In: New Topics in Theoretical Computer Science ISBN 978-1-60456-100-5
Editors: Oleg N. Terikhovsky *et al.*, pp. 119-141

Chapter 4

THE MAXIMUM AGREEMENT OF TWO NESTED PHYLOGENETIC NETWORKS*

Jesper Jansson[1,2†] ***and Wing-Kin Sung***[2,3‡]
[1]INRIA Lille - Nord Europe, Equipe SEQUOIA, Villeneuve d'Ascq, France
[2]School of Computing, National University of Singapore, Singapore
[3]Genome Institute of Singapore, Genome, Singapore

Abstract

Given a set $\mathcal{N}$ of phylogenetic networks, the maximum agreement phylogenetic subnetwork problem (MASN) asks for a subnetwork embedded in every $N_i \in \mathcal{N}$ with as many leaves as possible. MASN can be used to identify shared branching structure among phylogenetic networks or to measure their similarity. In this chapter, we prove that the general case of MASN is NP-hard already for two phylogenetic networks (in fact, even if one of the two input networks is a binary tree), but that the problem can be solved efficiently if each of the two input phylogenetic networks exhibits a nested structure. For this purpose, we introduce the concept of a nested phylogenetic network and study some of its underlying fundamental combinatorial properties. We first show that the total number of nodes $|V(N)|$ in any nested phylogenetic network N with n leaves and nesting depth d is $O(n(d+1))$. We then describe a simple algorithm for testing if a given phylogenetic network is nested, and if so, determining its nesting depth in $O(|V(N)| \cdot (d+1))$ time. Next, we present a polynomial-time algorithm for MASN for two nested phylogenetic networks N_1, N_2. Its running time is $O(|V(N_1)| \cdot |V(N_2)| \cdot (d_1+1) \cdot (d_2+1))$, where d_1 and d_2 denote the nesting depths of N_1 and N_2, respectively. In contrast, the previously fastest algorithm for this problem runs in $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$ time, where $f_1 \geq d_1$ and $f_2 \geq d_2$. Finally, we prove that if the nodes are allowed to have outdegree greater than 2 then the problem becomes NP-hard even if restricted to two phylogenetic networks with nesting depth 1.

*A preliminary version of this chapter has appeared in *Proceedings of the 15thAnnual International Symposium on Algorithms and Computation* (ISAAC 2004), volume 3341 of *Lecture Notes in Computer Science*, pages 581–593, Springer-Verlag, 2004.

†E-mail address: Jesper.Jansson@lifl.fr
‡E-mail address: ksung@comp.nus.edu.sg

1. Introduction

Phylogenetic trees are commonly used to describe evolutionary relationships among a set of objects (e.g., biological species, proteins, nucleic acids, viruses, or languages) believed to have been produced by an evolutionary process, and can help scientists to understand the mechanisms of evolution as well as to classify the objects being studied and to organize information [2, 20, 25, 26]. However, evolutionary events such as horizontal gene transfer or hybrid speciation (often referred to as *recombination events*) which suggest convergence between objects cannot be adequately represented in a single tree structure [12, 13, 21, 22, 23, 24, 28]. Phylogenetic *networks* solve this shortcoming by allowing internal nodes to have more than one parent, thereby making it easier for scientists to describe more complex evolutionary relationships. Phylogenetic networks can also be used to visualize several conflicting phylogenetic trees at the same time in order to represent ambiguity [4, 15, 16].

Various methods for constructing and comparing phylogenetic networks have been proposed recently [4, 6, 12, 16, 17, 21, 22, 23, 24, 28]. Phylogenetic network *comparison* has many uses; one application described in [22] is to assess the topological accuracy of different phylogenetic network construction methods[1]. Another application for phylogenetic network comparison is to identify a subnetwork with as many leaves as possible which is contained in all of the networks in a given set (obtained, for example, by employing different phylogenetic network construction methods or by using the same method on alternative data sets) to determine which ancestral relationships are present in all networks. Moreover, the size of such a subnetwork provides a measure of how similar the networks in a given set are. This problem was formalized as a computational problem called *the maximum agreement phylogenetic subnetwork problem* (MASN) and initially studied in [6].

The general case of MASN is NP-hard for three or more phylogenetic networks [6]. Actually, it is NP-hard even for just *two* networks, as we shall prove in Section 4.1.. On the other hand, in the special case of no recombination events at all, MASN for two networks (i.e., rooted, leaf-labeled binary trees) can be solved very efficiently[2]. Fortunately, in nature, recombination events usually do not occur in an unrestricted manner [12, 28]. It is therefore important to establish what structural restrictions on the input networks make the problem efficiently solvable. In this chapter, we investigate the computational complexity of MASN for two phylogenetic networks whose merge paths are *nested*, which is a natural generalization of rooted, leaf-labeled, binary trees and so called galled-trees previously studied in [12, 17, 23, 28] (see below for definitions), and prove that this case can be solved by a polynomial-time algorithm. The decomposition technique for nested phylogenetic networks that we develop here may also be applicable to other computational and combinatorial problems related to phylogenetic network construction and comparison.

[1] To evaluate a construction method $\mathcal{M}$, the following steps are performed a number of times. First, a phylogenetic network N is randomly generated and a sequence is evolved down the edges of N according to some chosen model of evolution, then a phylogenetic network N' for the resulting set of sequences is reconstructed using $\mathcal{M}$, and finally the similarity between N' and N is measured.

[2] See the comments about *the maximum agreement subtree problem* (MAST) in Section 1.3..

1.1. Problem Definition

A *phylogenetic network* 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 a finite set 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. We denote the set of all nodes and the set of leaves in a phylogenetic network N by $V(N)$ and $\Lambda(N)$, respectively.

Given a phylogenetic network N and a set 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 a leaf 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).

Given a set $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$ of phylogenetic networks, an *agreement subnetwork of* $\mathcal{N}$ is a phylogenetic network A such that $\Lambda(A) \subseteq \bigcap_{N_i \in \mathcal{N}} \Lambda(N_i)$ and for every $N_i \in \mathcal{N}$, it holds that A is isomorphic to a graph obtained from $N_i \mid \Lambda(A)$ by deleting zero or more edges and contracting each outgoing edge from a node with resulting outde-

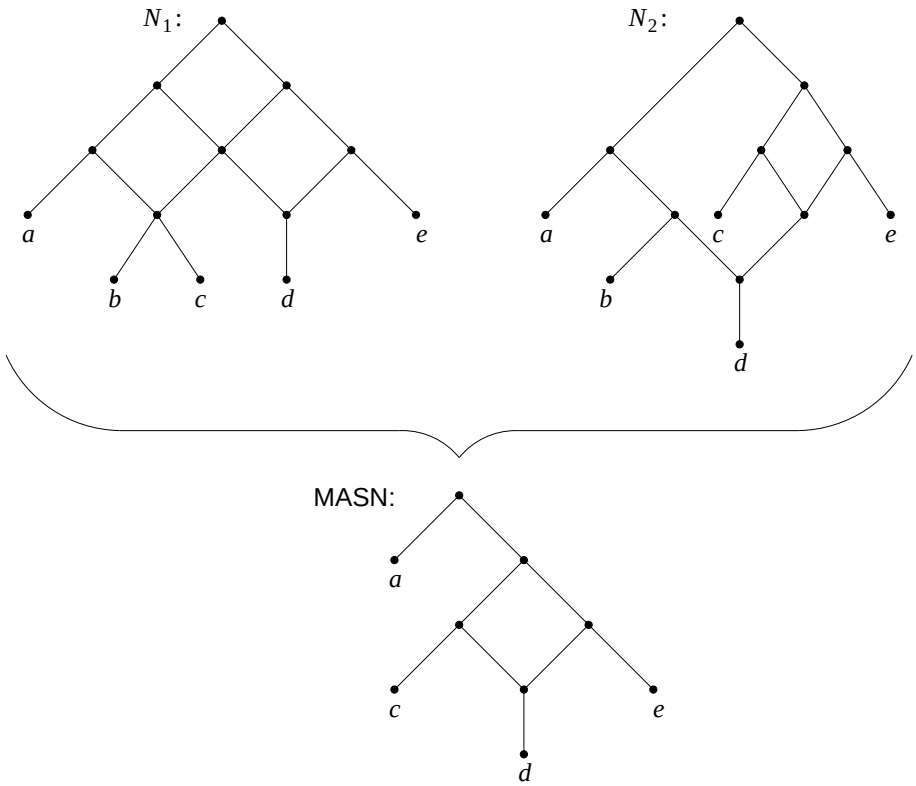

Figure 1. A maximum agreement subnetwork of two given phylogenetic networks N_1 and N_2. Another maximum agreement subnetwork of N_1 and N_2 (not shown here) has leaf set $\{a, b, d, e\}$.

gree 1 and indegree less than 2. 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 set $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$ of phylogenetic networks, find a maximum agreement subnetwork of $\mathcal{N}$. See Figure 1 for an example. A leaf can appear in a maximum agreement subnetwork of $\mathcal{N}$ only if it is present in every network in $\mathcal{N}$, so we assume without loss of generality that $\Lambda(N_1) = \Lambda(N_2) = \ldots = \Lambda(N_k)$ and call this leaf set L. Throughout this chapter, we let n denote the number of different leaves and k the number of input networks, i.e., $n = |L|$ and $k = |\mathcal{N}|$ in the problem definition above.

1.2. Terminology

Let N be a phylogenetic network. Recall that nodes in N with outdegree 0 are called *leaves*. We refer to nodes with indegree 2 as *hybrid nodes*. For any hybrid node h, every ancestor s of h such that h can be reached using two disjoint directed paths starting at the children of s is termed a *split node of* h. If s is a split node of h then any path starting at s and ending at h is called a *merge path of* h, and any path starting at a child of s and ending at a parent of h is called a *clipped merge path of* h.

For any hybrid node h, let $\mathcal{M}(h)$ denote the set of all merge paths of h. We say that N is a *nested phylogenetic network* if for each pair of hybrid nodes h_1, h_2, one of the following three conditions holds: (1) each $P_1 \in \mathcal{M}(h_1)$ and $P_2 \in \mathcal{M}(h_2)$ are internally disjoint paths; (2) for each $P_1 \in \mathcal{M}(h_1)$, there exists a $P_2 \in \mathcal{M}(h_2)$ such that P_1 is a subpath of P_2; or (3) for each $P_2 \in \mathcal{M}(h_2)$, there exists a $P_1 \in \mathcal{M}(h_1)$ such that P_2 is a subpath of P_1. For example, in Figure 1, the phylogenetic network N_2 and the displayed maximum agreement subnetwork are nested, but N_1 is not.

For each node u in a nested phylogenetic network N, define the *nesting depth of* u, $d(u)$, as the number of hybrid nodes in N that have a clipped merge path passing through u. Figure 3 contains an example of a nested phylogenetic network where the nesting depths of some nodes are shown. The *nesting depth of* N, denoted by $d(N)$, is the maximum value of $d(u)$ over all $u \in V(N)$. Observe that $d(N) = 0$ if and only if N is a binary tree. Gusfield *et al.* [12] defined a *galled-tree* (also referred to in the literature as a *gt-network* [23] or a *topology with independent recombination events* [28]) as a phylogenetic network in which all clipped merge paths are disjoint. For a discussion on the biological significance of galled-trees, see [12]. Clearly, $d(N) \leq 1$ if and only if N is a galled-tree. Thus, nested phylogenetic networks naturally extend the notion of rooted, leaf-labeled, binary trees and galled-trees.

Finally, given any phylogenetic network N, 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 the number of nodes it contains which are hybrid nodes in N. 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. To illustrate, N_1 and N_2 in Figure 1 are level-3 and level-2 phylogenetic networks, respectively, and the shown maximum agreement subnetwork of N_1 and N_2 is a level-1 phylogenetic network. If N is a nested phylogenetic network with nesting depth d then $f \geq d$ because any node in N that has nesting depth d must belong to the same biconnected component in $\mathcal{U}(N)$ as at least d different hybrid

nodes. Also, $f = 0$ if and only if $d = 0$, and $f = 1$ if and only if $d = 1$.

1.3. Previous Results

Median-joining, split decomposition (SplitsTree), PYRAMIDS, statistical parsimony (TCS), molecular-variance parsimony (Arlequin), reticulogram (T-REX), and netting are some of the existing general methods for *constructing* phylogenetic networks (see [21] and [24] for a survey). More recently presented methods include Neighbor-Net [4] and the Z-closure method [16]. Algorithms for some reconstruction problems with additional constraints on the networks were given in [5, 12, 17, 23, 28]; in particular, these papers considered problems involving constructing a phylogenetic network with nesting depth 1.

As for *comparing* two given phylogenetic networks, one method based on the Robinson-Foulds (RF) measure for phylogenetic trees was proposed in [22]. MASN was introduced in [6], where it was shown to be NP-hard if restricted to $k = 3$ and an $O(n^2)$-time algorithm for the special case of two level-1 phylogenetic networks (i.e., having nesting depth 1) was presented. [6] also showed that MASN for a level-f_1 phylogenetic network N_1 and a level-f_2 phylogenetic network N_2 can be solved in $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$ time.

MASN extends a well-studied problem known as *the maximum agreement subtree problem* (MAST)[3] (see, e.g., [1, 3, 7, 9, 11, 14, 18, 19, 27] and the numerous references therein) in which the input is a set of distinctly leaf-labeled trees and the goal is to compute a tree embedded in all of the input trees with the maximum possible number of labeled leaves. The fastest known algorithm for MAST for two trees 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 [18]. Note that this is $O(n \log n)$ for two trees with D bounded by a constant and $O(n^{1.5})$ for two trees with unbounded D. MAST is NP-hard for three trees with unbounded degrees [1], and solvable in $O(kn^3 + n^\delta)$ time for $k \geq 3$ trees, where δ is an upper bound on at least one of the input trees' degrees [3, 9] (for $\delta = 2$, even faster algorithms exist [19]). The inapproximability of MAST has been studied in [11] and [14], in terms of how the heights and degrees of the input trees as well as the number of input trees affect the polynomial-time approximability of MAST.

1.4. Our Results and Organization of Chapter

In this chapter, we focus on MASN for two nested phylogenetic networks.

In Section 2., we derive some useful combinatorial properties of nested phylogenetic networks. We first prove that $|V(N)| = O(n(d + 1))$ for any nested phylogenetic network N with n leaves and nesting depth d and then show how to test whether a given phylogenetic network is nested, and if so, determine its nesting depth in $O(|V(N)| \cdot (d + 1))$ time. In Section 3., we present a fast dynamic programming-based algorithm for solving MASN for two nested phylogenetic networks N_1 and N_2 running in $O(|V(N_1)| \cdot |V(N_2)| \cdot (d_1+1) \cdot (d_2+1))$ time, where d_1 and d_2 are the nesting depths of N_1 and N_2, respectively, which generalizes the algorithm from [6]. (The algorithm given in [6] could be applied here

[3] MAST is also known in the literature as *the maximum homeomorphic subtree problem* (MHT).

directly but its running time is $O(|V(N_1)| \cdot |V(N_2)| \cdot 2^{f_1+f_2})$, where $f_1 \geq d_1$ and $f_2 \geq d_2$.) For the special case $d_1 = 1$, $d_2 = 1$, i.e., two galled trees/level-1 networks, the running time of our new algorithm coincides with the running time of $O(n^2)$ of the algorithm in [6]. Next, in Section 4.1., we strengthen the NP-hardness result of [6] by proving that MASN is NP-hard already for *two* phylogenetic networks, even when one of the networks is required to be a binary tree[4]. In Section 4.2., we consider a new variant of MASN in which the definition of a phylogenetic network is relaxed to allow nodes to have outdegree greater than 2 and prove that with this modification, the problem becomes NP-hard even if restricted to two nested phylogenetic networks with nesting depth 1 (i.e., two galled-trees/level-1 networks). Finally, we discuss possible extensions of our techniques in Section 5..

2. Preliminaries

We first investigate some basic properties of nested phylogenetic networks.

Lemma 1. *If N is a nested phylogenetic network then: (1) each split node in N is a split node of exactly one hybrid node, and (2) each hybrid node in N has exactly one split node.*

Proof. Let s be any split node in N and denote the two children of s by c and d. Suppose, for the sake of contradiction, that there exist two hybrid nodes h_1 and h_2 such that s is a split node of both h_1 and h_2. For $i \in \{1, 2\}$, let C_i and D_i be two disjoint clipped merge paths of h_i starting at c and d, respectively, and ending at the two parents of h_i, and let C'_i and D'_i be the corresponding (non-clipped) merge paths. Since the intersection of C'_1 and C'_2 contains c, one must be a subpath of the other by the definition of a nested phylogenetic network, and similarly for D'_1 and D'_2. Now, if C'_1 is a subpath of C'_2 then D'_1 must be a subpath of D'_2 (otherwise, there would exist a directed path from h_1 to h_2 and from h_2 to h_1, contradicting that a phylogenetic network has no cycles), but then C_2 and D_2 are not disjoint because both pass through h_1. Contradiction. The case where C'_2 is a subpath of C'_1 is analogous. (1) follows.

To prove (2), suppose some hybrid node h has two split nodes s_1 and s_2. Denote the parents of h by p and q. For $i \in \{1, 2\}$, let P_i and Q_i be two disjoint clipped merge paths of h starting at the two children of s_i and ending at p and q, respectively, and let P'_i and Q'_i be the corresponding (non-clipped) merge paths. Let h_p be the node in the intersection of P'_1 and P'_2 closest to the root, let h_q be the node in the intersection of Q'_1 and Q'_2 closest to the root, and let s be the lowest common ancestor of s_1 and $_2$. If $s \neq s_1$ and $s \neq s_2$ then s is a split node of three hybrid nodes (h, h_p, and h_q), and if $s = s_1$ or $s = s_2$ then s is a split node of two hybrid nodes (h and either h_p or h_q). In both cases, we have a contradiction with (1). □

Because of Lemma 1, each hybrid node in a nested phylogenetic network corresponds to a unique split node. For any such hybrid node h and split node s, s is called *the split node of* h and h is called *the hybrid node of* s.

[4]The reduction in [1] for proving the NP-hardness of MAST restricted to three trees with unbounded degrees cannot be used directly for MASN with $k = 2$ because it constructs *three* trees and because here we require all nodes to have outdegree at most two. It is interesting to note that MAST for two binary trees is solvable in $O(n \log n)$ time [7, 18].

Lemma 2. *Let h be a hybrid node in a nested phylogenetic network and let s be the split node of h. Then $d(h) = d(s)$.*

Proof. Suppose $d(h) < d(s)$. Then there exists some clipped merge path P containing s but not h. Let P' be the corresponding (non-clipped) merge path. Since s has outdegree 2, P' must contain one of the outgoing edges from s. Let Q be the merge path of h which also uses this edge. Now, P' and Q are not disjoint and one is not a subpath of the other, yet their intersection contains at least two nodes, contradicting the definition of a nested phylogenetic network. The case $d(h) > d(s)$ can be disproved in the same way. □

We now derive an upper bound on the total number of nodes in a nested phylogenetic network. The next two lemmas generalize Lemmas 2 and 3 in [6].

Lemma 3. *If N is a nested phylogenetic network with n leaves and nesting depth d then the number of hybrid nodes in N is at most $(n-1) \cdot d$.*

Proof. Let $T_N(d)$ be the phylogenetic network N. Then, for $i \in \{0, 1, \ldots, d-1\}$, define $T_N(i)$ as the rooted directed graph constructed from $T_N(i+1)$ as follows. For every hybrid node h in $T_N(i+1)$ with $d(h) = i$, remove h's two incoming edges, contract the split node of h and all nodes on the two clipped merge paths of h to a single node s, and add a directed edge from s to h. (Note that the obtained $T_N(i)$ may contain nodes with outdegree greater than 2.) $T_N(0)$ is a tree because every node with indegree 2 in N has indegree 1 in $T_N(0)$ and no contraction increases the indegree of any node. Furthermore, $T_N(0)$ contains n leaves. Thus, the number of internal nodes in $T_N(0)$ with outdegree > 1 is at most $n-1$. Next, observe that at most d split nodes in N correspond to each internal node in $T_N(0)$ with outdegree > 1 and that the number of hybrid nodes in N equals the number of split nodes in N since N is nested. □

Lemma 4. *If N is a phylogenetic network with n leaves and H hybrid nodes then the total number of nodes in N is at most $2(n+H)-1$.*

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 of nodes in N is $|V(N)| = z_{02} + z_{10} + z_{12} + z_{20} + z_{21} + z_{22}$. For every $u \in V(N)$, let $in(u)$ and $out(u)$ denote the number of incoming and outgoing edges incident to u. Since

$$\begin{cases} \sum\limits_{u \in V(N)} in(u) \;=\; z_{02} \cdot 0 \;+\; (z_{10} + z_{12}) \cdot 1 \;+\; (z_{20} + z_{21} + z_{22}) \cdot 2 \\ \sum\limits_{u \in V(N)} out(u) \;=\; (z_{10} + z_{20}) \cdot 0 \;+\; z_{21} \cdot 1 \;+\; (z_{02} + z_{12} + z_{22}) \cdot 2 \end{cases}$$

and $\sum\limits_{u \in V(N)} in(u) = \sum\limits_{u \in V(N)} out(u)$, we have $z_{12} \;=\; z_{10} + 2z_{20} + z_{21} - 2z_{02}$.

Next, $H = z_{20} + z_{21} + z_{22}$, $n \;=\; z_{10} + z_{20}$, and $z_{02} = 1$ give us $z_{12} \;\leq\; n + H - 2$. Hence, $|V(N)| \;\leq\; 1 + n + (n + H - 2) + H \;=\; 2n + 2H - 1$. □

For an example showing that the bounds given above are tight, refer to Figure 2. By combining Lemmas 3 and 4, we get:

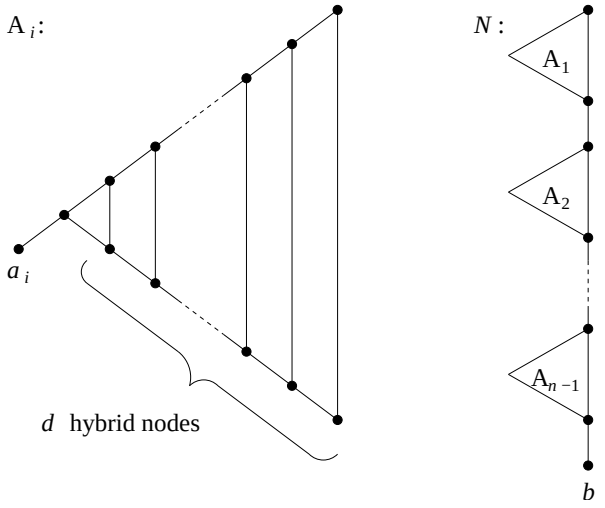

Figure 2. An example of a nested phylogenetic network N with nesting depth d and n leaves for which the upper bounds given in Lemmas 3 and 4 are tight. N (shown on the right) consists of $n-1$ copies of A_i (shown on the left) and is distinctly leaf-labeled by $\{a_1, a_2, \ldots, a_{n-1}, b\}$. The number of hybrid nodes H in N equals $(n-1) \cdot d$ and $|V(N)| = (2d+2) \cdot (n-1) + 1 = 2H + 2n - 1$.

Theorem 5. *If N is a nested phylogenetic network with n leaves and nesting depth d then $|V(N)| \leq 2dn + 2n - 2d - 1$, i.e., $|V(N)| = O(n(d+1))$.*

We also have the following.

Theorem 6. *Let N be a phylogenetic network with n leaves and H hybrid nodes. We can test whether N is nested in $O(|V(N)| \cdot (H+1))$ time; if N is nested, the test takes only $O(|V(N)| \cdot (d(N)+1))$ time and its nesting depth can be determined in the same asymptotic time bound.*

Proof. Use the following method to construct a list $L(u)$ for every $u \in V(N)$ consisting of all hybrid nodes which have a clipped merge path passing through u, plus u itself if u is a hybrid node. Associate an initially empty list $L(u)$ to each $u \in V(N)$, and define $L(\emptyset) = \emptyset$. Visit the nodes of N according to a reverse topological ordering of N. Whenever a non-leaf node u is visited, examine $L(u_L)$ and $L(u_R)$, where u_L and u_R are the children of u (if u only has one child then let u_R equal $\emptyset$). If $L(u_L)$ is empty then let $L(u) := L(u_R)$; else if $L(u_R)$ is empty then let $L(u) := L(u_L)$. Otherwise, check whether $L(u_L)$ equals $L(u_R)$. If no then N is not nested, and the algorithm terminates; if yes then let $L(u) := L(u_L)$ and remove the last element ℓ from $L(u)$ (in this case, u is the split node for the hybrid node ℓ). Finally, if u is a hybrid node then insert u at the end of $L(u)$. Note that a node may be both a split node and a hybrid node. The length of any $L(u)$ can never exceed the number of hybrid nodes in N. Moreover, when the algorithm is finished, if N is a nested phylogenetic network then its nesting depth $d(N)$ equals the maximum length of $L(u)$ over all $u \in V(N)$ since $d(u) = |L(u)|$ for each non-hybrid node u.

The time taken at each node in N is bounded by $O(1 + \max_{u \in V(N)} |L(u)|)$. □

3. An Algorithm for MASN for Two Nested Phylogenetic Networks

In this section, we show how to solve MASN for two nested phylogenetic networks N_1, N_2 with n leaves in $O(|V(N_1)| \cdot |V(N_2)| \cdot (d_1 + 1) \cdot (d_2 + 1))$ time, where d_1 and d_2 are the nesting depths of N_1 and N_2, respectively. We first introduce some additional notation.

Let N be any nested phylogenetic network. From this point onward, 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 let u_L and u_R denote the left and right child of u, respectively, and if u only has one child c then set $u_L = c$ and $u_R = \emptyset$. For every $u \in V(N)$, $N[u]$ 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.

Each $u \in V(N)$ belongs to $d(u)$ different clipped merge paths. Since N is nested, the $d(u)$ different hybrid nodes corresponding to these clipped merge paths have nesting depths $0, 1, \ldots, d(u) - 1$. For $i \in \{1, \ldots, d(u)\}$, we define $h^i(u)$ as the hybrid node h which has a clipped merge path passing through u and which satisfies $d(h) = i - 1$. Next, for $i \in \{1, \ldots, d(u)\}$, let $N^i[u]$ be the subgraph of $N[u]$ where $N[h^i(u)]$ and $h^i(u)$'s incoming edge have been removed, and let $N^0[u]$ be $N[u]$. Define $N^i[u]$ for $i > d(u)$ as $N^0[u]$ if u is not a hybrid node, and as $N[\emptyset]$ if u is a hybrid node. See Figure 3. Intuitively,

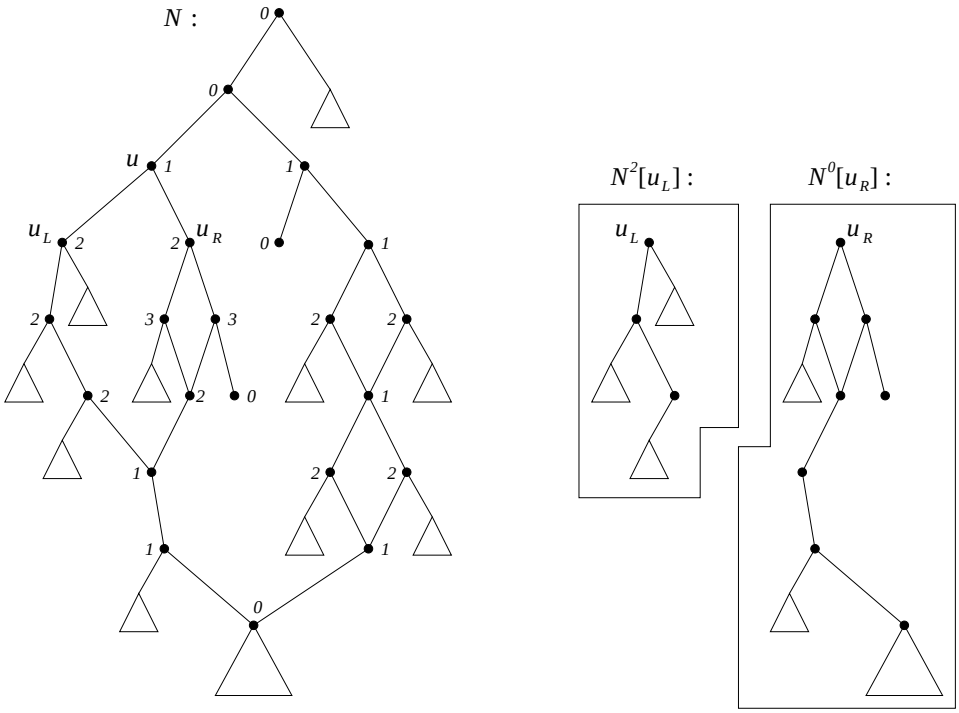

Figure 3. N is a nested phylogenetic network with nesting depth 3 and u is a split node in N. The numbers shown next to the nodes of N are their respective nesting depths. $N^2[u_L]$ and $N^0[u_R]$ are the subgraphs of N displayed on the right.

the parameter i informs us at which descendant hybrid node of u to cut $N[u]$ to obtain $N^i[u]$.

Lemma 7. *For any nested phylogenetic network N, $u \in V(N)$, and $0 \leq j < i \leq d(u)$, it holds that $N^i[u]$ is a proper subgraph of $N^j[u]$.*

Proof. If $j = 0$ then $N^i[u]$ is trivially a proper subgraph of $N^j[u]$ ($= N^0[u] = N[u]$). If $j > 0$, the node $h^j(u)$ is a descendant of $h^i(u)$ since $j < i$, so $N[h^j(u)]$ is a proper subgraph of $N[h^i(u)]$, and therefore $N^i[u]$ is a proper subgraph of $N^j[u]$. □

Lemma 8. *Let N be a nested phylogenetic network. For any $u \in V(N)$ and $i \in \{0, 1, \ldots, d(u)\}$, it holds that: (1) $N^i[u_L]$ and $N^x[u_R]$ are disjoint, and (2) $N^x[u_L]$ and $N^i[u_R]$ are disjoint, where $x = d(u) + 1$ if u is a split node and $x = i$ otherwise.*

Proof. If u is a split node then let h be the hybrid node of u. By Lemma 2, $d(h) = d(u)$. Let c_1 be a child of u with $c_1 \neq h$ and let c_2 be the other child of u, possibly with $c_2 = h$. We have $h^x(c_1) = h^{d(u)+1}(c_1) = h$, which means that $N^x[c_1]$ does not contain any nodes in $N[h]$; hence, $N^x[c_1]$ and $N^0[c_2]$ are disjoint, and Lemma 7 then implies that $N^x[c_1]$ and $N^i[c_2]$ are disjoint. Similarly, $N^i[c_1]$ and $N^x[c_2]$ are disjoint (if $c_2 \neq h$ then $h^x(c_2) = h^{d(u)+1}(c_2) = h$ so $N^x[c_2]$ contains no nodes in $N[h]$ and thus no nodes in $N^i[c_1]$; if $c_2 = h$ then $N^x[c_2] = N^{d(u)+1}[h] = N^{d(h)+1}[h] = N[\emptyset]$).

If u is not a split node then $N[u_L]$ ($= N^0[u_L]$) and $N[u_R]$ ($= N^0[u_R]$) are always disjoint. By Lemma 7, $N^i[u_L]$ and $N^i[u_R]$ are disjoint. □

For any two phylogenetic networks N_1, N_2, define $Masn(N_1, N_2)$ as the number of leaves in a maximum agreement subnetwork. If N_1 or N_2 is an empty network then $Masn(N_1, N_2)$ is equal to 0. Otherwise, $Masn(N_1, N_2)$ for two nested phylogenetic networks can be expressed recursively using the following lemma which is a generalization of the main lemma in [27] for MAST. In the $Match$ case, when trying to match two subnetworks $N_1^i[u_L]$ and $N_1^x[u_R]$ to two subnetworks $N_2^k[v_L]$ and $N_2^y[v_R]$, Lemma 8 ensures that the set of nodes in the intersection of $V(N_1[u_L])$ and $V(N_1[u_R])$ is matched to only one of $N_2^k[v_L]$ and $N_2^y[v_R]$, and vice versa.

Lemma 9. *Let N_1 and N_2 be two nested phylogenetic networks. For every $(u, v) \in V(N_1) \times V(N_2)$ and $0 \leq i \leq d(u)$, $0 \leq k \leq d(v)$,*

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

where

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

and

$$\begin{aligned} Match(N_1^i[u], N_2^k[v]) = \max\{&Masn(N_1^i[u_L],\ N_2^k[v_L]) + Masn(N_1^x[u_R],\ N_2^y[v_R]), \\ &Masn(N_1^i[u_L],\ N_2^y[v_L]) + Masn(N_1^x[u_R],\ N_2^k[v_R]), \\ &Masn(N_1^i[u_L],\ N_2^k[v_R]) + Masn(N_1^x[u_R],\ N_2^y[v_L]), \\ &Masn(N_1^i[u_L],\ N_2^y[v_R]) + Masn(N_1^x[u_R],\ N_2^k[v_L]), \end{aligned}$$

$$Masn(N_1^x[u_L],\, N_2^k[v_L]) \,+\, Masn(N_1^i[u_R],\, N_2^y[v_R]),$$
$$Masn(N_1^x[u_L],\, N_2^y[v_L]) \,+\, Masn(N_1^i[u_R],\, N_2^k[v_R]),$$
$$Masn(N_1^x[u_L],\, N_2^k[v_R]) \,+\, Masn(N_1^i[u_R],\, N_2^y[v_L]),$$
$$Masn(N_1^x[u_L],\, N_2^y[v_R]) \,+\, Masn(N_1^i[u_R],\, N_2^k[v_L])\},$$

$$\textit{where } x = \begin{cases} d(u)+1, & \text{if } u \text{ is a split node} \\ i, & \text{otherwise} \end{cases}$$
$$y = \begin{cases} d(v)+1, & \text{if } v \text{ is a split node} \\ k, & \text{otherwise} \end{cases}$$

Proof. [Generalization of [27]] If at least one of u and v is a leaf ℓ then the size of a maximum agreement subnetwork of $N_1^i[u]$ and $N_2^k[v]$ is either 0 or 1, depending on whether or not ℓ occurs in the other subnetwork, i.e., $Masn$ is equal to $|\Lambda(N_1^i[u]) \cap \Lambda(N_2^k[v])|$.

If neither of u and v are leaves then let A be any maximum agreement subnetwork of $N_1^i[u]$ and $N_2^k[v]$ which is a tree (such an A must exist because for any agreement subnetwork B which is not a tree, if one parent edge of each hybrid node in B is deleted and edge contractions are performed, we get an agreement subnetwork A with $\Lambda(A) = \Lambda(B)$ which is a tree). Write $M = \Lambda(A)$ so that $|M| = Masn(N_1^i[u], N_2^k[v])$. Let a_1 and a_2 be the lowest common ancestor in $N_1^i[u]$ and $N_2^k[v]$, respectively, of the leaves in M. There are two main cases:

1. $a_1 \neq u$ or $a_2 \neq v$ (the $Diag$ case).
   Here, A is also a maximum agreement subnetwork of each pair of networks $(N_1^i[x], N_2^k[y])$ where x belongs to any path from u to a_1 and y belongs to any path from v to a_2. Hence, $Masn(N_1^i[u], N_2^k[v])$ is equal to $Masn(N_1^i[w_1], N_2^k[w_2])$ for some $(w_1, w_2) \in \{(u, v_L), (u, v_R), (u_L, v), (u_R, v)\}$.

2. $a_1 = u$ and $a_2 = v$ (the $Match$ case).
   The elements in M are descendants of both of u's children and also of both of v's children. Let A_a and A_b be the two subtrees of A rooted at the children of the root of A.

   By Lemma 8, $N_1^i[u_L]$ and $N_1^x[u_R]$ are disjoint; furthermore, every $v \in V(N_1^i[u]) \setminus \{u\}$ belongs to exactly one of $N_1^i[u_L]$ and $N_1^x[u_R]$. The same holds for $N_1^x[u_L]$ and $N_1^i[u_R]$ (observe that if u is not a split node then $x = i$ and these two cases coincide), and there are no other ways to divide $N_1^i[u]$ into two disjoint subnetworks rooted at u_L and u_R. Similarly, $N_2^k[v]$ can be divided into two disjoint subnetworks rooted at v_L and v_R in at most two ways. A_a is therefore a maximum agreement subnetwork of $N_1^{p_1}[u_a]$ and $N_2^{p_2}[v_a]$, and A_b is a maximum agreement subnetwork of $N_1^{q_1}[u_b]$ and $N_2^{q_2}[v_b]$ for some $u_a, u_b \in \{u_L, u_R\}$ with $u_a \neq u_b$ and some $v_a, v_b \in \{v_L, v_R\}$ with $v_a \neq v_b$, and where $p_1 = i$ and $q_1 = x$, or $p_1 = x$ and $q_1 = i$, and where $p_2 = k$ and $q_2 = y$, or $p_2 = y$ and $q_2 = k$. Now, $Masn(N_1^i[u], N_2^k[v]) = |M| = |\Lambda(A)| = |\Lambda(A_a)| + |\Lambda(A_b)|$ is given by one of the eight cases in the equation for $Match$.

Finally, note that in the $Diag$ case, the value of $Match$ is at most $|M|$, and in the $Match$ case, the value of $Diag$ is at most $|M|$. Taking the maximum of $Diag$ and $Match$ thus gives us the size of a maximum agreement subnetwork. □

Now, given two nested phylogenetic networks N_1 and N_2, we can use Lemma 9 to compute $Masn(N_1^i[u], N_2^k[v])$ for all $0 \leq i \leq d(u)$ and $0 \leq k \leq d(v)$ by applying 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 ordered according to a reverse topological ordering of N_i. The resulting algorithm (Algorithm *Nested*$Masn$) is listed in Figure 4.

Algorithm *Nested*$Masn$
Input: Two nested phylogenetic networks N_1 and N_2.
Output: The number of leaves in a maximum agreement subnetwork of $\{N_1, N_2\}$.
1 Compute and store $d(u)$ and $h^i(u)$ for all $u \in V(N_1) \cup V(N_2)$, $i \in \{1, \ldots, d(u)\}$.
2 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 a reverse topological ordering of N_i.
3 **for** each $(u, v) \in V(N_1) \times V(N_2)$ in increasing order in $\mathcal{O}$ **do**
Compute $Masn(N_1^i[u], N_2^k[v])$ for all $0 \leq i \leq d(u)$, $0 \leq k \leq d(v)$ by using the expression in Lemma 9.
endfor
4 **return** $Masn(N_1^0[r_1], N_2^0[r_2])$, where r_i is the root of N_i for $i \in \{1, 2\}$.
End *Nested*$Masn$

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

Lemma 10. *Nested*$Masn$ *runs in* $O(|V(N_1)| \cdot |V(N_2)| \cdot (d(N_1) + 1) \cdot (d(N_2) + 1))$ *time.*

Proof. In Step 1 of Algorithm *Nested*$Masn$, we may compute $d(u)$ and $h^i(u)$ for all $u \in V(N)$, $i \in \{1, \ldots, d(u)\}$ for a given nested phylogenetic network N in a way similar to the algorithm in the proof of Theorem 6 by traversing the nodes of N in bottom-up order. (For every leaf u, $d(u) = 0$; when a non-leaf node u is reached, compute $h^i(u)$ for all valid i by using $d(u_L)$, $d(u_R)$, $h^i(u_L)$, and $h^i(u_R)$ and checking if any of u_L and u_R is a hybrid node, and then assign $d(u)$.) This takes $O(|V(N)| \cdot (d(N) + 1))$ time.

Next, the algorithm evaluates $O(|V(N_1)| \cdot |V(N_2)|)$ pairs of nodes. For any such pair (u, v), if neither u nor v is a leaf then it takes constant time to compute each one of the $O((d(N_1) + 1) \cdot (d(N_2) + 1))$ different $Masn(N_1^i[u], N_2^k[v])$-values from previously computed values. If u is a leaf then the value of each $|\Lambda(N_1^i[u]) \cap \Lambda(N_2^k[v])|$ can be obtained in constant time as follows. Associate 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 $O((|V(N_1)| + |V(N_2)|) \cdot n)$ time by traversing each of N_1 and N_2 according to a reverse topological ordering). Then to determine whether $u \in \Lambda(N_2^0[v])$, check if bit u in $L(v)$ equals 1; for $k \geq 1$, the condition $u \in \Lambda(N_2^k[v])$ is equivalent to $u \in \Lambda(N_2^0[v])$ and $u \notin \Lambda(N_2^0[h^k(v)])$. The case where v is a leaf is analogous. □

Algorithm *Nested*$Masn$ 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 attained as it is computed, e.g., by saving pointers. To construct an actual maximum agreement subnetwork from such a set L', we may

use a standard traceback technique to obtain a tree with leaf set L' which is an agreement subnetwork. This yields:

Theorem 11. *Given two nested phylogenetic networks N_1 and N_2 with nesting depths d_1 and d_2, respectively, a maximum agreement subnetwork can be computed in $O(|V(N_1)| \cdot |V(N_2)| \cdot (d_1 + 1) \cdot (d_2 + 1))$ time.*

4. New NP-hardness Results

Below, we first show that MASN is NP-hard already for $k = 2$. We then show that if our definition of a phylogenetic network is relaxed so that the outdegrees of the nodes are unbounded, then the problem becomes NP-hard even if restricted to two nested phylogenetic networks with nesting depth 1.

4.1. MASN with $k = 2$ is NP-hard

To prove the NP-hardness of MASN for every fixed $k \geq 2$, we provide a polynomial-time reduction from the following problem.

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 NP-complete (see, e.g., [10]). Given an arbitrary instance of 3DM, construct an instance of MASN with two phylogenetic networks N_1 and N_2 with a leaf set L as described next. (In fact, N_1 will be a leaf-labeled binary tree.) The elements of M are encoded in subtrees called S_{x_i,z_k} in N_1 and in subtrees called U_{y_j} in N_2. The purpose of the subtrees named A_{x_i}, B_{x_i,z_k}, and W_{z_k} is to make sure that for any two triples e and f in M, a maximum agreement subnetwork of N_1 and N_2 can contain both of the two leaves representing e and f if and only if e and f differ in all coordinates.

Let the leaf set L equal $M \cup A \cup B$, where A is a set of $q^6 \cdot (q + 2)$ elements not in M and B is a set of q^6 elements not in M or A. Let $A_{x_0}, A_{x_1}, \ldots, A_{x_q}, A_{x_{q+1}}$ be $q + 2$ binary trees with q^6 leaves each, distinctly labeled by A. For every $(x_i, z_k) \in X \times Z$, let B_{x_i,z_k} be a binary tree with q^4 leaves, distinctly labeled by B.

For every $(x_i, z_k) \in X \times Z$, define: (1) M_{x_i,z_k} as the subset of M containing all triples of the form (x_i, y, z_k) where $y \in Y$; and (2) S_{x_i,z_k} to be a tree obtained from a binary caterpillar tree with $|M_{x_i,z_k}| + 1$ leaves distinctly labeled by M_{x_i,z_k} and where one of the bottommost leaves has been replaced by the root of B_{x_i,z_k}. See Figure 5. For every $y_j \in Y$, define: (1) M_{y_j} as the subset of M containing all triples of the form (x, y_j, z) where $x \in X$ and $z \in Z$; and (2) U_{y_j} to be a binary caterpillar tree with $|M_{y_j}| + q$ leaves in which the $|M_{y_j}|$ leaves closest to the root are distinctly labeled by M_{y_j} and the rest are unlabeled nodes referred to as v_{y_j,z_k} for $1 \leq k \leq q$. Finally, for every $z_k \in Z$, define W_{z_k} to be a tree

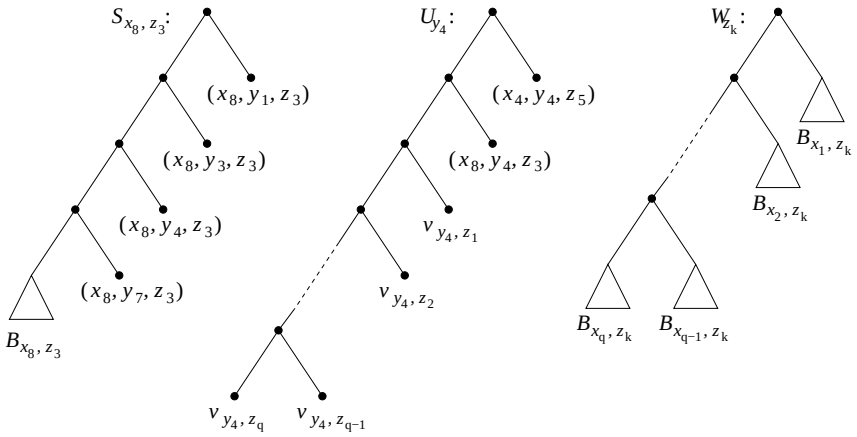

Figure 5. Assume $M_{x_8,z_3} = \{(x_8, y_1, z_3), (x_8, y_3, z_3), (x_8, y_4, z_3), (x_8, y_7, z_3)\}$ and $M_{y_4} = \{(x_4, y_4, z_5), (x_8, y_4, z_3)\}$. S_{x_8,z_3} and U_{y_4} are shown on the left and in the center, respectively. The structure of each W_{z_k} is shown on the right.

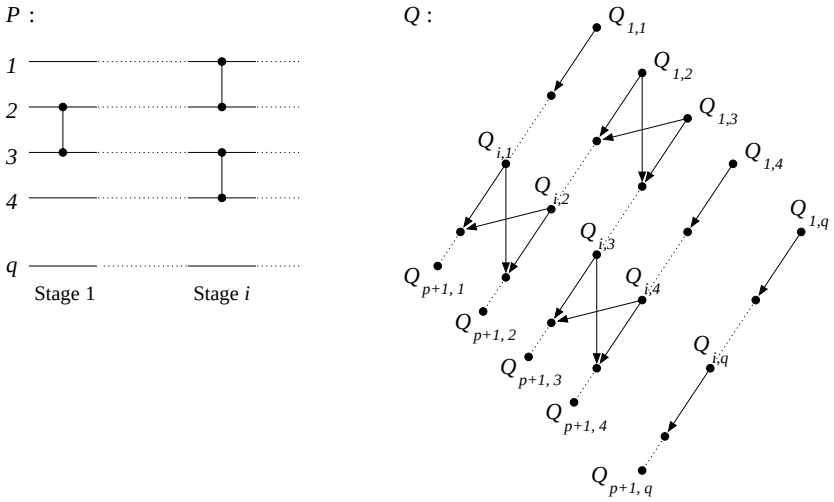

Figure 6. The sorting network P on the left yields a directed acyclic graph Q.

obtained from the binary caterpillar tree with q leaves by replacing the leaves with the roots of $B_{x_1,z_k}, \ldots, B_{x_q,z_k}$.

Next, let P be any sorting network (see, e.g., [8]) for q elements with a polynomial number p of comparator stages. Build a directed acyclic graph Q from P with $(p+1) \cdot q$ nodes $\{Q_{i,j} \mid 1 \leq i \leq p{+}1,\ 1 \leq j \leq q\}$ such that there is a directed edge $(Q_{i,j},\ Q_{i+1,j})$ 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 stage i in P for $1 \leq i \leq p$, as illustrated in Figure 6. Furthermore, construct q directed paths $\{G_1, \ldots, G_q\}$ where each $G_k = (G_{1,k}, \ldots, G_{q,k})$.

Let N_1 be a phylogenetic network obtained by attaching to a directed path $(m_1, m_2, \ldots, m_{q^2+q+2})$, in order of non-decreasing distance from m_1, the roots of A_{x_0},

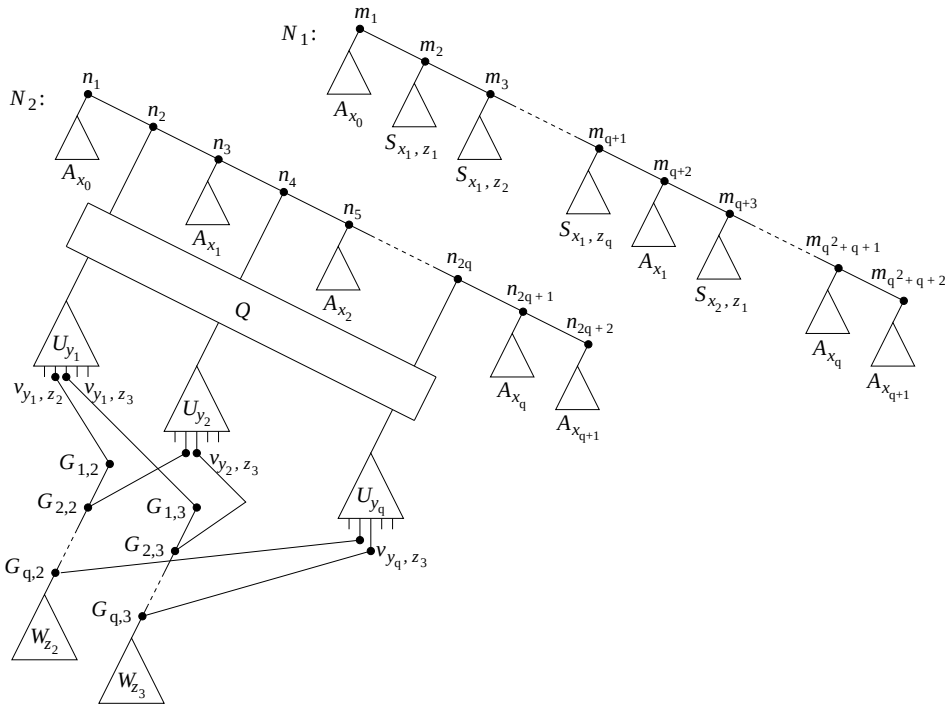

Figure 7. The phylogenetic networks N_1 and N_2.

S_{x_1,z_1}, S_{x_1,z_2}, …, S_{x_1,z_q}, A_{x_1}, S_{x_2,z_1}, …, S_{x_q,z_q}, A_{x_q}, and $A_{x_{q+1}}$, and letting m_1 be the root of N_1. (Note that N_1 is actually a binary tree.) See Figure 7. The phylogenetic network N_2 is obtained by first attaching to a directed path $(n_1, n_2, \ldots, n_{2q+2})$, in order of non-decreasing distance from n_1, the root of A_{x_0}, the node $Q_{1,1}$, the root of A_{x_1}, the node $Q_{1,2}$, the root of A_{x_2}, …, the root of A_{x_q}, and the root of $A_{x_{q+1}}$, and letting n_1 be the root of N_2. Then, for $j \in \{1, \ldots, q\}$, let $Q_{p+1,j}$ coincide with the root of U_{y_j}, and for every $1 \le j \le q$ and $1 \le k \le q$ add a directed edge $(v_{y_j,z_k}, G_{j,k})$. Next, for every $1 \le k \le q$ add a directed edge from $G_{q,k}$ to the root of W_{z_k}. Again, see Figure 7. Finally, for every node in N_1 and N_2 having indegree 1 and outdegree 1, contract its outgoing edge.

Lemma 12. *If M has a matching of size q then there exists an agreement subnetwork of (N_1, N_2) with $q^7 + 2q^6 + q^5 + q$ leaves.*

Proof. Suppose M has a matching M' of size q. For every $(x_i, z_k) \in X \times Z$, denote by V_{x_i,z_k} the set of all leaves in B_{x_i,z_k}. Let $C = M' \cup \bigcup_{(x_i,y_j,z_k)\in M'} V_{x_i,z_k}$ and let T be $N_1 \,|\, (A \cup C)$. For each $x_i \in X$, there is precisely one triple (x_i, y_j, z_k) in M', so the path in T from the root of S_{x_i,z_k} to the root of B_{x_i,z_k} has one leaf (x_i, y_j, z_k) attached to it. Now consider the structure of $N_2 \,|\, (A \cup C)$. Since P is a sorting network, there are q disjoint paths in Q from $(Q_{1,\pi(1)}, Q_{1,\pi(2)}, \ldots, Q_{1,\pi(q)})$ to $(Q_{p+1,1}, Q_{p+1,2}, \ldots, Q_{p+1,q})$ for any given permutation π of $\{1, 2, \ldots, q\}$; in particular, this holds for the permutation π defined by the relation $\pi(j) = i$ for all $(x_i, y_j, z_k) \in M'$. Thus, for every (x_i, y_j, z_k) in M', there exists a path in N_2 from node n_{2i} to the root of B_{x_i,z_k} (passing through the root of U_{y_j} and the nodes v_{y_j,z_k} and $G_{q,k}$) along which the leaf (x_i, y_j, z_k) is attached. This implies

that T is a subgraph of $N_2 \,|\, (A \cup C)$, i.e., T is an agreement subnetwork of (N_1, N_2) with $|A| + q \cdot (1 + q^4) = q^7 + 2q^6 + q^5 + q$ leaves. □

Lemma 13. *If there exists an agreement subnetwork of (N_1, N_2) with $q^7 + 2q^6 + q^5 + q$ leaves then M has a matching of size q.*

Proof. Suppose there exists an agreement subnetwork T' with a leaf set $L' \subseteq L$ such that $|L'| = q^7 + 2q^6 + q^5 + q$. Write $M' = L' \cap M$, $A' = L' \cap A$, and $B' = L' \cap B$. First observe that the number of elements in L' is strictly greater than the number of elements in $L \setminus \{a \mid a \text{ is a leaf of } A_{x_0}\}$, so at least one leaf from A_{x_0} must be included in L' by the pigeonhole principle. Hence, the root of T' corresponds to the roots of N_1 and N_2. Similarly, at least one leaf ℓ_i from A_{x_i} for every $x_i \in X$ and at least one leaf ℓ_{q+1} from $A_{x_{q+1}}$ must belong to L'. Also by the pigeonhole principle, a total of at least $|L'| - |M| - |A| \geq q^5 + q - q^3$ leaves from B must be included in L', and these leaves must in fact belong to at least q different subtrees of the form B_{x_i,z_k} (this is because $q-1$ different subtrees of the form B_{x_i,z_k} can only contain $(q-1) \cdot q^4$ leaves and $(q-1) \cdot q^4 < q^5 + q - q^3$). However, L' cannot contain leaves from both $B_{x_{i_1},z_{k_1}}$ and $B_{x_{i_2},z_{k_2}}$ if $i_1 \neq i_2$ and $k_1 = k_2$ (if b_1 and b_2 are two such leaves then they appear in different S_{x_i,z_k} in N_1 but in the same W_{z_k} in N_2, so, e.g., $N_1 \,|\, \{b_1, b_2, \ell_{q+1}\}$ and $N_2 \,|\, \{b_1, b_2, \ell_{q+1}\}$ differ, which contradicts that $\{b_1, b_2, \ell_{q+1}\}$ are leaves in T'), or if $i_1 = i_2$ and $k_1 \neq k_2$ (if b_1 and b_2 were two such leaves then $N_1 \,|\, \{b_1, b_2, \ell_{i_1-1}, \ell_{i_1}\}$ and $N_2 \,|\, \{b_1, b_2, \ell_{i_1-1}, \ell_{i_1}\}$ would differ); thus B' consists of leaves from at most q (and hence, precisely q by the above) different subtrees of the form B_{x_i,z_k}, and we have $|B'| \leq q \cdot q^4$, yielding $|M'| = |L'| - |A'| - |B'| \geq |L'| - |A| - q^5 = q$.

We now show that 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', i.e., M' is a matching of M. Using the same argument as above, if L' contains a leaf in $B_{x_{i_1},z_{k_1}}$ then L' cannot contain any triple $(x_{i_2}, y_{j_2}, z_{k_2})$ with $i_1 \neq i_2$ and $k_1 = k_2$, or with $i_1 = i_2$ and $k_1 \neq k_2$. Then for any $(x_{i_1}, y_{j_1}, z_{k_1}) \in L'$, L' must also contain a leaf from $B_{x_{i_1},z_{k_1}}$ since leaves from q different subtrees of the form B_{x_i,z_k} must be included in L', so e and f cannot both belong to L' if $i_1 \neq i_2$ and $k_1 = k_2$, or if $i_1 = i_2$ and $k_1 \neq k_2$. Next, if $i_1 \neq i_2$, $j_1 = j_2$, and $k_1 \neq k_2$ then $N_1 \,|\, \{e, f, \ell_{q+1}\}$ and $N_2 \,|\, \{e, f, \ell_{q+1}\}$ differ, implying that L' cannot contain both e and f. Finally, if $i_1 = i_2$, $j_1 \neq j_2$, and $k_1 = k_2$ and $e, f \in L'$ then the roots of $U_{y_{j_1}}$ and $U_{y_{j_2}}$ in N_2 both have to correspond to nodes located in the same subtree $S_{x_{i_1},z_{k_1}}$ in N_1 because e and f belong to $S_{x_{i_1},z_{k_1}}$, and then there are strictly less than $q-1$ available U_{y_j}-roots for the remaining $q-1$ subtrees of the form S_{x_i,z_k} with leaves in L', which is a contradiction. □

From the above, we obtain:

Theorem 14. *MASN is NP-hard even if restricted to $k = 2$, and even if one of the two input networks is a binary tree.*

4.2. MASN with Unrestricted Outdegrees is NP-hard

Here, we prove that MASN for two nested phylogenetic networks with nesting depth 1 (i.e., two galled trees/level-1 networks) is NP-hard if the nodes are allowed to have unrestricted

outdegree. We give a polynomial-time reduction from the problem 3SAT, which is known to be NP-complete (see, e.g., [10]).

Three-Satisfiability (3SAT)

Instance: A set $U = \{u_1, ..., u_p\}$ of Boolean variables and a collection $C = \{c^1, ..., c^q\}$ of disjunctive clauses over U, each containing exactly 3 literals.

Question: Is there a truth assignment for U that makes every clause in C true?

For every $u_i \in U$, let $J(u_i)$ be the set $\{j : u_i \text{ occurs in clause } c^j\}$. Without loss of generality, assume that $|J(u_i)| \geq 2$. Let $J(u_i)_k$ be the kth smallest integer in $J(u_i)$ so that $J(u_i)_1 \leq J(u_i)_2 \leq ... \leq J(u_i)_{|J(u_i)|}$. Now, given an instance of 3SAT, construct an instance of MASN with two nested phylogenetic networks $\mathcal{N} = \{N_1, N_2\}$ (where the outdegrees of the nodes are unrestricted) having a leaf set L as follows.

For each $u_i \in U$, define a set of new elements $V(u_i) = \{v_i^j, \overline{v}_i^j, w_i^j, \overline{w}_i^j : j \in J(u_i)\}$. Similarly, for each $c^j \in C$, define a set of six new elements $D(c^j) = \{d^j[1], d^j[2], d^j[3], e^j[1], e^j[2], e^j[3]\}$. Let $L = \bigcup_{u_i \in U} V(u_i) \cup \bigcup_{c^j \in C} D(c^j)$. Note that for each $c^j \in C$, there are exactly 18 elements with the symbol j in their exponent, so $|L| = 18q$.

For any nonempty $L_1, L_2, L_3 \subseteq L$, define $S(L_1; L_2; L_3)$ to be a nested phylogenetic network with nesting depth 1 having a single hybrid node h where: (1) $|L_1|$ leaves distinctly labeled by L_1 are attached to a path of length $|L_1|$ starting at the left child of the root and ending at h; (2) $|L_2|$ leaves distinctly labeled by L_2 are attached to a path of length $|L_2|$ starting at the right child of the root and ending at h; and (3) h is the parent of $|L_3|$ leaves distinctly labeled by L_3. See Figure 8 for an example. For any $c^j \in C$, let $T(c^j)$ be a tree whose root has one child with three children x_1, x_2, x_3, and where for $k \in \{1, 2, 3\}$, x_k is the parent of two leaves labeled by $d^j[k]$ and $e^j[k]$.

We build the nested phylogenetic network N_1 as follows. First, for every $u_i \in U$, construct for all $k \in \{1, ..., |J(u_i)|\}$ the networks $S(\{v_i^{J(u_i)_k}\}; \{\overline{v}_i^{J(u_i)_{k+1}}\}; \{w_i^{J(u_i)_k}, \overline{w}_i^{J(u_i)_{k+1}}\})$, where $J(u_i)_{|J(u_i)|+1} \equiv J(u_i)_1$, and let all their roots coincide with the root of N_1. Then, construct $T(c^1), \ldots, T(c^q)$ and make all of their roots also coincide with the root of N_1. See Figure 9.

Next, for every $u_i \in U$ and all $j \in J(u_i)$, if u_i is the kth literal in c^j then define $R(u_i, j)$ as $S(\{v_i^j\}; \{\overline{v}_i^j, d^j[k], e^j[k]\}; \{w_i^j, \overline{w}_i^j\})$; otherwise, if $\overline{u}_i$ is the kth literal in c^j then let $R(u_i, j)$ be $S(\{v_i^j, d^j[k], e^j[k]\}; \{\overline{v}_i^j\}; \{w_i^j, \overline{w}_i^j\})$. Let N_2 be the nested phylogenetic network whose root node coincides with the roots of all $R(u_i, j)$, where $u_i \in U$ and $j \in J(u_i)$. See Figure 10.

Lemma 15. *If U has a truth assignment that makes every clause in C true then there exists an agreement subnetwork of (N_1, N_2) with $10q$ leaves.*

Proof. Suppose U has a truth assignment $A : U \to \{\text{true, false}\}$ that satisfies all clauses in C. For each $u_i \in U$, construct a set $L(u_i)$ as follows. If $A(u_i) = \text{true}$ then let $L(u_i) = \{v_i^j, w_i^j : j \in J(u_i)\}$ and define $\ell(u_i^j) = \overline{v}_i^j$ for all $j \in J(u_i)$, and if $A(u_i) = \text{false}$ then $L(u_i) = \{\overline{v}_i^j, \overline{w}_i^j : j \in J(u_i)\}$ and $\ell(u_i^j) = v_i^j$ for all $j \in J(u_i)$. Next, for every $j \in J(u_i)$, if u_i is the variable with the lowest index which makes c^j true then add $d^j[k]$

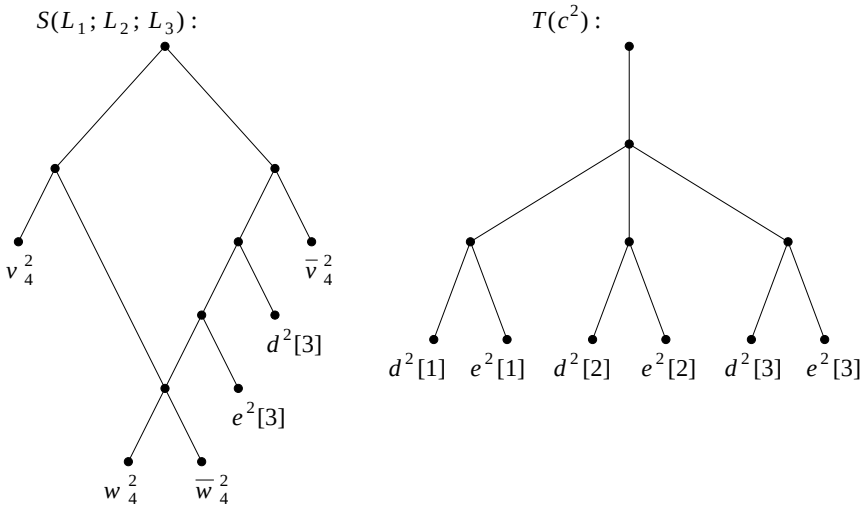

Figure 8. $S(L_1; L_2; L_3)$ with $L_1 = \{v_4^2\}$, $L_2 = \{\overline{v}_4^2, d^2[3], e^2[3]\}$, and $L_3 = \{w_4^2, \overline{w}_4^2\}$ is shown on the left. $T(c^2)$ is shown on the right.

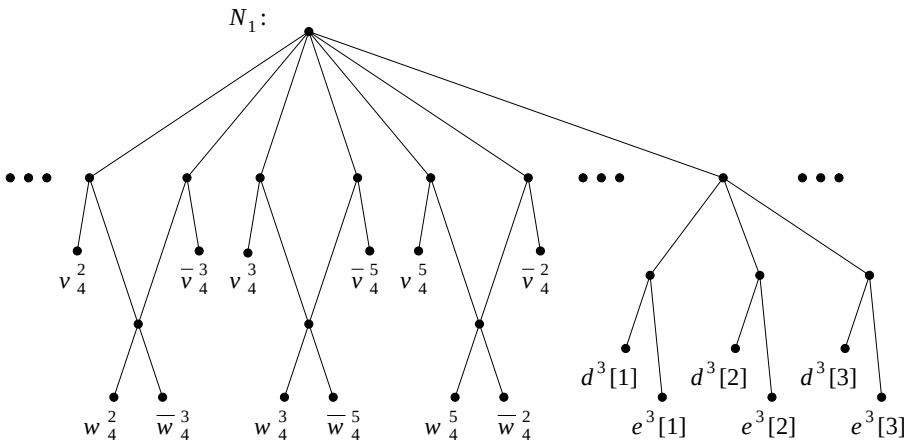

Figure 9. The phylogenetic network N_1. Assume that variable u_4 occurs in c^2, c^3, and c^5 in the given instance of 3SAT. Then the portion of N_1 that corresponds to u_4 has the structure shown above. Also shown is $T(c^3)$, the part corresponding to c^3.

and $e^j[k]$ to $L(u_i)$, where u_i is the kth variable in c^j; otherwise, if u_i is not the variable with the lowest index which makes c^j true then add $\ell(u_i^j)$ to $L(u_i)$. Let $L' = \bigcup_{u_i \in U} L(u_i)$.

Let T be the following tree, distinctly leaf-labeled by L'. For each $c^j \in C$, the root r of T has 6 children. Let u_i be the variable (the one with the lowest index, if there exists more than one) which makes c^j true when assigned the value $A(u_i)$, and denote the other two variables in c^j by u_x and u_y. Two of the children of r corresponding to c^j are leaves labeled by $\ell(u_x^j)$ and $\ell(u_y^j)$. Another one of the children of r is a node with two children labeled by $d^j[k]$ and $e^j[k]$, where u_i is the kth variable in c^j. The remaining three children of r corresponding to c^j are nodes with two children each, labeled by either v_z^j and w_z^j (if

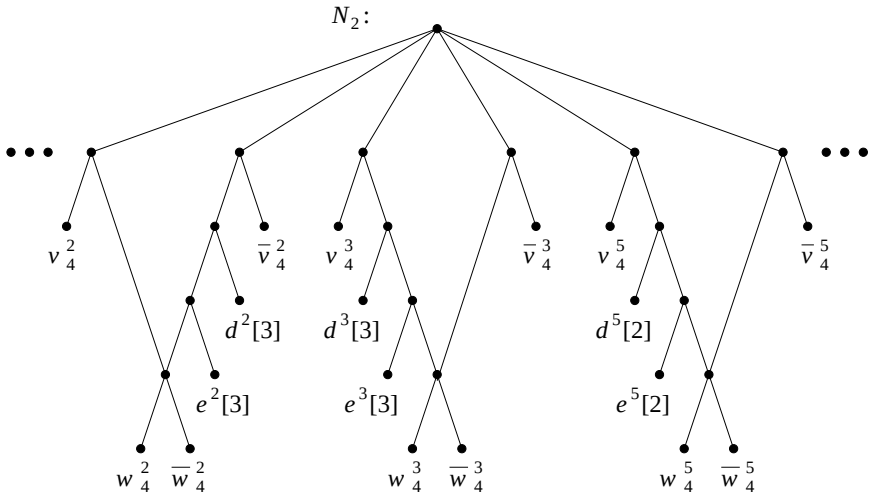

Figure 10. The phylogenetic network N_2. If $c^2 = (\ldots \vee \ldots \vee u_4)$, $c^3 = (\ldots \vee \ldots \vee \overline{u}_4)$, and $c^5 = (\ldots \vee \overline{u}_4 \vee \ldots)$ then the part corresponding to u_4 looks as above.

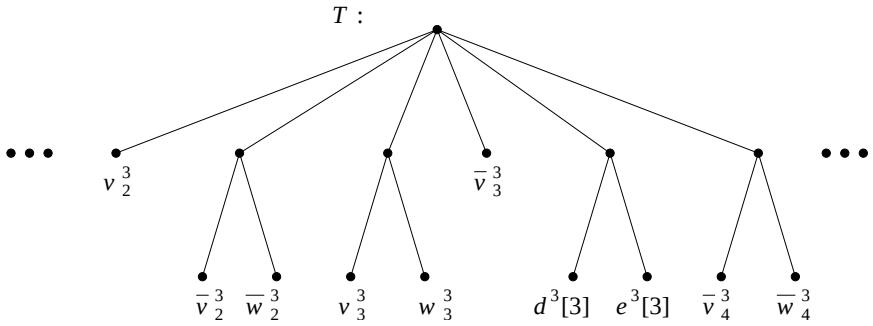

Figure 11. Assume $c^3 = (u_2 \vee \overline{u}_3 \vee \overline{u}_4)$, $A(u_2) =$ false, $A(u_3) =$ true, and $A(u_4) =$ false. The part of T corresponding to c^3 is displayed. $d^3[3]$ and $e^3[3]$ belong to T since c^3 is satisfied because of u_4, i.e., the third variable in c^3.

$A(u_z) =$ true) or $\overline{v}_z^j$ and $\overline{w}_z^j$ (if $A(u_z) =$ false) for $z \in \{i, x, y\}$. See Figure 11. T has 10 leaves for each $c^j \in C$, and thus $10q$ leaves in total. It is easily verified that T is a subgraph of $N_1 \mid L'$ and also a subgraph of $N_2 \mid L'$, and hence an agreement subnetwork of (N_1, N_2). □

Lemma 16. *If there exists an agreement subnetwork of* (N_1, N_2) *with* $10q$ *leaves then* U *has a truth assignment that makes every clause in* C *true.*

Proof. Suppose there exists an agreement subnetwork T' with a leaf set $L' \subseteq L$ such that $|L'| = 10q$. For each $c^j \in C$, denote the set of all leaves in L' with the symbol j in their exponent as L_j. By the structure of N_1 and N_2, at most one of w_i^j and $\overline{w}_i^j$ for every $u_i \in U$ and $j \in J(u_i)$ may appear in L'. Also, for each $c^j \in C$, elements from at most one of the three pairs $(d^j[1], e^j[1])$, $(d^j[2], e^j[2])$, and $(d^j[3], e^j[3])$ can belong to L'. Furthermore, if some $d^j[k]$ or $e^j[k]$ is in L' then either v_i^j (if the kth literal of c^j is of the form u_i) or $\overline{v}_i^j$ (if

the kth literal of c^j is of the form $\overline{u}_i$) cannot be in L'. Hence, for each $c^j \in C$, at most 10 leaves with the symbol j in their exponent belong to L'. Denote the three variables which are included in c^j by u_x, u_y, and u_z. Since $|L'| = 10q$ it follows that $|L_j| = 10$ and L_j must consist of: (1) $d^j[k]$ and $e^j[k]$ for some $k \in \{1, 2, 3\}$; (2) five of the six elements in $\{v_x^j, \overline{v}_x^j, v_y^j, \overline{v}_y^j, v_z^j, \overline{v}_z^j\}$; and (3) for each $t \in \{x, y, z\}$, either w_t^j or $\overline{w}_t^j$.

Next, for each $u_i \in U$, if $w_i^j \in L'$ for some $j \in J(u_i)$ then $w_i^k \in L'$ for every $k \in J(u_i)$ (and analogously if $\overline{w}_i^j \in L'$). This follows because $w_i^{J(u_i)_k} \in L'$ implies that $\overline{w}_i^{J(u_i)_{k+1}} \notin L'$ (they belong to the same S in N_1 but different S in N_2), where $J(u_i)_{|J(u_i)|+1} \equiv J(u_i)_1$, and by (3) above, $w_i^{J(u_i)_{k+1}} \in L'$. We can now define a truth assignment $A' : U \to \{\text{true}, \text{false}\}$ as follows. For each $u_i \in U$, if $w_i^k \in L'$ for every $k \in J(u_i)$ then set $A'(u_i) = \text{true}$, and if $\overline{w}_i^k \in L'$ for every $k \in J(u_i)$ then set $A'(u_i) = \text{false}$.

Finally, we show that each $c^j \in C$ is satisfied by A'. By the above, $d^j[k]$ and $e^j[k]$ for some $k \in \{1, 2, 3\}$ belong to L_j. Let ℓ be the kth literal in c^j. If ℓ is of the form u_i then $\overline{v}_i^j$ lies on the same side as $d^j[k]$ and $e^j[k]$ in $R(u_i, j)$ in N_2, and hence $\overline{v}_i^j \notin L'$ and $\overline{w}_i^j \notin L'$, giving us $w_i^j \in L'$ and $A'(u_i) = \text{true}$. Otherwise, ℓ is of the form $\overline{u}_i$ and then v_i^j lies on the same side as $d^j[k]$ and $e^j[k]$ in $R(u_i, j)$ in N_2, and hence $v_i^j \notin L'$ and $w_i^j \notin L'$, giving us $\overline{w}_i^j \in L'$ and $A'(u_i) = \text{false}$. In both cases, c^j is satisfied. □

Lemmas 15 and 16 give us the next theorem.

Theorem 17. *If the restriction on the outdegrees of the nodes is removed then MASN is NP-hard even for two nested phylogenetic networks with nesting depth* 1 *(i.e., two galled-trees/level-*1 *networks).*

5. Conclusion

MASN with $k = 2$ is NP-hard (as proved in Section 4.1.), but efficiently solvable for some special types of phylogenetic networks. For example, if N_1 and N_2 are trees then the problem can be solved in $O(n \log n)$ time [7, 18], if N_1 and N_2 are level-1 phylogenetic networks (i.e., "galled-trees", using the terminology of [12]) then the problem is solvable in $O(n^2)$ time [6], and more generally, if N_1 and N_2 are level-f phylogenetic networks, where $f = O(\log(|V(N_1)| + |V(N_2)|))$, then the $O(|V(N_1)| \cdot |V(N_2)| \cdot 4^f)$-time algorithm in [6] runs in time which is polynomial in the input size. In this chapter, we have demonstrated that even when the parameter f is unrestricted, the problem can be solved in polynomial time if N_1 and N_2 are nested.

Does MASN for other types of structurally restricted phylogenetic networks admit efficient algorithms? In particular, is it possible to extend our method in Section 3. to two networks in which every hybrid node has exactly one split node? An example of such a network is shown in Figure 12. We would also like to know if MASN can be solved efficiently for an even more complex structure which we call a *planar phylogenetic network*, defined as follows: for any positive integers a, b, let $M(a, b)$ be a rooted, directed graph with node set $\{M_{i,j} \mid 1 \leq i \leq a,\ 1 \leq j \leq b\}$ such that there is one directed edge from $M_{i,j}$ to $M_{i-1,j}$ for every $2 \leq i \leq a$ and $1 \leq j \leq b$, and one directed edge from $M_{i,j}$ to $M_{i,j-1}$ for every

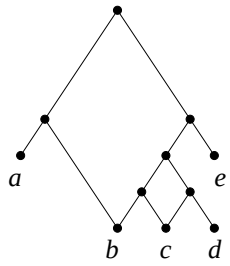

Figure 12. This phylogenetic network is not nested although every hybrid node has exactly one split node, and every split node has exactly one hybrid node. (Thus, the converse of Lemma 1 is not true.)

$1 \leq i \leq a$ and $2 \leq j \leq b$; we say that the network N is an (a, b)-*planar phylogenetic network* if each biconnected component in $\mathcal{U}(N)$ is isomorphic to a subgraph of $M(a, b)$.

We believe MASN for more than two nested phylogenetic networks can be solved in polynomial time when $k = O(1)$. It would also be interesting to investigate if any other computational problems which are hard to solve for unrestricted phylogenetic networks but known to be solvable in polynomial time for galled-trees can be solved efficiently for nested phylogenetic networks with unrestricted nesting depths. One example of such a problem might be *the perfect phylogenetic network with recombination problem*, which is NP-hard for unrestricted networks [28] but solvable in polynomial time for galled-trees [12].

The final open question is: can the running time of our algorithm for two nested phylogenetic networks be improved, e.g., by applying sparsification techniques?

References

[1] 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.

[2] M. Bonet, C. Phillips, T. Warnow, and S. Yooseph. Constructing evolutionary trees in the presence of polymorphic characters. *SIAM Journal on Computing*, **29**(1):103–131, 1999.

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

[4] D. Bryant and V. Moulton. Neighbor-Net: An agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution*, **21**(2):255–265, 2004.

[5] H.-L. Chan, J. Jansson, T.-W. Lam, and S.-M. Yiu. Reconstructing an ultrametric galled phylogenetic network from a distance matrix. *Journal of Bioinformatics and Computational Biology*, **4**(4):807–832, 2006.

[6] C. Choy, J. Jansson, K. Sadakane, and W.-K. Sung. Computing the maximum agreement of phylogenetic networks. *Theoretical Computer Science*, **335**(1):93–107, 2005.

[7] 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.

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

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

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

[11] L. Gąsieniec, J. Jansson, A. Lingas, and A. Östlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, **3**(2–3):183–197, 1999.

[12] D. Gusfield, S. Eddhu, and C. Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *Journal of Bioinformatics and Computational Biology*, **2**(1):173–213, 2004.

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

[14] J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, **71**:153–169, 1996.

[15] B. Holland and V. Moulton. Consensus networks: A method for visualising incompatibilities in collections of trees. In *Proceedings of the 3rd Workshop on Algorithms in Bioinformatics* (WABI 2003), volume 2812 of *LNCS*, pages 165–176. Springer, 2003.

[16] D. H. Huson, T. Dezulian, T. Klöpper, and M. Steel. Phylogenetic super-networks from partial trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**(4):151–158, 2004.

[17] J. Jansson, N. B. Nguyen, and W.-K. Sung. Algorithms for combining rooted triplets into a galled phylogenetic network. *SIAM Journal on Computing*, **35**(5):1098–1121, 2006.

[18] 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.

[19] C.-M. Lee, L.-J. Hung, M.-S. Chang, C.-B. Shen, and C.-Y. Tang. An improved algorithm for the maximum agreement subtree problem. *Information Processing Letters*, **94**(5):211–216, 2005.

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

[21] C. R. Linder, B. M. E. Moret, L. Nakhleh, and T. Warnow. Network (reticulate) evolution: Biology, models, and algorithms. Tutorial presented at *the 9^{th}Pacific Symposium on Biocomputing* (PSB 2004), 2004.

[22] L. Nakhleh, 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.

[23] L. Nakhleh, T. Warnow, C. R. Linder, and K. St. John. Reconstructing reticulate evolution in species – theory and practice. *Journal of Computational Biology*, **12**(6):796–811, 2005.

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

[25] C. Semple and M. Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2003.

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

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

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