Reconstructing an Ultrametric Galled Phylogenetic Network from a Distance Matrix

Ho-Leung Chan, Jesper Jansson, Tak-Wah Lam, and Siu-Ming Yiu

Department of Computer Science, The University of Hong Kong,
Pokfulam Road, Hong Kong
{hlchan, jjansson, twlam, smyiu}@cs.hku.hk

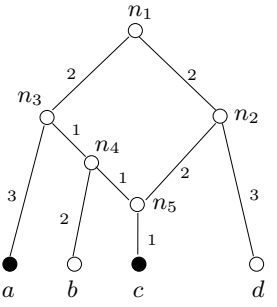
Abstract. Given a distance matrix M that specifies the pairwise evolutionary distances between n species, the phylogenetic *tree* reconstruction problem asks for an edge-weighted phylogenetic tree that satisfies M, if one exists. We study some extensions of this problem to rooted phylogenetic *networks*. Our main result is an $O(n^2 \log n)$-time algorithm for determining whether there is an ultrametric galled network that satisfies M, and if so, constructing one. In fact, if such an ultrametric galled network exists, our algorithm is guaranteed to construct one containing the minimum possible number of nodes with more than one parent (*hybrid* nodes). We also prove that finding a largest possible submatrix M' of M such that there exists an ultrametric galled network that satisfies M' is NP-hard. Furthermore, we show that given an incomplete distance matrix (i.e., where some matrix entries are missing), it is also NP-hard to determine whether there exists an ultrametric galled network which satisfies it.

1 Introduction

A phylogenetic network is a generalization of a phylogenetic tree which can be used to describe the evolutionary history of a set of species that is non-treelike, for example, due to recombination events such as hybrid speciation or horizontal gene transfer [8, 14, 15, 17] or to represent several conflicting phylogenetic trees at once in order to identify parts where the trees disagree [2, 10].

To develop efficient methods for inferring phylogenetic networks is an important topic in computational biology. In particular, one promising category of methods which includes methods such as Neighbor-Net [2] and several others (see [15] for a survey) is known as *distance-based*. Here, the input consists of a (symmetric and non-negative) distance matrix which specifies the pairwise evolutionary distances between the species. To infer a phylogenetic *tree* from such a matrix is a well-studied problem [3, 5, 6, 16, 18], the basic objective being to construct an edge-weighted phylogenetic tree such that for any two species, the length of the path between them in the tree equals the corresponding entry

J. Jędrzejowicz and A. Szepietowski (Eds.): MFCS 2005, LNCS 3618, pp. 224–235, 2005.



M

	a	b	c	d
a	0	6	6	10
b		0	10	10
c			0	6
d				0

Fig. 1. The (galled and ultrametric) phylogenetic network on the left satisfies the distance matrix M on the right. There are two evolutionary paths (a, n_3, n_4, n_5, c) and $(a, n_3, n_1, n_2, n_5, c)$ with lengths 6 and 10, respectively, connecting a and c. The entry $M(a, c)$ corresponds to the first path. Note that there does not exist any phylogenetic tree that satisfies M.

in the matrix. Note that in a phylogenetic tree, the path between two specified leaves is always unique. On the other hand, due to recombination events, for any two species in a phylogenetic network, there can be more than one path connecting them with different path lengths. The entry in the input matrix may correspond to one of these paths only. Hence, in some cases, there may exist a phylogenetic network that satisfies the given distance matrix (see the definition below) while no such phylogenetic tree exists. See Figure 1 for an example. In this paper, we consider some natural extensions of the distance-based variant of the phylogenetic tree reconstruction problem to phylogenetic networks and present a new algorithm.

Problem Definitions: A *rooted phylogenetic network* for a set S of species is a rooted, connected, directed acyclic graph such that: (1) exactly one node (the *root*) has indegree 0 and all other nodes have indegree 1 or 2; (2) any node with indegree 2 (called a *hybrid node*) has outdegree 1 and all other nodes have outdegree 0 or 2; and (3) each node with outdegree 0 (a *leaf*) is labeled with a distinct species from S. A rooted phylogenetic network is called a *galled phylogenetic network*, or *galled network* for short[1], if all cycles in the underlying undirected graph (i.e., where edge orientations are ignored) are node-disjoint. For example, the phylogenetic network in Figure 1 and the network N_1 in Figure 2 are galled networks. From here on, we only consider phylogenetic networks that are *edge-weighted*, i.e., where each edge has a positive length. In analogy with the standard usage of the term "ultrametric" for phylogenetic trees, we say that a galled network is *ultrametric* if every directed path from the root to a leaf has the same length.

[1] Galled networks are also known in the literature as *topologies with independent recombination events* [17], *galled-trees* [8], *gt-networks* [14], and *level-1 phylogenetic networks* [13].

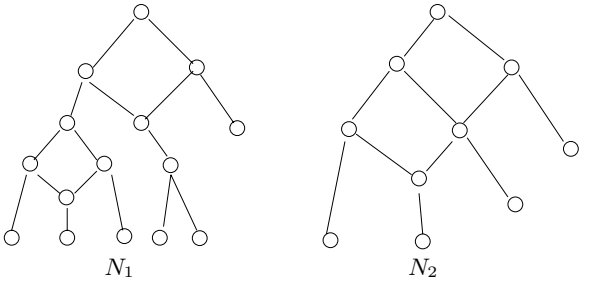


Fig. 2. N_1 is a galled network, while N_2 is not (The leaf labels are omitted for clarity.)

For any rooted phylogenetic network N, an *evolutionary path* between two leaves a and b is a simple path which goes up (i.e., moving in a child-to-parent direction) from a to a common ancestor u of a and b, and then down (i.e., moving in a parent-to-child direction) from u to b. Observe that even if N is galled and ultrametric, there can be more than one evolutionary path between a and b, and moreover, these paths may have different lengths (again, see Figure 1). However, in an ultrametric galled network, there can exist at most two different evolutionary path lengths between each pair of leaves.

A *distance matrix* for a set S of n species is a symmetric, non-negative $(n \times n)$-matrix M such that $M(a, a) = 0$ for every $a \in S$. Intuitively, for each $a, b \in S$, $M(a, b)$ contains the measured evolutionary distance between a and b. A rooted phylogenetic network N for S *satisfies* M if, for every $a, b \in S$, it holds that N contains an evolutionary path between a and b of length equal to $M(a, b)$. In this case, we also say that M is *satisfied by* N. We are now ready to define the problem which is the main focus of this paper.

Problem Statement: Given a distance matrix M for a set S of n species, return an ultrametric galled network for S satisfying M, if one exists; otherwise, return *fail*.

Motivation: The rationale behind the way we define the problem is as follows. There are a number of methods to estimate the evolutionary distance between two species. One common approach is to align the DNA sequences for some related genes from the species. The alignment score usually provides a reasonable estimation on the evolutionary distance between the species. However, if recombination events had occurred, there may exist more than one common ancestor (at different evolutionary distances) for a pair of species. Thus, depending on which common ancestor the selected genes were inherited from, the measured evolutionary distance may reflect only one of the possible evolutionary paths. Therefore, for any two species in the phylogenetic network, we only require one of their evolutionary paths to satisfy the matrix entry.

If there are no restrictions on the topological structure of the constructed phylogenetic network, it may not make sense from a biological point of view.

We therefore concentrate on galled networks, a very useful class of rooted phylogenetic networks which despite their simple structure are powerful enough to describe evolutionary history when the frequency of recombination events is moderate or when most of the recombination events have occurred recently [8]. See [8] for a discussion on the importance of galled networks. Also, the biological meaning of the ultrametric assumption is that the species have evolved according to a constant rate; see, e.g., [3, 5, 6, 18] and the references therein for justification of this assumption.

Finally, there may be more than one ultrametric galled network that satisfies an input matrix. From the biological point of view, it is more reasonable if we could find the simplest explanation that is consistent with the observed distances. So, although recombination events (corresponding to hybrid nodes) may occur, a more reasonable network is the one with the minimum number of hybrid nodes.

Our Contributions: Our main result in this paper is an exact $O(n^2 \log n)$-time algorithm to construct an ultrametric galled network (if one exists) that satisfies a given distance matrix M. When a solution exists, our algorithm always outputs one having as few hybrid nodes as possible. On the other hand, we prove that finding a largest possible submatrix M' of M such that there exists an ultrametric galled network that satisfies M' is an NP-hard problem. We also show that given an incomplete distance matrix (i.e., where some matrix entries are missing), it is NP-hard to determine whether there exists an ultrametric galled network which satisfies it.

Related Works: In the context of reconstructing a phylogenetic network from distance data, the most related work is the *Neighbor-Net* method, developed by Bryant and Moulton [2], which outputs a planar, unrooted phylogenetic network from a given distance matrix. Neighbor-Net is based on the well-known Neighbor-Joining method for trees [16]. Earlier proposed distance-based methods for reconstructing phylogenetic networks include [4] and others described in [15]. However, all of these approaches are heuristics-based and there is no guarantee that the output is a phylogenetic network that satisfies the given matrix exactly, even when a galled network exists. Also, Neighbor-Net runs in $O(n^3)$ time, which is slower than the method we present here.

Some other models of computation for reconstructing phylogenetic networks (i.e., assuming other types of input) are reviewed in [15]. Recently, in addition to distance-based methods, researchers have also studied *character-based* [7, 8, 17] and *supertree-based* [9, 10, 11, 12, 14] methods for inferring phylogenetic networks.

To reconstruct a phylogenetic *tree* with n species consistent with a given distance matrix (if one exists) can easily be done in $O(n^2)$ time (see [5, 6]). Note however, that when an exact solution does not exist, obtaining a tree that is as "close" as possible to the matrix has been shown to be NP-hard on several closeness metrics [3, 5, 18].

2 Preliminaries

Let N be a galled network. In the rest of this paper, we will use the following terminology. A node h in N is a *hybrid node* if the indegree of h is equal to 2. Let s be an ancestor of h such that there are two edge-disjoint paths from s to h. Then s is called *the split node of* h. In a galled network, each split node is a split node of exactly one hybrid node, and each hybrid node has exactly one split node (see Lemma 1 in [13]). The two paths from s to h are *the merge paths of* h, and they form a *galled loop* rooted at s. The galled loop rooted at s is *skew* if one of its two merge paths consists of a single edge from s to h; otherwise, it is *non-skew*. Nodes other than h and s on the merge paths of h are called *side nodes*, and a node is called a *tree node* if it is not on any galled loop. For any node u in N, the *subnetwork* rooted at u is the minimal subgraph of N including all nodes and directed edges reachable from u, and is denoted by N_u. Finally, N_u is a *side network* if the parent of u belongs to a merge path P in N but u itself is not on P.

In a galled network, the smallest possible galled loop is skew and consists of exactly three nodes (a split node, a hybrid node, and a side node). A simple induction can show that a galled network with n leaves contains at most $3n-3$ internal nodes. This property is useful to our algorithm.

For any internal node u of an ultrametric galled network N, every directed path from u to a leaf under u has the same length. We call this length the *height* of u and denote it by $height(u)$. For any leaf a, $height(a) = 0$. Note that the length of any edge (a, b) can be calculated from $height(a)$ and $height(b)$. Thus, to find a network for M, we only need to determine the heights of all internal nodes and the parent-child relations between nodes.

3 Framework of the Algorithm

Given an $n \times n$ distance matrix M for a set S of n species, we first analyze some properties for the ultrametric galled network satisfying M. For simplicity, we say a *network* to refer to an ultrametric galled network. M is *satisfiable* if there exists a network satisfying it. For any $S' \subseteq S$, if a network N for S' satisfies the submatrix of M induced by the species in S', we say that N satisfies S'.

Consider any two species a and b in S. To satisfy M, the network contains an evolutionary path between a and b with length equal to $M(a, b)$. We notice that this path starts from a, goes up to a common ancestor of height $M(a, b)/2$, and then goes down to b. Let D_S be the maximum distance between two species in S as specified by M. If M is satisfiable, then there is a network satisfying M whose root has height $D_S/2$.

Also, we have the following observation about the internal nodes of N.

Observation 1. Assume that M can be satisfied by a network N. For any node u that is a tree node or a split node, let N_u be the subnetwork rooted at u, and let S_u be the set of species in N_u.

- For any two species $a, b \in S_u$, $M(a,b) = 2 \times height(v)$ for some internal node v in N_u, and hence $M(a,b) \leq 2 \times height(u)$.
- For any species $a \in S_u$ and $c \in S - S_u$, $M(a,c) > 2 \times height(u)$.

Observation 1 motivates us to consider the following definition.

Definition 1. For any set of species $S' \subseteq S$, S' is called a *cluster* if there exists a value x such that for any two species $a, b \in S'$, $M(a,b) \leq x$ and for any species $a \in S'$ and $c \in S - S'$, $M(a,c) > x$.

S itself is the biggest cluster. Note that clusters are nested, i.e., two clusters are always either disjoint or one is a subset of the other. Observation 1 states that every tree node and split node in N corresponds to a cluster. In fact, the reverse is also true.

Lemma 1. Assume that M can be satisfied by some network. Then there exists one such network N such that, for every cluster $S' \subseteq S$, N has a tree node or a split node u such that all species in S' are in the subnetwork N_u, and no species in $S - S'$ are in N_u.

To prove Lemma 1, we let N be any network satisfying M. If N does not satisfy Lemma 1, we can modify it to obtain a network satisfying Lemma 1. Details will be given in the full paper.

We call a network satisfying Lemma 1 a *well-structured* network, which has a very nice property as follows. Consider any $S' \subseteq S$ that is a cluster. Let $S_1, S_2, \ldots, S_t$ be all the maximal clusters which are proper subsets of S'. We call $S_1, S_2, \ldots, S_t$ the *side clusters* of S'. Note that $S' = S_1 \cup S_2 \cup \ldots \cup S_t$.

Lemma 2. Let S' be a cluster with side clusters $S_1, \ldots, S_t$. Let N be any well-structured network satisfying S' (w.r.t. the submatrix of M induced by S'). N consists of a root node u, with the networks satisfying $S_1, \ldots, S_t$ attached to u, or attached to a galled loop rooted at u.

Proof. As N is well-structured, for each side cluster S_i, there is a tree node or a split node v whose subnetwork contains exactly all species in S_i. We notice that on the path from v to the root u, there is no tree node or split node other than u or v (otherwise, let v' be that intermediate node; the species under the subnetwork rooted at v' form a cluster S'' and $S_i \subset S'' \subset S'$, meaning that S_i is not a side cluster of S'). Thus, v is directly attached to u or a galled loop rooted at u. It means that N is formed by attaching the networks for $S_1, \ldots, S_t$ to u, or to a galled loop rooted at u. □

The Algorithm

Lemma 2 states that we can construct a network for a cluster by connecting the networks for its side clusters. Thus, our algorithm takes a bottom-up approach, which continuously identifies subsets of S that are clusters, starting from smaller ones to bigger ones. It maintains an invariant that as soon as a cluster S' is found, a subnetwork satisfying S' is constructed. For the base case, a set containing only a single species is a cluster, and the corresponding network is a single leaf for this species. Since S is the biggest cluster, the algorithm will eventually find a network satisfying S.

To ease the finding of clusters, our algorithm constructs a graph G as follows. Initially, G has n isolated nodes, each representing a species in S. Edges which represent the distance among the species are added in rounds, where two nodes u, v will be connected by an edge of length $M(u, v)$. In the i-th round, all edges with the i-th shortest length are added. Suppose that after the i-th round, a connected component of G becomes a clique. Then the species inside this connected component form a cluster for which a network is built immediately. The algorithm is shown below. Details of Step 2c will be given in the next section.

Algorithm 1. GalledNet

Step 1. Sort the entries in M and let $m_1 < m_2 < ... < m_r$ be the distinct positive values in M. If $r > 3n - 3$, return failure.

Step 2. Build the networks while constructing a graph G. Initially, G contains n isolated nodes representing the n species. For $i = 1, 2, \ldots, r$,

a. Add all edges of length m_i to G.
b. Identify all connected components that become a new clique.
c. For each new clique, let $S' \subseteq S$ be the corresponding cluster. Run the procedure ConnectingSideClusters (shown in the next section) which constructs a network satisfying S', if S' is satisfiable. This is done by creating a new root u, and attaching the networks for the side clusters of S' to u, or to a galled loop rooted at u.

Note that any galled network for n species can contain at most $3n-3$ internal nodes, and the length of any evolutionary path is $2 \times height(u)$ for some internal node u. Thus, if there are more than $3n - 3$ distinct positive values in M, no network can satisfy M.

We analyse the running time of GalledNet as follows. Step 1 takes $O(n^2 \log n)$ time. Step 2a takes $O(n^2)$ time over the whole algorithm. With some straightforward bookkeeping (which takes $O(1)$ time for each edge added), Step 2b can be done in $O(n)$ time in each iteration and $O(n^2)$ time in total. We will show in the next section that Step 2c, which calls ConnectingSideClusters, takes totally $O(n^2)$ time. Thus, the whole algorithm takes $O(n^2 \log n)$ time.

Theorem 1. *Algorithm GalledNet runs in* $O(n^2 \log n)$ *time.*

4 Attaching Side Clusters to a Galled Loop

This section explains how Step 2c of GalledNet is performed. Let S be a satisfiable cluster with side clusters $S_1, S_2, \cdots, S_t$. Suppose that we have constructed side networks for these side clusters. Below we overload S_i to also denote the corresponding side network. To build a network for S, we need to determine how these side clusters (more precisely, their side networks) are attached to a new root or to a galled loop, and compute the height of the new root and nodes on the loop.

We skip the simple case of $t = 2$ and we consider only the general case that $t \geq 3$, i.e., S has three or more side clusters. We need to build a galled loop to accommodate the corresponding side networks S_i's. We focus on the network N that satisfies S and we show that the structure of N can be determined from the relations between the side clusters. Recall that N has a galled loop at the top. Let S_h be the side cluster attached to the hybrid node. Let LEFT(S_h) be the group of side clusters attached to the side nodes on the left merge path. Define RIGHT(S_h) similarly. The following lemma tells how to identify the side clusters in the two groups. For simplicity, we say a species a is in LEFT(S_h) (resp. RIGHT(S_h)) if a belongs to some side cluster in LEFT(S_h) (resp. RIGHT(S_h)).

Lemma 3 (Partitioning the side clusters to the two merge paths). Let D_S be the maximum distance between two species in S. **(i)** For any two species a, b in LEFT(S_h), $M(a, b) < D_S$; similarly, for any two species a, b in RIGHT(S_h), $M(a, b) < D_S$; and **(ii)** for any species a in LEFT(S_h) and c in RIGHT(S_h), $M(a, c) = D_S$.

Assume that LEFT(S_h) contains ℓ side clusters and their side networks are attached to side nodes $v_1, v_2, \cdots, v_\ell$ on the left merge path of N, where v_i is the i-th node next the hybrid node. Let r be the root. Denote the side cluster (as well as the side network) attached to v_i as $S(v_i)$. That is, LEFT(S_h) = $\{S(v_i) \mid 1 \leq i \leq \ell\}$.

The following lemmas provide some structural characteristics of each side network $S(v_i)$, which allow us to identify each of them easily. For each side cluster S' of S, let *inter_dist*(S') denote the minimum distance $M(x, y)$ between a species x in S' and a species y in $S - S'$.

Lemma 4 (Identifying the order of side clusters). (i) *inter_dist*($S(v_1)$) $\leq$ *inter_dist*($S(v_2)$), and *inter_dist*($S(v_2)$) $<$ *inter_dist*($S(v_3)$) $< \cdots <$ *inter_dist* ($S(v_\ell)$); **(ii)** *height*(v_i) = *inter_dist*($S(v_i)$)/2 for $i = 2, \cdots, \ell$.

Lemma 4(i) allows us to identify which side cluster in LEFT(S_h) is attached to each v_i, except when *inter_dist*($S(v_1)$) = *inter_dist*($S(v_2)$). In this case, we

exploit the relationship with S_h to distinguish the side clusters attached to v_1 and v_2. Note that a species x in $S(v_2)$ and a species y in S_h are connected by two evolutionary paths, with the root r and v_2 as the highest node, respectively. Since N satisfies S, the distance of x and y (i.e., $M(x, y)$) must equal the length of either path, i.e., $2 \times \mathit{height}(r)$ or $2 \times \mathit{height}(v_2)$. The latter value is strictly less than $2 \times \mathit{height}(r) = D_S$.

Lemma 5 (Resolving ambiguity). (i) If $\mathit{inter\_dist}(S(v_1)) = \mathit{inter\_dist}(S(v_2))$, then $S(v_2)$, but not $S(v_1)$, contains a species x whose distance to some species y in S_h (i.e., $M(x, y)$) is less than D_S, and $\mathit{height}(v_1)$ can be any value in the range $(\mathit{height}(S_h), \mathit{height}(v_2))$. **(ii)** Otherwise, $\mathit{height}(v_1) = \mathit{inter\_dist}(S(v_1))/2$.

The above lemmas explain how the side clusters are attached to the merge paths, once the side cluster under the hybrid node is known. The following lemma shows that we can in fact find the side cluster attached to the hybrid node easily.

Lemma 6 (Finding S_h). (i) $\mathit{inter\_dist}(S_h) \leq \mathit{inter\_dist}(S_i)$ for any side cluster S_i of S; and **(ii)** there can be at most five side clusters S_i of S such that $\mathit{inter\_dist}(S_i) = \mathit{inter\_dist}(S_h)$.

Based on the above lemmas, we can construct a galled loop to connect the side clusters for S, as follows. By Lemma 6, there are at most five candidates for the side cluster attached to the hybrid node. We try to build the network using each of the candidate according to Lemma 3, 4 and 5. We verify each network constructed and return the one that satisfies S. Details of the algorithm are shown in Algorithm 2. It builds a network for S if and only if S is satisfiable.

Algorithm 2. ConnectingSideClusters $(S, S_1, S_2, \ldots, S_t)$, $t \geq 3$.

Find $\mathit{inter\_dist}(S_i)$ for each side cluster S_i and sort the side clusters according to the *inter_dist* value. If there are more than five side clusters having the minimum *inter_dist* value, return failure. Otherwise, for each side cluster S_h with the minimum *inter_dist* value, try to build a network which attaches S_h to the hybrid node, as follows.

a. Divide the remaining side clusters into two groups LEFT(S_h) and RIGHT(S_h) that satisfy Lemma 3.

b. Sort the side clusters in LEFT(S_h) according to the *inter_dist* value and attach the side clusters to the left merge path according to Lemma 4 and 5. Repeat it for the side clusters in RIGHT(S_h).

c. Let h_l and h_r be the height of the lowest side node on the left and right merge path, respectively. Set $\mathit{height}(h)$ to any value in $(\mathit{height}(S_h), \min\{h_l, h_r\})$.

d. Verify that for any two species $a, b \in S$, there is an evolutionary path between a and b with length equal to $M(a, b)$. Return the network if it is true.

Runtime of ConnectingSideClusters. It is straightforward to implement the procedure ConnectingSideClusters in $O(t_S \log t_S + \#_S)$ time, where t_S is the number of side clusters in S and $\#_S$ is the number of species pair (x, y) where x and y are species belonging to two different side clusters of S.

Over the whole execution of GalledNet, $\sum t_S \leq 2n - 1$ and $\sum \#_S \leq \frac{n(n-1)}{2}$. Thus, the total runtime for ConnectingSideClusters is $\sum O(t_S \log t_S + \#_S) = O(n \log n + \frac{n(n-1)}{2}) = O(n^2)$.

The Minimality of Number of Hybrid Nodes. Given a satisfiable matrix M, the network produced by GalledNet has the minimum number of hybrid nodes among all networks satisfying M. Proofs will be given in the full paper.

5 NP-Hardness Results

In the following, we say that a distance matrix M *admits* an ultrametric galled network if there exists such a network which satisfies M. We first prove that finding a maximum submatrix M' of a given distance matrix M such that M' admits an ultrametric galled network is an NP-hard problem. Our proof consists of a reduction from the NP-hard independent set problem.

The Independent Set Problem

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

The Maximum Submatrix Admitting an Ultrametric Galled Network Problem, Decision Problem Version (MSGN-d)

Instance: A set S, a distance matrix M for S, and a positive integer $K \leq |S|$.
Question: Is there a subset S' of S with $|S'| = K$ such that M restricted to S' admits an ultrametric galled network?

The following shows the reduction of the independent set problem to MSGN-d. Let (G, I) be any given instance of the independent set problem. For convenience, write $n = |V|$ and $V = \{v_1, v_2, \ldots, v_n\}$. Construct an instance (S, M, K) of MSGN-d as follows. Let $S = V \cup P \cup Q$, where $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ are two disjoint sets of elements not in V, and set $K = I + 2n$. Next, let M be a distance matrix for S satisfying, for every $i, j \in \{1, 2, \ldots, n\}$: $M(p_i, p_j) = \max\{i, j\}$; $M(q_i, q_j) = \max\{i, j\}$; $M(p_i, q_j) = n + 1$; $M(v_i, v_j) = \max\{i, j\}$ if the edge $\{i, j\}$ does not belong to E and $M(v_i, v_j) = n + 1$ if the edge $\{i, j\}$ belongs to E; $M(v_i, p_j) = n + 1$; and $M(v_i, q_j) = n + 1$.

Lemma 7. *M has a submatrix of size $K \times K$ which admits an ultrametric galled network if and only if G has an independent set of size I.*

Theorem 2. *MSGN is NP-hard.*

Next, we prove that it is NP-hard to determine whether a given incomplete distance matrix admits an ultrametric galled network. The proof consists of a reduction from the NP-hard 3-coloring problem.

The 3-Coloring Problem

Instance: An connected undirected graph $G = (V, E)$.

Question: Can G be 3-colored, i.e., can V be partitioned into three disjoint subsets in such a way that E contains no edge between two vertices in the same subset?

The Incomplete Distance Matrix Admitting an Ultrametric Galled Network Problem (IDGN)

Instance: A set S and an incomplete distance matrix M (i.e., where some entries are missing) for S.

Question: Is there an ultrametric galled network which satisfies all of the nonempty entries in M?

Let G be any given instance of 3-coloring with at least two vertices. Construct an instance (S, M) of IDGN by setting $S = V$ and defining the $(|S| \times |S|)$-matrix M as follows: for every $i \in V$, let $M(i, i) = 0$; and for every edge $\{i, j\} \in E$, let $M(i, j) = M(j, i) = 1$. For every pair of vertices i, j in V such that $\{i, j\} \notin E$, leave the matrix entries $M(i, j)$ and $M(j, i)$ empty.

Lemma 8. *G is 3-colorable if and only if there exists an ultrametric galled network which satisfies all of the nonempty entries in M.*

Theorem 3. *IDGN is NP-hard.*

Acknowledgement. We thank Wing-Kin Sung for introducing this problem to us and for his useful comments.

References

[1] 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.

[2] 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.

[3] H.-F. Chen and M.-S. Chang. An efficient exact algorithm for the minimum ultrametric tree problem. In *Proceedings of the 15^{th} International Symposium on Algorithms and Computation* (ISAAC 2004), pages 282–293, 2004.

[4] E. Diday and P. Bertrand. An extension of hierarchical clustering: the pyramidal representation. *Pattern Recognition in Practice II*, pages 411–424, 1986.

[5] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1/2):155–179, 1995.

[6] D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, New York, 1997.

[7] D. Gusfield and V. Bansal. A fundamental decomposition theory for phylogenetic networks and incompatible characters. In *Proceedings of the 9^{th} Annual International Conference on Research in Computational Molecular Biology* (RECOMB 2005), pages 217–232, 2005.

[8] 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.

[9] Y.-J. He, T. N. D. Huynh, J. Jansson, and W.-K. Sung. Inferring phylogenetic relationships avoiding forbidden rooted triplets. In *Proceedings of the 3^{rd} Asia-Pacific Bioinformatics Conference* (APBC 2005), pages 339–348, 2005.

[10] D. H. Huson, T. Dezulian, T. Klöpper, and M. Steel. Phylogenetic super-networks from partial trees. In *Proceedings of the 4^{th} Workshop on Algorithms in Bioinformatics* (WABI 2004), pages 388–399, 2004.

[11] T. N. D. Huynh, J. Jansson, N. B. Nguyen, and W.-K. Sung. Constructing a smallest refining galled phylogenetic network. In *Proceedings of the 9^{th} Annual International Conference on Research in Computational Molecular Biology* (RECOMB 2005), pages 265–280, 2005.

[12] J. Jansson, N. B. Nguyen, and W.-K. Sung. Algorithms for combining rooted triplets into a galled phylogenetic network. In *Proc. of the 16^{th} Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA 2005), pages 349–358, 2005.

[13] J. Jansson and W.-K. Sung. The maximum agreement of two nested phylogenetic networks. In *Proceedings of the 15^{th} International Symposium on Algorithms and Computation* (ISAAC 2004), pages 581–593, 2004.

[14] L. Nakhleh, T. Warnow, and C. R. Linder. Reconstructing reticulate evolution in species – theory and practice. In *Proceedings of the 8^{th} Annual International Conference on Research in Computational Molecular Biology* (RECOMB 2004), pages 337–346, 2004.

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

[16] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

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

[18] B. Y. Wu, K.-M. Chao, and C. Y. Tang. Approximation and exact algorithms for constructing minimum ultrametric trees from distance matrices. *Journal of Combinatorial Optimization*, 3(2–3):199–211, 1999.