \text{ and } v \\ & \text{is a leaf}, \\ \max\{Diag(N_1[u], N_2[v]),\ Match(N_1[u], N_2[v])\}, & \\ & \text{otherwise}, \end{cases}$$

where

$$\begin{aligned} Diag(N_1[u], N_2[v]) = \max\{&Masn(N_1[u], N_2[v_L]),\ Masn(N_1[u], N_2[v_R]), \\ &Masn(N_1[u_L], N_2[v]),\ Masn(N_1[u_R], N_2[v])\} \end{aligned}$$

and

$$\begin{aligned} &Match(N_1[u], N_2[v]) \\ &\quad = \max\{Masn(N_1[u_L],\ N_2[v_L]) + Masn(N_1[u_R]',\ N_2[v_R]'), \\ &\qquad Masn(N_1[u_L],\ N_2[v_L]') + Masn(N_1[u_R]',\ N_2[v_R]), \\ &\qquad Masn(N_1[u_L],\ N_2[v_R]) + Masn(N_1[u_R]',\ N_2[v_L]'), \\ &\qquad Masn(N_1[u_L],\ N_2[v_R]') + Masn(N_1[u_R]',\ N_2[v_L]), \\ &\qquad Masn(N_1[u_L]',\ N_2[v_L]) + Masn(N_1[u_R],\ N_2[v_R]'), \\ &\qquad Masn(N_1[u_L]',\ N_2[v_L]') + Masn(N_1[u_R],\ N_2[v_R]), \\ &\qquad Masn(N_1[u_L]',\ N_2[v_R]) + Masn(N_1[u_R],\ N_2[v_L]'), \\ &\qquad Masn(N_1[u_L]',\ N_2[v_R]') + Masn(N_1[u_R],\ N_2[v_L])\}. \end{aligned}$$

Lemma 1 implies that we can compute $Masn(N_1[u], N_2[v])$ for all (u, v) in $V(N_1) \times V(N_2)$ by employing dynamic programming in a bottom-up manner, e.g., by evaluating all pairs in $V(N_1) \times V(N_2)$ in increasing order in the lexicographic ordering $\mathcal{O}$ of $V(N_1) \times V(N_2)$ where the nodes in each $V(N_i)$ are postordered. The resulting algorithm (Algorithm *ComputeMasn*) is displayed in Fig. 5.

Next, we analyze the time complexity of Algorithm *ComputeMasn*.

Lemma 2. *If N is a level-1 phylogenetic network then the number of hybrid nodes in N is at most $n-1$.*

Proof. Define T_N as the rooted directed graph obtained from N as follows: For every hybrid node h in N, remove $lhe(h)$ and $rhe(h)$, contract $sn(h)$ and all nodes on the two clipped merge paths of h to a single node s, and then add a directed edge from s to h. Clearly, every node with indegree 2 in N has indegree equal to 1 in T_N, and none of the contractions increase the indegree of any node, so T_N is a tree. Furthermore, T_N contains n leaves. Thus, the number of internal nodes in T_N with outdegree > 1 is at most $n-1$.

Finally, observe that every split node in N corresponds to a distinct internal node in T_N with outdegree > 1 and that the number of hybrid nodes in N equals the number of split nodes in N since N is a level-1 phylogenetic network. □

Lemma 3. *If N is a level-1 phylogenetic network then the total number of nodes in N is* $O(n)$.

Proof. Let z_{ij} denote the number of nodes in N which have i incoming edges and j outgoing edges. By the definition of a phylogenetic network, the total number t of nodes in N is $z_{02}+z_{10}+z_{12}+z_{20}+z_{21}+z_{22}$. For every $u \in V(N)$, let $in(u)$ and $out(u)$ denote the number of incoming and outgoing edges incident to u. Since

$$\sum_{u\in V(N)} in(u) = z_{02}\cdot 0 + (z_{10}+z_{12})\cdot 1 + (z_{20}+z_{21}+z_{22})\cdot 2,$$

$$\sum_{u\in V(N)} out(u) = (z_{10}+z_{20})\cdot 0 + z_{21}\cdot 1 + (z_{02}+z_{12}+z_{22})\cdot 2$$

and

$$\sum_{u\in V(N)} in(u) = \sum_{u\in V(N)} out(u), \text{ we have } \quad z_{12} = z_{10} + 2z_{20} + z_{21} - 2z_{02}.$$

Next, $z_{20}+z_{21}+z_{22}$ (the number of hybrid nodes) is at most $n-1$ by Lemma 2 and $n = z_{10}+z_{20}$, which gives us $z_{12} < 2n$. Hence, $t < 1+n+2n+(n-1) = O(n)$. □

Lemma 4. *The running time of* Algorithm *ComputeMasn is* $O(n^2)$.

Proof. By Lemma 3, the algorithm evaluates $O(n^2)$ pairs of nodes. For each such pair (u, v), if neither u nor v is a leaf then it takes constant time to compute the *Masn*-values from previously computed values. If u is a leaf then the value of $|\Lambda(N_1[u]) \cap \Lambda(N_2[v])|$ can be obtained in constant time by associating a binary vector $L(w)$ of length n to each $w \in V(N_1) \cup V(N_2)$, where the ith bit of $L(w)$ is set to 1 if and only if leaf i is a descendant of w (note that all $L(w)$-vectors can be computed in advance in $O(n^2)$ time by using a postorder traversal technique), and checking if bit u in $L(v)$ equals 1. The case where v is a leaf is analogous. □

Algorithm *ComputeMasn* can be modified to compute the set of leaves in a maximum agreement subnetwork without increasing the asymptotic running time by also recording

information about how each *Masn*-value is obtained as it is computed, e.g., by saving pointers. To obtain an actual maximum agreement subnetwork from such a set of leaves, we may use a standard traceback technique.

Theorem 5. *Given two level-*1 *phylogenetic networks with* n *leaves, a maximum agreement subnetwork can be computed in* $O(n^2)$ *time.*

By extending this technique, we get the following.

Corollary 6. *Given a level-*f_1 *phylogenetic network* N_1 *and a level-*f_2 *phylogenetic network* N_2*, a maximum agreement subnetwork of* N_1 *and* N_2 *can be computed in* $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$ *time, where* $V(N_i)$ *denotes the set of nodes of* N_i *for* $i \in \{1, 2\}$.

Proof. A maximum agreement subnetwork of N_1 and N_2 can be computed by modifying the dynamic programming algorithm for two level-1 phylogenetic networks described above. For any level-f phylogenetic network N, $u \in V(N)$, and binary string $b = b_1 b_2 \cdots b_f$ of length f, define $N[u]^b$ as the subnetwork of N rooted at u in which for each hybrid node h_i in the same maximal biconnected component as u, either its left or right hybrid edge has been deleted according to whether b_i equals 0 or 1. (Note that the maximal biconnected component of N containing u is a tree in $N[u]^b$ because every one of its nodes has exactly one parent. Hence, $V((N[u]^b)[u_L])$ and $V((N[u]^b)[u_R])$ are disjoint.) Now, we define $Masn(N_1[u], N_2[v])$ and $Diag(N_1[u], N_2[v])$ like in Lemma 1, but change $Match(N_1[u], N_2[v])$ to be the maximum value of $Masn((N_1[u]^{b_1})[u_L], (N_2[v]^{b_2})[v_L]) + Masn((N_1[u]^{b_1})[u_R], (N_2[v]^{b_2})[v_R])$ and $Masn((N_1[u]^{b_1})[u_L], (N_2[v]^{b_2})[v_R]) + Masn((N_1[u]^{b_1})[u_R], (N_2[v]^{b_2})[v_L])$ taken over all binary strings b_1 and b_2 of length f_1 and f_2, respectively.

As before, using dynamic programming in a bottom-up manner, we compute $Masn(N_1[u], N_2[v])$ for all $(u, v) \in V(N_1) \times V(N_2)$. We also compute and store $Masn(N_1[u]^{b_1}, N_2[v])$, $Masn(N_1[u], N_2[v]^{b_2})$, and $Masn(N_1[u]^{b_1}, N_2[v]^{b_2})$ for every binary string b_1 of length f_1 and every binary string b_2 of length f_2. The algorithm's total running time becomes $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$. □

Hence, MASN with $k = 2$ and f upper-bounded by $O(\log n)$ is solvable in polynomial time.

4. NP-hardness of MASN for $k = 3$

In this section, we prove that MASN is NP-hard for every fixed $k \geqslant 3$. Our reduction is a non-trivial modification of the NP-hardness proof by Amir and Keselman [2] for MAST restricted to three trees with unbounded degrees. Note that the definition of MASN requires all nodes to have outdegree at most two, so the fact that MAST with unbounded degrees is NP-hard does not immediately imply that MASN is NP-hard.

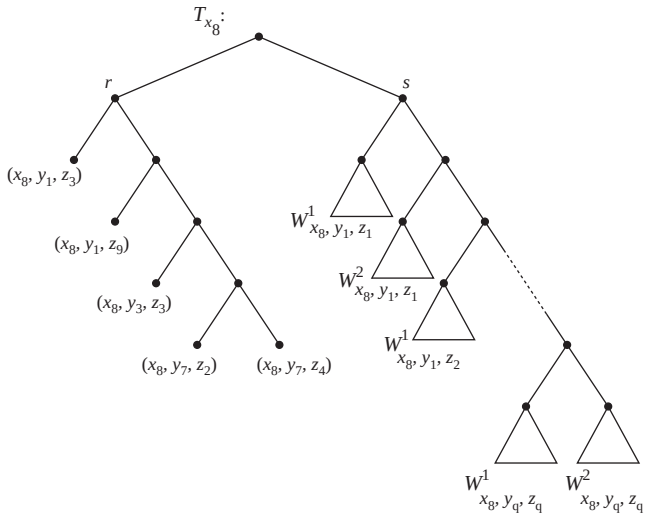

Fig. 6. Assume $M_{x_8} = \{(x_8, y_1, z_3), (x_8, y_1, z_9), (x_8, y_3, z_3), (x_8, y_7, z_2), (x_8, y_7, z_4)\}$. R_{x_8} and S_{x_8} are the subtrees of T_{x_8} rooted at the nodes marked r and s, respectively.

4.1. Three-dimensional matching (3DM)

Instance: A set $M \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets and $X = \{x_1, \ldots, x_q\}$, $Y = \{y_1, \ldots, y_q\}$, and $Z = \{z_1, \ldots, z_q\}$.

Question: Is there a subset M' of M with $|M'| = q$ such that M' is a matching, i.e., such that for every pair $e_1, e_2 \in M'$ it holds that e_1 and e_2 differ in all coordinates?

3DM is known to be NP-complete (see, e.g., [13]). To prove the NP-hardness of MASN, we describe a polynomial-time reduction from 3DM. Given an arbitrary instance of 3DM, construct an instance $(L, \mathcal{N})$ of MASN with three phylogenetic networks $\mathcal{N}=\{N_1, N_2, N_3\}$ for L as follows.

Take $L = M \cup W \cup B$, where W is a set of $2q^6$ arbitrary elements not in M, and B is a set of $2q^7$ arbitrary elements not in M or W. Let B_1 and B_2 be two binary trees with q^7 leaves each, distinctly labeled by B. For every $(x_i, y_j, z_k) \in X \times Y \times Z$, define $W^1_{x_i,y_j,z_k}$ and $W^2_{x_i,y_j,z_k}$ to be two binary trees with q^3 leaves each, distinctly labeled by W. Next, for every $x_i \in X$, define (1) M_{x_i} as the subset of M containing all triples of the form (x_i, y, z) where $y \in Y$ and $z \in Z$; (2) R_{x_i} as a binary caterpillar tree with leaves distinctly labeled by M_{x_i}; (3) S_{x_i} as the tree obtained from the binary caterpillar tree with $2q^2$ leaves by replacing them (in order of non-decreasing distance from the root) with the roots of $W^1_{x_i,y_1,z_1}$, $W^2_{x_i,y_1,z_1}$, $W^1_{x_i,y_1,z_2}$, …, $W^2_{x_i,y_q,z_q}$; (4) S'_{x_i} as the tree obtained from the binary caterpillar tree with $2q^2$ leaves by replacing them (in order of non-decreasing distance from the root) with the roots of $W^2_{x_i,y_q,z_q}$, $W^1_{x_i,y_q,z_q}$, $W^2_{x_i,y_q,z_{q-1}}$, …, $W^1_{x_i,y_1,z_1}$; (5) T_{x_i} as a binary tree with a root node connected to the roots of R_{x_i} and S_{x_i}; and (6) T'_{x_i} as a binary tree with a root node connected to the roots of R_{x_i} and S'_{x_i}. Define M_{y_i}, M_{z_i}, R_{y_i}, R_{z_i}, etc. for every $y_i \in Y$ and $z_i \in Z$ analogously. See Fig. 6 for an example.

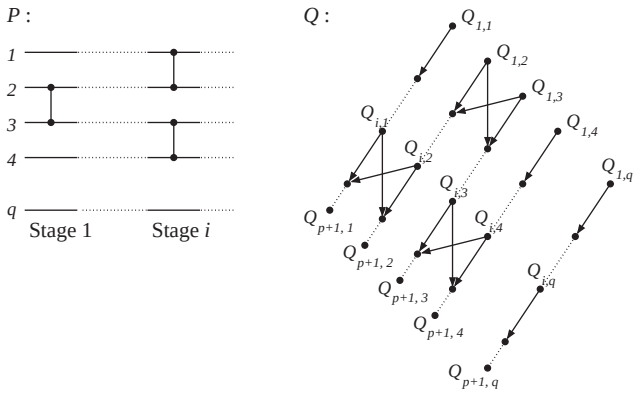

Fig. 7. The sorting network P on the left yields a directed acyclic graph Q.

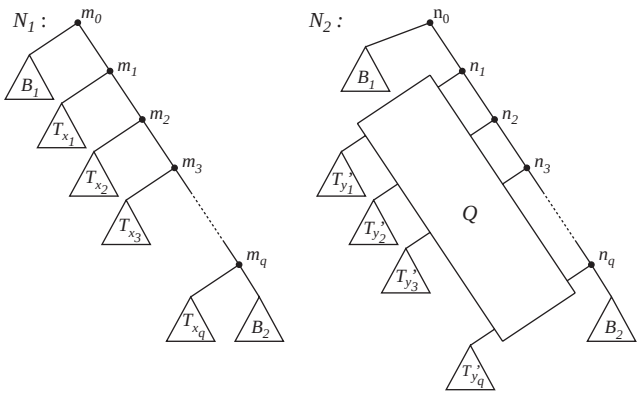

Fig. 8. The phylogenetic networks N_1 and N_2.

Next, let P be any sorting network (see, e.g., [8]) for q elements with a polynomial number p of comparator stages. Construct a directed acyclic graph Q from P with $(p+1)\cdot q$ nodes $\{Q_{i,j} \mid 1 \leqslant i \leqslant p+1,\ 1 \leqslant j \leqslant q\}$ such that there is a directed edge $(Q_{i,j},\ Q_{i+1,j})$ for every $1 \leqslant i \leqslant p$ and $1 \leqslant j \leqslant q$, and two directed edges $(Q_{i,j},\ Q_{i+1,k})$ and $(Q_{i,k},\ Q_{i+1,j})$ for every comparator (j,k) at stage i in P for $1 \leqslant i \leqslant p$, as illustrated in Fig. 7.

We now let N_1 be a phylogenetic network (in fact, a leaf-labeled binary tree) obtained by attaching B_1, B_2, and $T_{x_1}, \ldots, T_{x_q}$ to a path $(m_0, m_1, m_2, \ldots, m_q)$ so that m_0 becomes the root of N_1 and the root of B_1 is a child of m_0, the root of each T_{x_i} is a child of m_i, and the root of B_2 is the second child of m_q. See Fig. 8. The phylogenetic network N_2 is obtained by attaching B_1, B_2, Q, and $T'_{y_1}, \ldots, T'_{y_q}$ to a path $(n_0, n_1, n_2, \ldots, n_q)$ so that n_0 becomes the root of N_2 and the root of B_1 is a child of n_0, each node $Q_{1,j}$ in Q is a child of n_j and each node $Q_{p+1,j}$ in Q coincides with the root of T'_{y_j}, and the root of B_2 is the second child of n_q. Next, N_3 is defined in the same way as N_2 but using T'_{z_j} instead of T'_{y_j}. Finally, for each node in N_2 or N_3 having indegree 1 and outdegree 1, contract its outgoing edge.

Lemma 7. *There exists an agreement subnetwork of* (N_1, N_2, N_3) *with* $2q^7 + 2q^4 + q$ *leaves if and only if* M *has a matching of size* q.

Proof. Suppose M has a matching M' of size q. Then for each x_i, there is precisely one triple of the form (x_i, y, z) in M'. For any $(x_i, y_j, z_k) \in X \times Y \times Z$, denote by V_{x_i,y_j,z_k} the set of all leaves in $W^1_{x_i,y_j,z_k}$ and $W^2_{x_i,y_j,z_k}$. Let $C = M' \cup \bigcup_{(x_i,y,z)\in M'} V_{x_i,y,z}$ and let T be $N_1 \mid (B \cup C)$. Now consider the structure of $N_2 \mid (B \cup C)$ and $N_3 \mid (B \cup C)$. First, observe that for each (x_i, y_j, z_k) in M', there exists an agreement subnetwork of T_{x_i}, T'_{y_j}, and T'_{z_k} containing the $(1 + 2q^3)$ leaves in $\{(x_i, y_j, z_k)\} \cup V_{x_i,y_j,z_k}$. Next, since P is a sorting network, there are q disjoint paths in Q from $(Q_{1,1}, Q_{1,2}, \ldots, Q_{1,q})$ to $(Q_{p+1,\pi(1)}, Q_{p+1,\pi(2)}, \ldots, Q_{p+1,\pi(q)})$ for any given permutation π of $\{1, 2, \ldots, q\}$; in particular, this holds for the permutations π_y and π_z defined by the relations $\pi_y(i) = j$ and $\pi_z(i) = k$ for all $(x_i, y_j, z_k) \in M'$. This means that T is a subgraph of $N_2 \mid (B \cup C)$ and $N_3 \mid (B \cup C)$. Thus, T is an agreement subnetwork of (N_1, N_2, N_3) with $|B| + q \cdot (1 + 2q^3)$ leaves.

Conversely, suppose there exists an agreement subnetwork T with leaf set $L' \subseteq L$ such that $|L'| = 2q^7 + 2q^4 + q$. Write $M' = L' \cap M$ and $W' = L' \cap W$. By the pigeonhole principle, $|M'| + |W'| \geqslant 2q^4 + q$. Also, at least one leaf in B_1 and at least one leaf ℓ in B_2 must be included in L'. It follows that the root of T corresponds to the roots of N_1, N_2, and N_3, and for any two triples $e = (x_{i_1}, y_{j_1}, z_{k_1})$ and $f = (x_{i_2}, y_{j_2}, z_{k_2})$ in M, if e and f agree on at least one coordinate then they cannot both belong to L'. (To see this, if $i_1 \neq i_2$ and $j_1 = j_2$, then e and f would appear in different subtrees of the form T_{x_i} in N_1 but in the same subtree of the form T'_{y_j} in N_2, so, e.g., $N_1 \mid \{e, f, \ell\}$ and $N_2 \mid \{e, f, \ell\}$ would differ, which contradicts that $\{e, f, \ell\}$ are leaves in T. If $i_1 = i_2$ and $j_1 \neq j_2$ then $|W'| \leqslant (q-1) \cdot 2q^3$ since otherwise there would have to exist a w in W' such that w appears in $T'_{y_{j_1}}$ and then $N_1 \mid \{e, f, w\}$ and $N_2 \mid \{e, f, w\}$ would differ; thus, $|M'| + |W'| \leqslant |M| + |W'| \leqslant q^3 + (q-1) \cdot 2q^3 \leqslant 2q^4 - q^3$, contradicting that $|M'| + |W'| \geqslant 2q^4 + q$. The cases $(i_1 \neq i_2, k_1 = k_2)$ and $(i_1 = i_2, k_1 \neq k_2)$ are analogous.) Thus, M' is a matching of M. Next, assume that $|M'| < q$. Then W' has cardinality $|L'| - |M'| - |L' \cap B| > 2q^4$. This implies that W' contains leaves from at least three different subtrees of the form W^m_{x,y_j,z_k} for some fixed $x \in X$, but at most two such leaves can appear in the same S'_{y_j} and in the same S'_{z_k} for any $y_j \in Y$ and $z_k \in Z$. Contradiction. Hence, $|M'| \geqslant q$. □

From the above, we obtain:

Theorem 8. *MASN is* NP*-hard even if restricted to* $k = 3$.

5. Final remarks

An open problem is to determine the computational complexity of MASN restricted to two level-f phylogenetic networks where f is unbounded. If it is NP-hard, can it be approximated efficiently in polynomial time? We would also like to know if it is possible to improve the running time of our algorithm for two level-1 phylogenetic networks.

References

[1] J. Alber, J. Gramm, R. Niedermeier, Faster exact algorithms for hard problems: a parameterized point of view, Discrete Math. 229 (2001) 3–27.

[2] A. Amir, D. Keselman, Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms, SIAM J. Comput. 26 (6) (1997) 1656–1669.

[3] C.H. Bennett, M. Li, B. Ma, Chain letters and evolutionary histories, Sci. Amer. 288 (6) (2003) 76–81.

[4] M. Bonet, C. Phillips, T. Warnow, S. Yooseph, Constructing evolutionary trees in the presence of polymorphic characters, SIAM J. Comput. 29 (1) (1999) 103–131.

[5] D. Bryant, Building trees, hunting for trees, and comparing trees: theory and methods in phylogenetic analysis, Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, 1997.

[6] Canterbury Tales Project, De Montfort University, Harvard University, University of Leeds, National Library of Wales, Keio University, Brigham Young University, Virginia Polytechnic Institute and State University (Virginia Tech), University of Münster, and New York University, Website: http://www.cta.dmu.ac.uk/projects/ctp/.

[7] R. Cole, M. Farach-Colton, R. Hariharan, T. Przytycka, M. Thorup, An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees, SIAM J. Comput. 30 (5) (2000) 1385–1404.

[8] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, The MIT Press, MA, 1990.

[9] M. Farach, T. Przytycka, M. Thorup, On the agreement of many trees, Inform. Process. Lett. 55 (1995) 297–301.

[10] M. Farach, M. Thorup, Fast comparison of evolutionary trees, in: Proc. Fifth Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'94), 1994, pp. 481–488.

[11] M. Farach, M. Thorup, Sparse dynamic programming for evolutionary-tree comparison, SIAM J. Comput. 26 (1) (1997) 210–230.

[12] C.R. Finden, A.D. Gordon, Obtaining common pruned trees, J. Classification 2 (1985) 255–276.

[13] M. Garey, 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, A. Östlin, On the complexity of constructing evolutionary trees, J. Combin. Optim. 3 (2–3) (1999) 183–197.

[15] D. Gusfield, S. Eddhu, C. Langley, Efficient reconstruction of phylogenetic networks with constrained recombination, in: Proc. Comput. Systems Bioinformatics Conf. (CSB2003), 2003, pp. 363–374.

[16] J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, Math. Biosci. 98 (2) (1990) 185–200.

[17] J. Hein, T. Jiang, L. Wang, K. Zhang, On the complexity of comparing evolutionary trees, Discrete Appl. Math. 71 (1996) 153–169.

[18] D.H. Huson, T. Dezulian, T. Klöpper, M. Steel, Phylogenetic super-networks from partial trees, in: Proc. Fourth Workshop on Algorithms in Bioinformatics (WABI 2004), 2004, pp. 388–399.

[19] J. Jansson, W.-K. Sung, Inferring a level-1 phylogenetic network from a dense set of rooted triplets. in: Proc. Tenth Internat. Comput. and Combin. Conf. (COCOON 2004), 2004, pp. 462–472.

[20] M.-Y. Kao, Tree contractions and evolutionary trees, SIAM J. Comput. 27 (6) (1998) 1592–1616.

[21] M.-Y. Kao, T.-W. Lam, W.-K. Sung, H.-F. Ting, An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings, J. Algorithms 40 (2) (2001) 212–233.

[22] W.-H. Li, Molecular Evolution, Sinauer Associates, Inc., Sunderland, 1997.

[23] C.R. Linder, B.M.E. Moret, L. Nakhleh, T. Warnow, Network (reticulate) evolution: biology, models, and algorithms, Tutorial Presented at the 9th Pacific Symposium on Biocomputing (PSB 2004), 2004.

[24] A. Meyers, R. Yangarber, R. Grishman, Alignment of shared forests for bilingual corpora, in: Proc. 16th Internat. Conf. on Computational Linguistics (COLING-96), 1996, pp. 460–465.

[25] L. Nakhleh, J. Sun, T. Warnow, C.R. Linder, B.M.E. Moret, A. Tholse, Towards the development of computational tools for evaluating phylogenetic reconstruction methods, in: Proc. Eighth Pacific Symp. on Biocomputing (PSB 2003), 2003, pp. 315–326.

[26] L. Nakhleh, T. Warnow, C.R. Linder, Reconstructing reticulate evolution in species—theory and practice, in: Proc. Eighth Ann. Internat. Conf. on Research in Comput. Molecular Biology (RECOMB 2004), 2004, pp. 337–346.

[27] D. Posada, K.A. Crandall, Intraspecific gene genealogies: trees grafting into networks, Trends Ecol. Evol. 16 (1) (2001) 37–45.
[28] J.C. Setubal, J. Meidanis, Introduction to Computational Molecular Biology, PWS Publishing Company, Boston, 1997.
[29] M. Steel, T. Warnow, Kaikoura tree theorems: computing the maximum agreement subtree, Inform. Process. Lett. 48 (1993) 77–82.
[30] L. Wang, K. Zhang, L. Zhang, Perfect phylogenetic networks with recombination, J. Comput. Biol. 8 (1) (2001) 69–78.