Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 91 (2004) 134–147
www.elsevier.com/locate/entcs

# Computing the Maximum Agreement of Phylogenetic Networks

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

[a] *School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543.*
*Email:* {choychih,jansson,ksung}@comp.nus.edu.sg

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

**Abstract**

We introduce the maximum agreement phylogenetic subnetwork problem (MASN) of finding a 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 contains at most $f$ nodes having indegree 2 in $N$. Our algorithm can be extended to yield a polynomial-time algorithm for two level-$f$ phylogenetic networks $N_1, N_2$ for any $f$ which is upper-bounded by a constant; more precisely, its running time is $O(|V(N_1)| \cdot |V(N_2)| \cdot 4^f)$, where $V(N_i)$ denotes the set of nodes of $N_i$.

*Keywords:* phylogenetic network comparison, maximum agreement subnetwork, algorithm, computational complexity

## 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 [2,15,19]. 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 hybridization


doi:10.1016/j.entcs.2003.12.009

which suggest convergence between objects cannot be adequately represented in a single tree structure [4,11,12,17,18,21]. 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., [4,11,17,18,21]). 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 study the computational complexity of MASN and some of its restrictions to determine when the problem can be solved by efficient algorithms.

Further motivation for studying 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 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 species during phylogenetic reconstruction, but also to measure the similarity of a given set of trees [7,9,14] or to estimate a classification's stability to small changes in the data [9]. Moreover, MAST-based algorithms have been used to prepare and improve bilingual context-using dictionaries for automated language translation systems [5,16].

[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., [1] or [20] for a formal definition. MAST is also referred to as the maximum homeomorphic subtree problem (MHT) by some researchers.

### *1.1 Problem definition*

Let $L$ be a finite set. A *phylogenetic network 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 two, contracting its outgoing edge (any set of multiple edges between two nodes is replaced by a single edge). See Fig. 1 for an example.

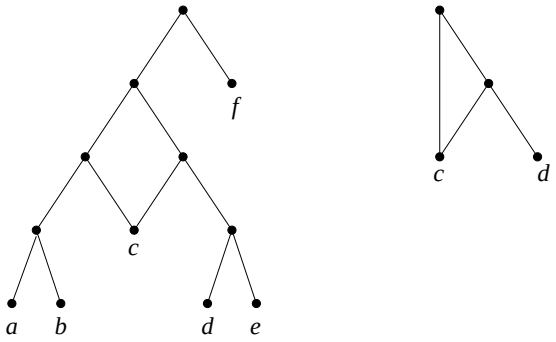


Fig. 1. Let $N$ be the phylogenetic network on the left. Then $N \mid \{c, d\}$ is the phylogenetic network on the right.

Given a set $\mathcal{N} = \{N_1, N_2, ..., N_k\}$ 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 $N_1 \mid L'$, $N_2 \mid L'$, ..., 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 maximum agreement phylogenetic subnetwork problem* (MASN) is: Given a finite set $L$ and a set $\mathcal{N}$ of phylogenetic networks for $L$, find a maximum agreement subnetwork of $\mathcal{N}$.

Throughout this paper, $n$ and $k$ represent the cardinalities of $L$ and $\mathcal{N}$, respectively, in the problem definition above.

### 1.2 Previous results

A survey of existing algorithms for *constructing* phylogenetic networks can be found in [18]; see also [4], [11], and [21]. A method for *comparing* two given phylogenetic networks (more exactly, measuring their similarity in order to assess the topological accuracy of different phylogenetic network construction methods) was proposed in [17].

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 maximum agreement subtree problem (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 [9] 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 [20] 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., [5,8,13,14]). Today, the fastest currently known algorithm for MAST for two trees, invented by Kao, Lam, Sung, and Ting [14], 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 [1] considered the case of $k \geq 3$ 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, Przytycka, and Thorup [7] 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 [3] proposed a conceptually different algorithm with the same running 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. This is the first polynomial-time algorithm ever for this problem. Our algorithm can be extended to solve MASN for two level-$f$ phylogenetic networks $N_1$ and $N_2$ in $O(|V(N_1)| \cdot |V(N_2)| \cdot 4^f)$ time (where $V(N_i)$ denotes the set of nodes of $N_i$), which is polynomial in the input size for any $f$ which is upper-bounded by a

constant. Next, in Section 4, we prove that in the general case (i.e., for level-$f$ phylogenetic networks where $f$ is unbounded), the maximum agreement phylogenetic subnetwork problem 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 0 (i.e., the root of $N$) or 1 as *tree nodes* and nodes with indegree 2 as *hybrid nodes*.

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 the two paths from the children of $s$ to $h$ are called *merge paths of* $h$. Any node $u$ which belongs to a merge path of some node $v$, where $v \neq u$, is referred to as a *merge path node*. If $h$ only has one split node, we denote it by $sn(h)$. See Fig. 2.

$N$ is said to be a *level-$f$ phylogenetic network* if every biconnected component in the undirected graph obtained from $N$ by replacing each directed edge by an undirected edge contains at most $f$ nodes that are hybrid nodes in $N$. Note that $N$ is a tree if $f = 0$. If $f = 1$ then every node in $N$ belongs to at most one pair of merge paths[2]. 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 [5] or [14]. In this section, we consider how to compute a maximum agreement subnetwork of two level-$f$ phylogenetic networks for $f > 0$. We present an $O(n^2)$-time algorithm for the

[2] The biological relevance of level-1 phylogenetic networks (there referred to as *galled-trees*) is discussed in [11].

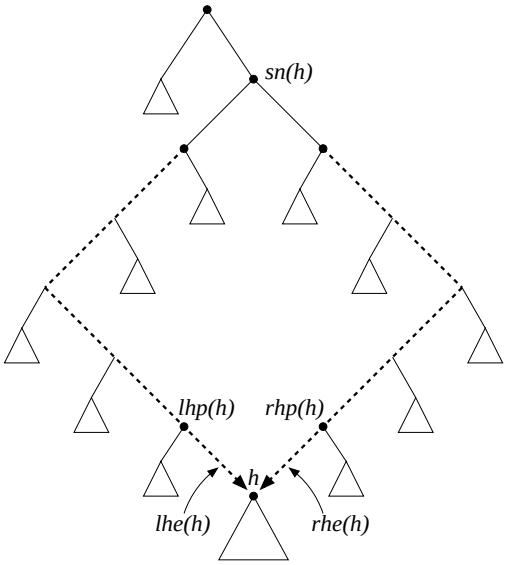


Fig. 2. Here, $h$ is a hybrid node with a unique split node. The figure shows $sn(h)$, $lhp(h)$, $rhp(h)$, $lhe(h)$, $rhe(h)$, and as dashed lines, two merge paths of $h$.

case $f = 1$ which can be extended to solve MASN in polynomial time for any $f$ which is upper-bounded by a constant.

From this point onward, we assume that some arbitrary left-to-right ordering of the children of every node has been fixed. 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 $N$. 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 $u_L = c$ and $u_R = \emptyset$. Since $N$ is a level-1 phylogenetic network, if $u \in V(N)$ is a merge path node then there exists a unique hybrid node belonging to the same merge path as $u$; we denote this node by $hn(u)$. 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 $u$. If $u \in V(N)$ is a merge path node in $N$ then $N[u]'$ is defined as the subnetwork of $N[u]$ where $N[hn(u)]$ and $hn(u)$'s incoming edge have been removed. Otherwise, if $u$ is not a merge path node in $N$ then $N[u]'$ is equal to $N[u]$. Finally, $N[\emptyset]$ refers to the empty network with no nodes or edges. See Fig. 3 for an example.

For any two level-1 phylogenetic networks $N_1$ and $N_2$, define $Masn(N_1, N_2)$ as the number of leaves in a maximum agreement subnetwork of $N_1$ and $N_2$. If $N_1$ or $N_2$ is the empty network then $Masn(N_1, N_2)$ is equal to 0. Otherwise,

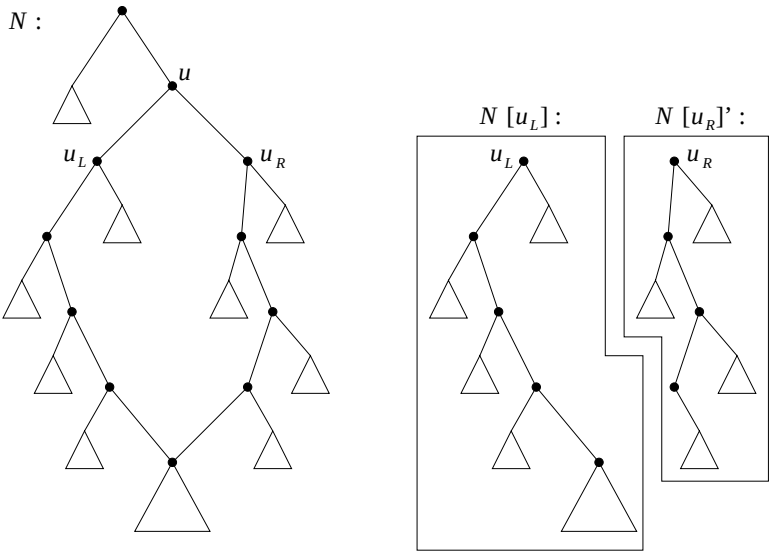


Fig. 3. $N$ is a level-1 phylogenetic network and $u$ is a split node in $N$. $N[u_L]$ and $N[u_R]'$ are the subnetworks of $N$ shown on the right.

$Masn(N_1, N_2)$ can be expressed recursively using the following lemma which is a straightforward generalization of the main lemma in [20] for MAST.

**Lemma 3.1** *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*

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

*and*

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

$$Masn(N_1[u_L]', N_2[v_L]) + Masn(N_1[u_R], N_2[v_R]'),$$
$$Masn(N_1[u_L]', N_2[v_L]') + Masn(N_1[u_R], N_2[v_R]),$$
$$Masn(N_1[u_L]', N_2[v_R]) + Masn(N_1[u_R], N_2[v_L]'),$$
$$Masn(N_1[u_L]', N_2[v_R]') + Masn(N_1[u_R], N_2[v_L])\}.$$

Lemma 3.1 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. 4.

**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 3.1.
**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. 4. A dynamic programming algorithm for computing all values of $Masn$.

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

**Lemma 3.2** *If $N$ is a level-1 phylogenetic network then the total number of nodes in $N$ is $O(n)$.*

**Proof.** (Given in the full-length version of our paper.) □

**Lemma 3.3** *The running time of* Algorithm *ComputeMasn is $O(n^2)$.*

**Proof.** By Lemma 3.2, 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 $i$th 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 prefix walk 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 $L'$, we can compute $N_1 \mid L'$ and $N_2 \mid L'$ and then delete all edges which are not in $N_2 \mid L'$ from $N_1 \mid L'$.

**Theorem 3.4** *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 3.5** *Given two level-$f$ phylogenetic networks $N_1$ and $N_2$, a maximum agreement subnetwork of $N_1$ and $N_2$ can be computed in $O(|V(N_1)| \cdot |V(N_2)| \cdot 4^f)$ time, where $V(N_i)$ denotes the set of nodes of $N_i$.*

**Proof.** (Given in the full-length version of our paper.) □

Hence, MASN with $k = 2$ and $f$ upper-bounded by a constant 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 $k \geq 3$. Our reduction is a non-trivial modification of the NP-hardness proof by Amir and Keselman [1] 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.

*Three-Dimensional Matching (3DM)*

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

**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., [10]). 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 $S'_{x_i}$. Furthermore, 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. 5 for an example.

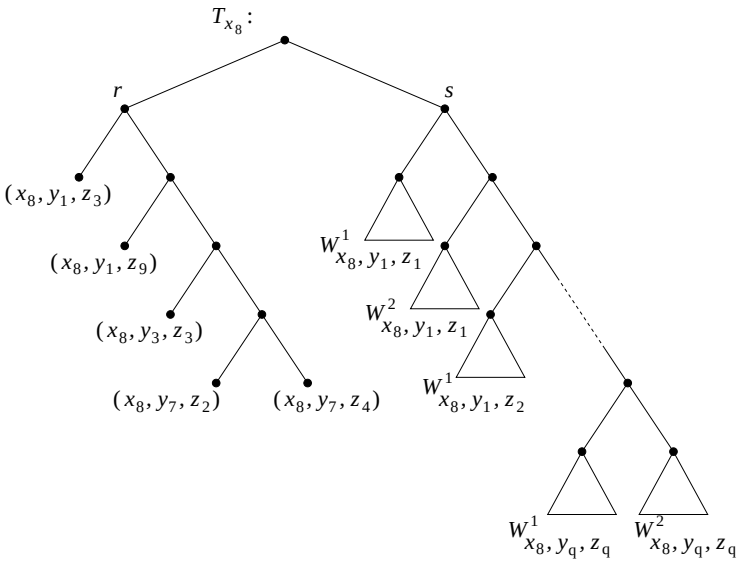


Fig. 5. Assume $M_{x_8} = \{(x_8, y_1, z_3), (x_8, y_1, z_9), (x_8, y_3, z_3), (x_8, y_7, z_2), (x_8, y_7, z_4)\}$. $R_{x_8}$ and $S_{x_8}$ are the subtrees of $T_{x_8}$ rooted at the nodes marked $r$ and $s$, respectively.

Next, let $P$ be any sorting network (see, e.g., [6]) of polynomial depth $p$ for $q$ elements. Construct a directed acyclic graph $Q$ from $P$ with $(p+1) \cdot q$ nodes $\{Q_{i,j} \mid 1 \le i \le p+1,\ 1 \le j \le q\}$ such that there is a directed edge $(Q_{i,j},\ Q_{i+1,j})$

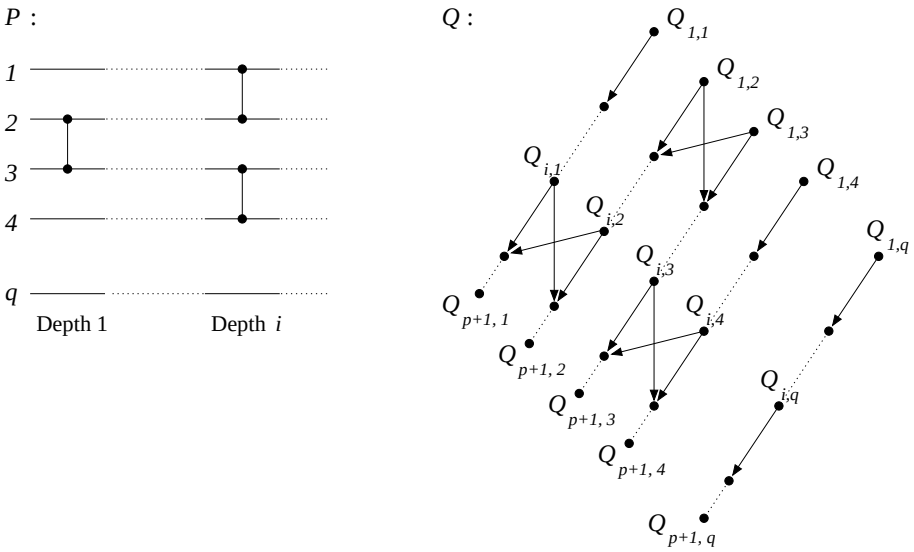


Fig. 6. The sorting network $P$ on the left yields a directed acyclic graph $Q$.

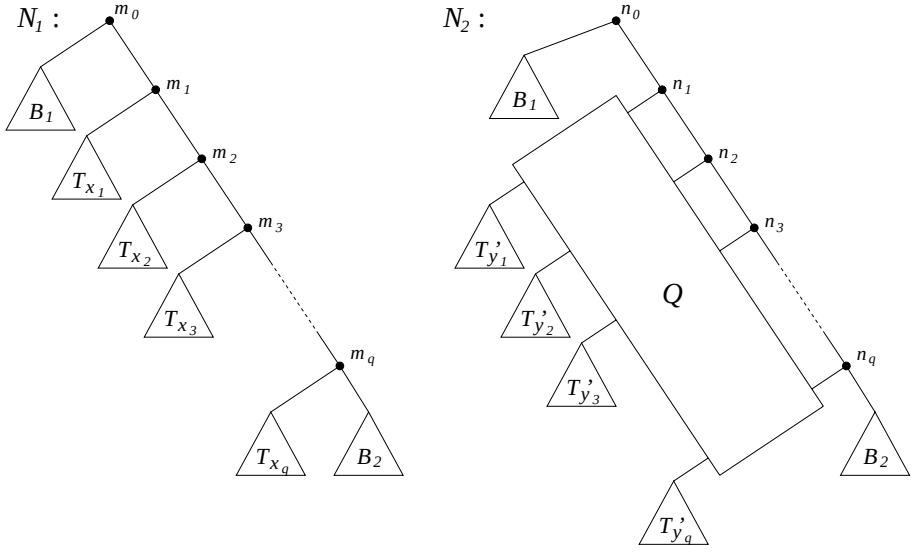


Fig. 7. The phylogenetic networks $N_1$ and $N_2$.

for every $1 \leq i \leq p$ and $1 \leq j \leq q$, 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 depth $i$ in $P$ for $1 \leq i \leq p$, as

illustrated in Fig. 6.

We now let $N_1$ be a phylogenetic network (in fact, a leaf-labeled binary tree) obtained by attaching $B_1$, $B_2$, and $T_{x_1}, ..., T_{x_q}$ to a path $(m_0, m_1, m_2, ..., m_q)$ 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. 7. The phylogenetic network $N_2$ is obtained by attaching $B_1$, $B_2$, $Q$, and $T'_{y_1}, ..., T'_{y_q}$ to a path $(n_0, n_1, n_2, ..., n_q)$ 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 4.1** *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 $N_1 \,|\, (B \cup C)$. Now consider the structure of $N_2 \,|\, (B \cup C)$ and $N_3 \,|\, (B \cup C)$. 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 $(Q_{1,1}, Q_{1,2}, ..., Q_{1,q})$ to $(Q_{p+1,\pi(1)}, Q_{p+1,\pi(2)}, ..., Q_{p+1,\pi(q)})$ for any given permutation $\pi$ of $\{1, 2, ..., q\}$; in particular, this holds for the permutations $\pi_y$ and $\pi_z$ 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 $N_2 \,|\, (B \cup C)$ and $N_3 \,|\, (B \cup C)$. 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 a 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, $|M'| + |W'| \geq 2q^4 + q$. Also, at least one leaf in $B_1$ and at least one leaf $\ell$ 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., $N_1 \,|\, \{e, f, \ell\}$ and $N_2 \,|\, \{e, f, \ell\}$ would differ, which contradicts that $\{e, f, \ell\}$ are leaves in $T$. If $i_1 = i_2$ and $j_1 \neq j_2$ then $|W'| \leq (q - 1) \cdot 2q^3$ since otherwise there would have to exist a $w$ in $W'$ such that $w$ appears in $T'_{y_{j_1}}$ and then $N_1 \,|\, \{e, f, w\}$ and $N_2 \,|\, \{e, f, w\}$

would differ; thus, $|M'| + |W'| \leq |M| + |W'| \leq q^3 + (q-1) \cdot 2q^3 \leq 2q^4 - q^3$, contradicting that $|M'| + |W'| \geq 2q^4 + q$. 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, $|M'| \geq q$. □

From the above, we obtain:

**Theorem 4.2** *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] Amir, A. 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.

[2] Bennett, C. H., M. Li and B. Ma, Chain letters and evolutionary histories. *Scientific American*, to appear.

[3] Bryant, D., *Building Trees, Hunting for Trees, and Comparing Trees: Theory and Methods in Phylogenetic Analysis*. PhD thesis, University of Canterbury, Christchurch, New Zealand, 1997.

[4] Bryant, D. and V. Moulton, NeighborNet: an agglomerative method for the construction of planar phylogenetic networks. In *Proceedings of the 2nd Workshop on Algorithms in Bioinformatics* (WABI 2002), volume 2452 of *Lecture Notes in Computer Science*, pages 375–391. Springer-Verlag Berlin Heidelberg, 2002.

[5] Cole, R., 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.

[6] Cormen, T., C. Leiserson and R. Rivest, *Introduction to Algorithms*. The MIT Press, Massachusetts, 1990.

[7] Farach, M., T. Przytycka and M. Thorup, On the agreement of many trees. *Information Processing Letters*, 55:297–301, 1995.

[8] Farach, M. and M. Thorup, Sparse dynamic programming for evolutionary-tree comparison. *SIAM Journal on Computing*, 26(1):210–230, 1997.

[9] Finden, C. R. and A. D. Gordon, Obtaining common pruned trees. *Journal of Classification*, 2:255–276, 1985.

[10] Garey, M. and D. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[11] Gusfield, D., S. Eddhu and C. Langley, Efficient reconstruction of phylogenetic networks with constrained recombination. In *Proceedings of the Computational Systems Bioinformatics* (CSB'03), pages 363–374, 2003.

[12] Hein, J., Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98(2):185–200, 1990.

[13] Kao, M.-Y., Tree contractions and evolutionary trees. *SIAM Journal on Computing*, 27(6):1592–1616, 1998.

[14] Kao, M.-Y., 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.

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

[16] Meyers, A., R. Yangarber and R. Grishman, Alignment of shared forests for bilingual corpora. In *Proceedings of the $16^{th}$ International Conference on Computational Linguistics* (COLING-96), pages 460–465, 1996.

[17] Nakleh, L., J. Sun, T. Warnow, C. R. Linder, B. M. E. Moret and A. Tholse, Towards the development of computational tools for evaluating phylogenetic reconstruction methods. In *Proceedings of the $8^{th}$ Pacific Symposium on Biocomputing* (PSB 2003), pages 315–326, 2003.

[18] Posada, D. and K. A. Crandall, Intraspecific gene genealogies: trees grafting into networks. *TRENDS in Ecology & Evolution*, 16(1):37–45, 2001.

[19] Setubal, J. C. and J. Meidanis, *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, 1997.

[20] Steel, M. and T. Warnow, Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, 48:77–82, 1993.

[21] Wang, L., K. Zhang and L. Zhang, Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8(1):69–78, 2001.