COUNTING TREES IN A PHYLOGENETIC NETWORK IS #P-COMPLETE [*]

SIMONE LINZ[†], KATHERINE ST. JOHN[‡], AND CHARLES SEMPLE[§]

Abstract. Answering a problem posed by Nakhleh, we prove that counting the number of phylogenetic trees inferred by a (binary) phylogenetic network is #P-complete. An immediate consequence of this result is that counting the number of phylogenetic trees commonly inferred by two (binary) phylogenetic networks is also #P-complete.

Key words. Phylogenetic trees, phylogenetic networks, #P-complete

1. Introduction. A fundamental problem in evolutionary biology is to represent the ancestral history of a collection of present-day species with a phylogenetic (evolutionary) tree. In the reconstruction of such trees, maximum parsimony methods have a long-standing history. However, recently, evolutionary biologists have become increasingly interested in the reconstruction of phylogenetic networks since such networks have the potential to give a more complete picture of the evolutionary past by including reticulation (non-tree-like) events such as hybridization and recombination [3, 6]. Given this, it is natural to reconstruct phylogenetic networks with parsimony methods that can handle the much more complex structure of networks. To date, research in this area has mostly focused on the so-called small maximum parsimony problem which calculates the parsimony score for a given phylogenetic network. Introduced by Hein [5] and formally stated by Nakhleh [12], this problem is as follows. Given a phylogenetic network $\mathcal{N}$ and a character c which assigns the state of some particular attribute to each of the species under consideration, the parsimony score of c on $\mathcal{N}$ is defined to be the minimum over all "parsimony scores" of c on any rooted phylogenetic tree that is inferred by $\mathcal{N}$. This optimization problem was shown to be NP-hard for recombination networks [13] and also for a more restricted type of network that fulfills certain temporal constraints [9] by establishing a polynomial-time reduction from Minimum Set Cover and Maximum 2 Satisfiability, respectively. Both reductions directly relate the parsimony score of a phylogenetic network $\mathcal{N}$ to the solution of the problem instance from which $\mathcal{N}$ was constructed.

Nakhleh [18] recently posed the problem of counting the number of distinct rooted phylogenetic trees inferred by a given phylogenetic network. This problem is closely

[*] The first author was supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme. The second author was supported by the US National Science Foundation (#0920920) and from the Lehman College/IBM Research Travel Fund. The third author was supported by the Allan Wilson Centre for Molecular Ecology and Evolution and the New Zealand Marsden Fund.

[†] Center for Bioinformatics, University of Tübingen, Tübingen, Germany (present address: Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand), linz@informatik.uni-tuebingen.de.

[‡] Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, NY, United States, stjohn@lehman.cuny.edu.

[§] Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand, charles.semple@canterbury.ac.nz.

related to solving the small maximum parsimony problem for networks when one considers the number of trees inferred by a network for which one has to find the tree that realizes the minimum parsimony score. The main result of this paper shows that this problem is #P-complete. This means that computing the number of distinct rooted phylogenetic trees inferred by a phylogenetic network is as hard as computing the permanent of a matrix or the number of Hamiltonian circuits in a graph and, thus, it is very unlikely that there exists a polynomial-time algorithm for computing this number. An almost immediate consequence is that counting the number of phylogenetic trees commonly inferred by two networks is #P-complete. This count could easily provide a basis for a new approach to assessing the dissimilarity between two phylogenetic networks. Other such approaches have for example been discussed in [1, 7].

For the interested reader, we remark that Kannan and Wheeler [11] have recently introduced an alternative definition of the parsimony score of a phylogenetic network that does not reflect the parsimony score of the best tree inferred by the network. Using their definition, they showed that well-known algorithms, such as the Sankoff and Fitch algorithm (e.g., see [4]), that efficiently calculate the maximum parsimony score for a rooted phylogenetic tree naturally extend to networks and provide heuristics for estimating this score quickly.

Throughout the paper, X denotes a finite set. A *rooted phylogenetic X-tree* $\mathcal{T}$ is a rooted tree in which the root has degree at least two and all other interior vertices have degree at least three, and whose leaf set is X. In addition, $\mathcal{T}$ is *binary* if, apart from the root which has degree two, all interior vertices have degree three. The leaf set of $\mathcal{T}$ is often referred to as the *label set* of $\mathcal{T}$ and is denoted by $\mathcal{L}(\mathcal{T})$. Since we are only interested in rooted, and not unrooted, phylogenetic trees, we will always refer to a rooted phylogenetic tree as a phylogenetic tree.

A *phylogenetic network* $\mathcal{N}$ *on* X is a rooted acyclic digraph with the following properties:

(i) the root has out-degree two;
(ii) a vertex with out-degree zero has in-degree one, and the set of vertices with out-degree zero is X;
(iii) all other vertices either have in-degree one and out-degree two, or in-degree two and out-degree one.

For a phylogenetic network $\mathcal{N}$, vertices with in-degree two and out-degree one are called *reticulation vertices* and edges directed into a reticulation vertex are called *reticulation edges*. Furthermore, vertices with out-degree zero are referred to as *leaves*, and the set X is referred to as the *label set* of $\mathcal{N}$. These networks are commonly referred to as *binary* networks.

Let $\mathcal{T}$ be a phylogenetic X-tree and let $\mathcal{N}$ be a phylogenetic network on X. We say that $\mathcal{N}$ *displays* $\mathcal{T}$ if $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting edges and vertices, and contracting vertices of degree two. Intuitively, $\mathcal{T}$ is displayed by $\mathcal{N}$ if the ancestral information inferred by $\mathcal{T}$ is also inferred by $\mathcal{N}$. Note that all phylogenetic trees displayed by $\mathcal{N}$ are binary. For ease of reading, if we say that a phylogenetic tree is displayed by a phylogenetic network without reference to the label sets, then we will assume that these sets are identical.

Our interest in this paper is in counting the number of phylogenetic X-trees displayed by a given phylogenetic network $\mathcal{N}$ on X. In particular, we are interested in the following counting problem:

#Trees in a Phylogenetic Network
Instance: A phylogenetic network $\mathcal{N}$ on X.
Question: How many phylogenetic X-trees are displayed by $\mathcal{N}$?

If $\mathcal{N}$ has r reticulation vertices, then it displays at most 2^r phylogenetic X-trees (e.g., see [10]). Here each phylogenetic tree is essentially obtained by deleting, for each reticulation vertex, exactly one of its incident reticulation edges. For example, this bound is sharp for the class of normal networks that are so-called 'tree-child networks' that satisfy an additional property [17]. However, in general, 2^r is simply an upper bound and there is no difficulty in finding instances for which this bound is not sharp. For example, the network displayed in Figure 2.1 with $t-1$ reticulation vertices and $t+1$ leaves displays $F_t \sim (1.618)^t$ trees, where F_t is the t-th Fibonacci number (see Lemmas 2.5 and 2.6).

The main result of this paper is Theorem 1.1 which shows that the above counting problem is computationally hard.

Theorem 1.1. *Computing* #Trees in a Phylogenetic Network *is* #P-*complete.*

Clearly, the result of Theorem 1.1 generalizes to counting all phylogenetic trees that are displayed in a phylogenetic network where vertices may have higher in-degree or out-degree. We remark here that an associated decision problem is NP-complete. In particular, for a given binary phylogenetic X-tree $\mathcal{T}$ and phylogenetic network $\mathcal{N}$ on X, determining whether $\mathcal{N}$ displays $\mathcal{T}$ is an NP-complete problem [10]. Indeed, it is also NP-complete for when $\mathcal{N}$ is restricted to various types of phylogenetic networks [8].

A problem closely related to #Trees in a Phylogenetic Network is the following counting problem.

#Common Trees Between Two Phylogenetic Networks
Instance: Two phylogenetic networks $\mathcal{N}_1$ and $\mathcal{N}_2$ on X.
Question: How many common phylogenetic X-trees are displayed by $\mathcal{N}_1$ and $\mathcal{N}_2$?

Again this problem is computationally hard as stated in Corollary 1.2. Its proof is an almost immediate consequence of Theorem 1.1.

Corollary 1.2. *Computing* #Common Trees Between Two Phylogenetic Networks *is* #P-*complete.*

The rest of this section contains some preliminaries including the counting problem which we use for our reductions to establish Theorem 1.1 and Corollary 1.2. The proofs of these results are given in Section 2.

Let $\mathcal{T}$ be a phylogenetic X-tree. A pair of leaves $\{a, b\}$ of $\mathcal{T}$ is a *cherry* if a and b are adjacent to a common vertex of $\mathcal{T}$. Now let $(x_1, x_2, \ldots, x_n)$ be an n-tuple. A phylogenetic tree $\mathcal{T}$ is the *caterpillar on* $(x_1, x_2, \ldots, x_n)$ if the leaf set of $\mathcal{T}$ is $\{x_1, x_2, \ldots, x_n\}$ and the following properties are satisfied:

(i) $\{x_1, x_2\}$ is the only cherry of $\mathcal{T}$ and
(ii) for each $i \in \{2, \ldots, n-1\}$, the parent of x_i is a child of the parent of x_{i+1}.

We next describe the counting problem that we use for our reductions. A *matching* of a graph $G = (V, E)$ is a subset M of E such that no two edges in M are incident with a common vertex in G. A matching M is *perfect* if every vertex in V is incident with an edge in M. A graph is 3-*regular* if the degree of each vertex is three. Dagum and Luby [2, Theorem 6.2] showed that the following counting problem is #P-complete.

#Perfect Matchings in 3-Regular Bipartite Graphs
Instance: A 3-regular bipartite graph G.
Question: How many perfect matchings are in G?

For the rest of the paper, we will take $G = (V_A \cup V_B, E)$ to be an instance of #Perfect Matchings in 3-Regular Bipartite Graphs, where $V_A = \{a_1, a_2, \ldots, a_k\}$ and $V_B = \{b_1, b_2, \ldots, b_k\}$. We may assume that $|V_A| = |V_B|$; otherwise the number of perfect matchings in G is zero. Furthermore, for each $j \in \{1, 2, \ldots, k\}$, we denote the vertices in V_A that are adjacent to b_j by $a_{j'}, a_{j''}, a_{j'''}$.

Given G, construct a phylogenetic network $\mathcal{N}$ as follows. Start with the caterpillar on $(x, y, a_1, a_2, \ldots, a_k)$, where x and y are new labels not in $V_A \cup V_B$. In turn, for each $j \in \{1, 2, \ldots, k\}$, adjoin b_j to the caterpillar by first adding an edge joining b_j and the pendant edge incident with $a_{j'}$, then adding an edge joining the pendant edge incident with b_j and the pendant edge incident with $a_{j''}$, and lastly adding an edge joining the pendant edge incident with b_j and the pendant edge incident with $a_{j'''}$. The reduction does not depend on the order in which the elements in V_B are chosen or the order in which the elements adjacent to b_j in G are chosen. The construction of $\mathcal{N}$ from G is illustrated in Figure 1.1.

Remark. Informally, a phylogenetic network is *temporal* if speciation events occur successively while reticulation events occur instantaneously. Jin et al. [9] showed that the small parsimony problem is NP-hard for such network. By imposing an ordering on the elements in V_B and considering this ordering for the pairing of each a_i with its three elements in V_B, it is easily checked that the network shown in Figure 1.1 (resp. the network shown in Figure 2.2) is temporal. The overall effect of making such an ordering is that #Trees in a Phylogenetic Network remains #P-complete for the restricted class of temporal networks.

2. Proofs of Theorem 1.1 and Corollary 1.2. The overall strategy of the proof of Theorem 1.1 is to take the instance $G = (V_A \cup V_B, E)$ of #Perfect Matchings in 3-Regular Bipartite Graphs, construct the phylogenetic network $\mathcal{N}$, and then incrementally extend $\mathcal{N}$ to a new phylogenetic network k times. The number of phylogenetic trees displayed by $\mathcal{N}$, and the number of each of the incremental extensions can be written in terms of the Fibonacci numbers. Using an oracle that counts the number of phylogenetic trees displayed by a phylogenetic network and facts about

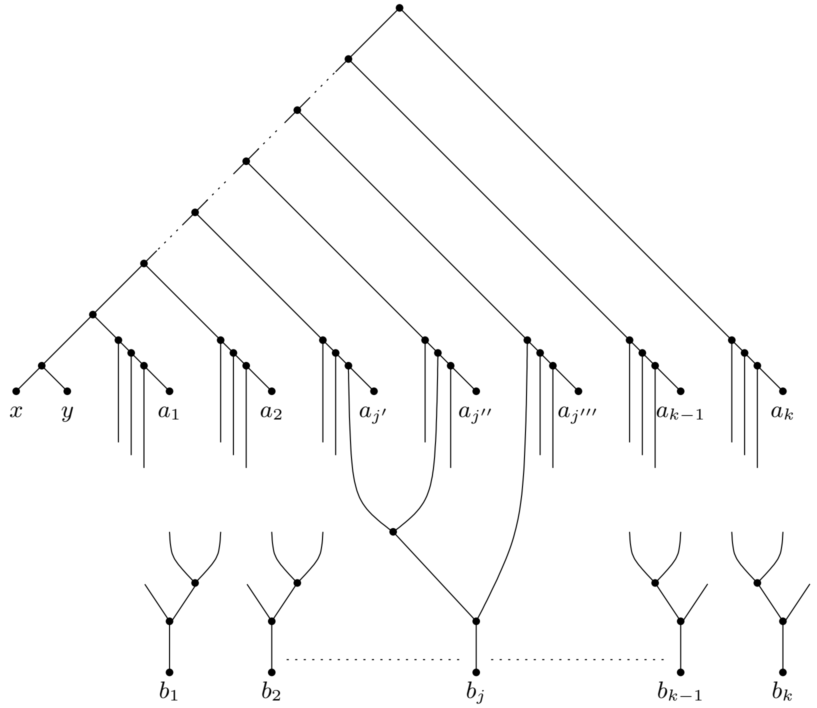

FIG. 1.1. *A phylogenetic network $\mathcal{N}$ reconstructed from an instance $G = (V_A \cup V_B, E)$ of* #PERFECT MATCHINGS IN 3-REGULAR BIPARTITE GRAPHS *with $\{\{a_{j'}, b_j\}, \{a_{j''}, b_j\}, \{a_{j'''}, b_j\}\} \subset E$. Since each $a_i \in V_A$ is adjacent to three vertices in V_B, each of the three "dangling" edges associated with a_i is paired with a "dangling" reticulation edge in the bottom part of the figure. The actual pairing depends on the elements in E.*

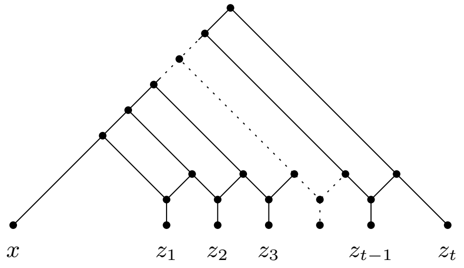

FIG. 2.1. *A caterpillar chain of length t.*

the Fibonacci numbers, one can recover the number of phylogenetic trees displayed by $\mathcal{N}$ of a certain type. This last number is the number of perfect matchings of G. The general technique of incremental extensions and relating it to the Fibonacci numbers is due to Vadhan [16].

To begin the proof of Theorem 1.1, consider the phylogenetic network shown in Figure 2.1, where $t \geq 1$. We call such a network a *caterpillar chain of length t*. We refer to the leaf labeled x as the *tail* while all other leaves are referred to as *chain*

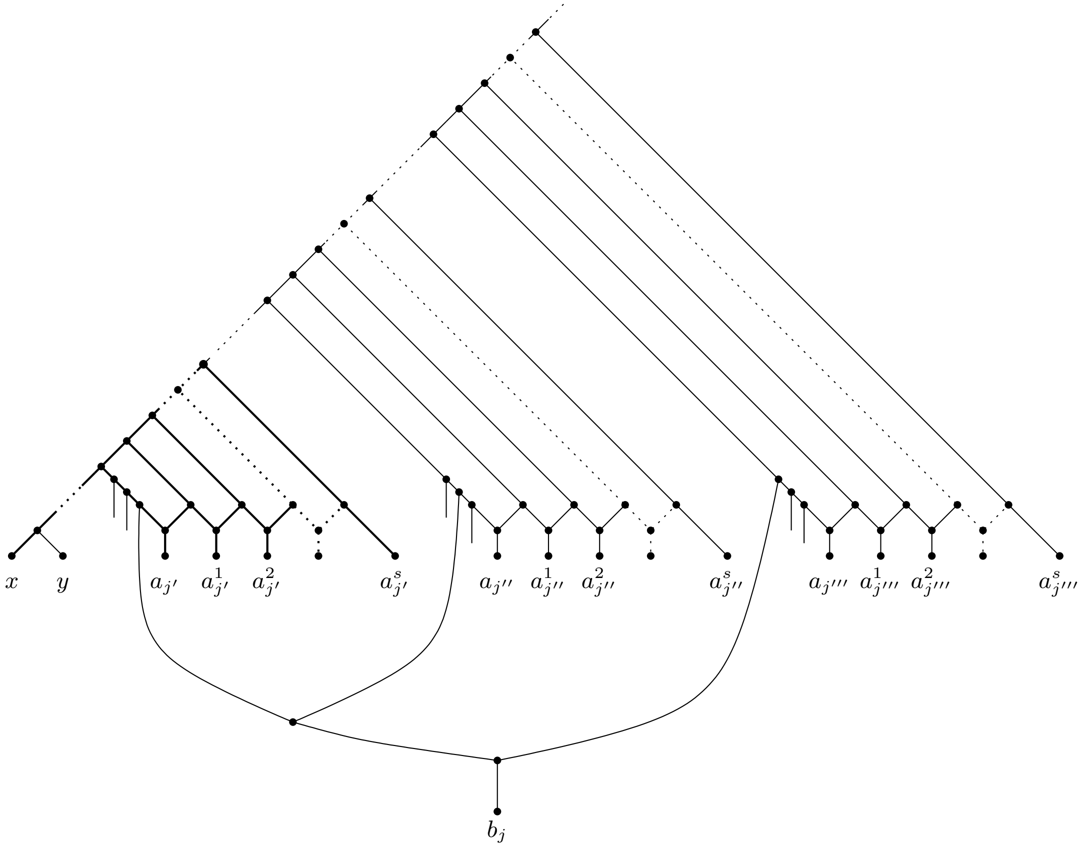

Fig. 2.2. *The phylogenetic network $\mathcal{N}_s$ obtained from the phylogenetic network that is shown in Figure 1.1. In particular, $\mathcal{N}_s$ is obtained by replacing each of the leaves $a_1, a_2, \ldots, a_k$ with a caterpillar chain of length $s+1$. Bold edges indicate one such caterpillar chain. For simplicity, we have omitted some parts of the phylogenetic network shown in Figure 1.1.*

leaves. The chain leaves are ordered, in this case the ordering is $(z_1, z_2, \ldots, z_t)$. A caterpillar chain of length t has $t-1$ reticulation vertices.

Now, let $\mathcal{N}_0$ be the phylogenetic network $\mathcal{N}$ constructed from the instance of #Perfect Matchings in 3-Regular Bipartite Graphs as described at the end of Section 1. For each $s \in \{1, 2, \ldots, k\}$, let $\mathcal{N}_s$ denote the phylogenetic network obtained from $\mathcal{N}_0$ by "attaching" k caterpillar chains of length $s+1$. In particular, for each leaf a_i with $i \in \{1, 2, \ldots, k\}$ in $\mathcal{N}_0$, we attach a caterpillar chain with tail x and whose chain leaves are ordered $(a_i, a_i^1, a_i^2, \ldots, a_i^s)$ by identifying the leaves x and a_i with their namesakes in $\mathcal{N}_0$, and identifying certain edges with certain paths in $\mathcal{N}_0$. An illustration of which edges and which paths are identified is shown in Figure 2.2, where the bold edges show the attachment of the caterpillar chain whose chain leaves are ordered $(a_{j'}, a_{j'}^1, a_{j'}^2, \ldots, a_{j'}^s)$.

We next establish some preliminary results, which in turn require further definitions. Let $\mathcal{T}$ be a phylogenetic tree displayed by $\mathcal{N}_s$, where $s \in \{0, 1, 2, \ldots, k\}$, and let a_i be an element of V_A. Relative to $\mathcal{T}$, we say that a_i is *paired* if a_i is in a cherry in $\mathcal{T}$ with an element in V_B. Note that each a_i is paired precisely if each b_j belongs to a cherry. The first two lemmas are straightforward consequences of the construction; their proofs are omitted.

Lemma 2.1. *Let $\mathcal{T}$ be a phylogenetic tree displayed by $\mathcal{N}_s$, where $s \in \{0, 1, 2, \ldots, k\}$. If each element in V_A is paired in $\mathcal{T}$, then the collection*

$$\big\{\{a_i, b_j\} : \{a_i, b_j\} \textit{ is a cherry of } \mathcal{T}, \textit{ where } a_i \in V_A \textit{ and } b_j \in V_B\big\}$$

is the collection of edges of a perfect matching M of G.

The converse of Lemma 2.1 also holds.

Lemma 2.2. *Let M be a perfect matching in G. Then, for each $s \in \{0, 1, 2, \ldots, k\}$, there is a phylogenetic tree $\mathcal{T}$ displayed by $\mathcal{N}_s$ such that, for each edge $\{a_i, b_j\}$ in M with $a_i \in V_A$ and $b_j \in V_B$, there exists a cherry $\{a_i, b_j\}$ in $\mathcal{T}$.*

Now, for some $s \in \{0, 1, \ldots, k\}$, let $\mathcal{T}'$ be a phylogenetic tree displayed by $\mathcal{N}_s$ and let $\mathcal{T}$ be a phylogenetic tree displayed by $\mathcal{N}_0$. We say that $\mathcal{T}'$ is an *extension* of $\mathcal{T}$ in $\mathcal{N}_s$ if, up to contracting degree two vertices, the minimal subtree of $\mathcal{T}'$ connecting the elements in $\mathcal{L}(\mathcal{T})$ is isomorphic to $\mathcal{T}$. Furthermore, for all $t \geq 1$, let w_t denote the number of phylogenetic trees displayed by a caterpillar chain of length t. For $t = 0$, set $w_0 = 1$.

Lemma 2.3. *Let $\mathcal{T}$ be a phylogenetic tree displayed by $\mathcal{N}_0$ and let $s \in \{1, 2, \ldots, k\}$. Then the number of ways that $\mathcal{T}$ can be extended to a phylogenetic tree in $\mathcal{N}_s$ is $(w_s)^p(w_{s+1})^{k-p}$, where p is the number of elements of V_A paired in $\mathcal{T}$.*

Proof. It is easily seen that if an element, a_i say, of V_A is paired in $\mathcal{T}$, then $\mathcal{T}$ can be extended with any phylogenetic tree displayed by the caterpillar chain whose leaf set of chain leaves is $\{a_i^1, a_i^2, \ldots, a_i^s\}$. On the other hand, if a_i is not paired in $\mathcal{T}$, then $\mathcal{T}$ can be extended with any phylogenetic tree displayed by the caterpillar chain whose leaf set of chain leaves is $\{a_i, a_i^1, a_i^2, \ldots, a_i^s\}$. As the caterpillar chains in $\mathcal{N}_s$ act independently of each other, we deduce the lemma by multiplying over all elements in V_A. □

Let t_p denote the number of phylogenetic trees displayed by $\mathcal{N}_0$ in which exactly p elements of V_A are paired. By construction, the next lemma is an immediate consequence of Lemma 2.3.

Lemma 2.4. *For each $s \in \{0, 1, 2, \ldots, k\}$, the phylogenetic network $\mathcal{N}_s$ displays exactly*

$$\sum_{p=0}^{k} t_p(w_s)^p(w_{s+1})^{k-p}$$

phylogenetic trees.

Lemma 2.5. *The number w_t of phylogenetic trees displayed by a caterpillar chain of length t is equal to the number of matchings in a path of t vertices.*

Proof. Let $\mathcal{C}_t$ be a caterpillar chain of length t whose chain leaves are ordered $(d_1, d_2, \ldots, d_t)$. If $\mathcal{T}$ is a phylogenetic tree displayed by $\mathcal{C}_t$, then the collection of cherries of $\mathcal{T}$ is a subset of

$$\Phi = \big\{\{x, d_1\}, \{d_1, d_2\}, \{d_2, d_3\}, \ldots, \{d_{t-1}, d_t\}\big\}.$$

Now let $P = d_1 d_2 \cdots d_t$ be a path of t vertices. If M is a matching of P, then M is a subset of

$$\big\{\{d_1, d_2\}, \{d_2, d_3\}, \ldots, \{d_{t-1}, d_t\}\big\}.$$

Therefore, if $\mathcal{T}$ is a phylogenetic tree displayed by $\mathcal{C}_t$, then the cherries of $\mathcal{T}$ in $\Phi - \big\{\{x, d_1\}\big\}$ are the edges of a matching of P. If the only cherry of $\mathcal{T}$ is $\{x, d_1\}$, we have the empty matching. Note that there is precisely one such tree that has this property. Thus, the number of such trees is at most the number of matchings of P. Moreover, if M is a matching of P, then it is easily seen that M is the collection of cherries of a phylogenetic tree displayed by $\mathcal{C}_t$. Here the empty matching is realized by the caterpillar on $(x, d_1, d_2, \ldots, d_t)$. Hence the number of such matchings is at most the number of phylogenetic trees displayed by $\mathcal{C}_t$. The lemma now follows. □

The next lemma is established in [16, Lemma 6.3]. Recall that the n-th Fibonacci number, denoted F_n, is given by the recursion $F_0 = F_1 = 1$ and $F_{n+2} = F_{n+1} + F_n$ for all $n \geq 0$.

Lemma 2.6. *For all $n \geq 1$, the number of matchings in a path of n vertices is F_n. Moreover, for all $n \geq 0$, the ratio $F_m/F_{m+1} \neq F_n/F_{n+1}$ for all $m \neq n$.*

With regards to the last lemma, observe that if $n = 2$, the path has two matchings, namely, the empty matching and the matching consisting of the single edge in the path. This is consistent with $F_2 = 2$. We now complete the proof of our main result.

Proof. [Proof of Theorem 1.1] To show that #Trees in a Phylogenetic Network is in #P, let $\mathcal{N}$ be a given instance of the problem and let $\mathcal{T}$ be an arbitrary phylogenetic X-tree. Let D be a binary decision tree, where each level corresponds to a distinct reticulation vertex of $\mathcal{N}$ and where the branching at a vertex corresponds to the two possible deletions of a single reticulation edge incident with the reticulation vertex for that level. Hence, each tree displayed by $\mathcal{N}$ is associated with at least one leaf of D. Using D, one can verify whether $\mathcal{N}$ displays $\mathcal{T}$ non-deterministically in polynomial time. Thus, #Trees in a Phylogenetic Network is in #P. We next show that it is #P-complete.

Using an oracle that can compute the answer to an instance of #Trees in a Phylogenetic Network in polynomial time, we can determine the value of

$$\sum_{p=0}^{k} t_p (w_s)^p (w_{s+1})^{k-p},$$

the number of phylogenetic trees displayed by $\mathcal{N}_s$, for each $s \in \{0, 1, 2, \ldots, k\}$, in polynomial time. This takes $k + 1$ oracle calls. Furthermore, by Lemmas 2.5 and 2.6, w_s is the s-th Fibonacci number and, thus, can be computed in polynomial time (e.g., see [15]). Dividing through by $(w_{s+1})^k$, we obtain the evaluations

$$p(w_s/w_{s+1}) = \sum_{p=0}^{k} t_p \left(\frac{w_s}{w_{s+1}}\right)^p$$

of the polynomial

$$p(x) = \sum_{p=0}^{k} t_p x^p$$

at the points $\frac{w_s}{w_{s+1}}$ for each $s \in \{0, 1, \ldots, k\}$. Now, by Lemma 2.6, the points $\frac{w_s}{w_{s+1}}$ are distinct and so, by [16, Fact 5.1], we can recover the coefficients of $p(x)$ in polynomial time. In particular, we can recover the coefficient t_k, the number of phylogenetic trees displayed by $\mathcal{N}_0$ in which each element of V_A is paired. By Lemmas 2.1 and 2.2, t_k is also the number of perfect matchings in G. Since the time to construct $\mathcal{N}_s$ and the size of $\mathcal{N}_s$ for each $s \in \{0, 1, \ldots, s\}$ is polynomial in the size of G, it now follows that computing #TREES IN A PHYLOGENETIC NETWORK is #P-complete. □

Proof. [Proof of Corollary 1.2] To see that #COMMON TREES BETWEEN TWO PHYLOGENETIC NETWORKS is in #P, let $\mathcal{N}_1$ and $\mathcal{N}_2$ be a given instance of the problem and let $\mathcal{T}$ be an arbitrary phylogenetic X-tree. For $i \in \{1, 2\}$, let D_i be a binary decision tree, where each level corresponds to a distinct reticulation vertex of $\mathcal{N}_i$ and where the branching at a vertex corresponds to the two possible deletions of a single reticulation edge incident with the reticulation vertex for that level. Using D_i, one can verify whether $\mathcal{N}_i$ displays $\mathcal{T}$, and therefore whether both $\mathcal{N}_1$ and $\mathcal{N}_2$ display $\mathcal{T}$, nondeterministically in polynomial time. It follows that #COMMON TREES BETWEEN TWO PHYLOGENETIC NETWORKS is in #P.

To see that the problem is #P-complete, let $\mathcal{N}$ be an instance of #TREES IN A PHYLOGENETIC NETWORK. By choosing $\mathcal{N}_1$ and $\mathcal{N}_2$ to both be $\mathcal{N}$, the number of common phylogenetic trees displayed by $\mathcal{N}_1$ and $\mathcal{N}_2$ is precisely the number of phylogenetic trees displayed by $\mathcal{N}$. This completes the proof of the corollary. □

Acknowledgements: We thank Dominic Welsh for helpful discussions and the anonymous referees for their insightful comments.

REFERENCES

[1] G. CARDONA, F. ROSSELLÓ, G. VALIENTE, *Comparison of tree-child phylogenetic networks*, IEEE Trans. Comput. Biol. Bioinform., 6 (2009), pp. 552–569.

[2] P. DAGUM, M. LUBY, *Approximating the permanent of graphs with large factors*, Theoret. Comput. Sci., 102 (1992), pp. 283–305.

[3] W. F. DOOLITTLE, E. BAPTESTE, *Pattern pluralism and the Tree of Life hypothesis*, Proc. Natl. Acad. Sci. USA., 104 (2007), pp. 2043–2049.

[4] J. FELSENSTEIN, *Inferring Phylogenies*, Sinauer Associates, 2004.

[5] J. HEIN, *Reconstructing evolution of sequences subject to recombination using parsimony*, Math. Biosci., 98 (1990), pp. 185–200.

[6] D. H. HUSON, D. BRYANT, *Application of phylogenetic networks in evolutionary studies*, Mol. Biol. Evol., 23 (2006), pp. 254–267.

[7] D. H. HUSON, R. RUPP, C. SCORNAVACCA, *Phylogenetic Networks: Concepts, Algorithms and Applications*, Cambridge University Press, 2010.

[8] L. van Iersel, C. Semple, M. Steel, *Locating a tree in a phylogenetic network*, Inform. Process. Lett., 110 (2010), pp. 1037–1043.

[9] G. Jin, L. Nakhleh, S. Snir, T. Tuller, *Parsimony score of phylogenetic networks: hardness results and a linear-time heuristic*, IEEE Trans. Comput. Biol. Bioinf., 6 (2009), pp.495–505.

[10] I. A. Kanj, L. Nakheleh, C. Than, G. Xia, *Seeing the trees and their branches in the network is hard*, Theoret. Comput. Sci., 401 (2008), pp. 153–164.

[11] L. Kannan, W. Wheeler, *Maximum parsimony on phylogenetic networks*, Algorithm. Mol. Biol., 7 (2012), 9.

[12] L. Nakhleh, G. Jin, F. Zhao, J. Mellor-Crummey, *Reconstructing phylogenetic networks using maximum parsimony*, In IEEE Computational Systems Bioinformatics Conference 2005, pp. 440–442.

[13] C. T. Nguyen, N. B. Nguyen, W.-K. Sung, L. Zhang, *Reconstructing recombination network from sequence data: the small parsimony problem*, IEEE Trans. Comput. Biol. Bioinf., 4 (2007), pp. 394–402.

[14] C. Semple, M. Steel, *Phylogenetics*, Oxford University Press, 2003.

[15] A. Tucker, *Applied Combinatorics*, 3rd ed., John Wiley & Sons, New York, 1995.

[16] S. P. Vadhan, *The complexity of counting in sparse, regular, and planar graphs*, SIAM J. Comput., 31 (2001), pp. 398–427.

[17] S. Willson, *Properties of normal phylogenetic networks*, B. Math. Biol., 72 (2010), pp. 340–358.

[18] www.math.canterbury.ac.nz/bio/events/south2012/files/penny_ante_problems.pdf