

Efficient Parsimony-based Methods for Phylogenetic Network Reconstruction

Guohua Jin^a, Luay Nakhleh^a, Sagi Snir^b, Tamir Tuller^c

^aDept. of Computer Science, Rice University, Houston, TX, USA, ^bDept. of Mathematics, University of California, Berkeley, CA, USA, ^cSchool of Computer Science, Tel Aviv University, Tel Aviv, Israel

ABSTRACT

Motivation: Phylogenies—the evolutionary histories of groups of organisms—play a major role in representing relationships among biological entities. Although many biological processes can be effectively modeled as a tree-like relationships, others, such as hybrid speciation, and horizontal gene transfer (HGT) result in *networks* of relationships rather than trees of relationships. Hybrid speciation is a significant evolutionary mechanism in plants, fish, and other groups of species. HGT plays a major role in bacterial genome diversification, and is a significant mechanism by which bacteria develop resistance to antibiotics. Maximum parsimony (MP) is one of the most commonly used criteria for phylogenetic tree inference. Roughly speaking, inference based on this criterion seeks the tree that minimizes the amount of evolution. In 1990, Jotun Hein proposed using this criterion for inferring the evolution of sequences subject to recombination. Preliminary results on small synthetic data sets (Nakhleh *et al.*, 2005) demonstrated the criterion's application to phylogenetic network reconstruction in general, and HGT detection in particular. However, the naive algorithms used by the authors are inapplicable to large data sets due to their demanding computational requirements. Further, no rigorous theoretical analysis of computing the criterion was given, nor was it tested on biological data.

Results: In this work, we prove that the problem of scoring the parsimony of a phylogenetic network is NP-hard, and provide an improved fixed parameter tractable algorithm for it. Further, we devise an efficient heuristics for parsimony-based reconstruction of phylogenetic networks. We test our methods on both synthetic and biological data (rbcl gene in bacteria), and obtain very promising results. (Due to space limitations, some proofs are omitted; the full version of the paper is available at <http://bioinfo.cs.rice.edu/Papers/eccb06full.pdf>.)

Contact: Sagi Snir (ssagi@math.berkeley.edu)

1 INTRODUCTION

Phylogenetic networks are a special class of *directed acyclic graphs* (DAGs) that models evolutionary histories when trees are inappropriate, such as in the cases of horizontal gene transfer (HGT) and hybrid speciation [16, 19, 18]. Fig. 1(a) shows

a phylogenetic network on four species with a single HGT event. In horizontal gene transfer (HGT), genetic material is transferred from one lineage to another, as in Fig. 1(a). In an evolutionary scenario involving horizontal transfer, certain sites (specified by a specific substring within the DNA sequence of the species into which the horizontally transferred DNA was inserted) are inherited through horizontal transfer from another species (as in Fig. 1(c)), while all others are inherited from the parent (as in Fig. 1(b)). Thus, *each site evolves down one of the trees induced by (or, contained in) the network*. Similar scenarios arise in the cases of other reticulate evolution events (such as hybrid speciation and interspecific recombination). Hybrid speciation is a signifi-

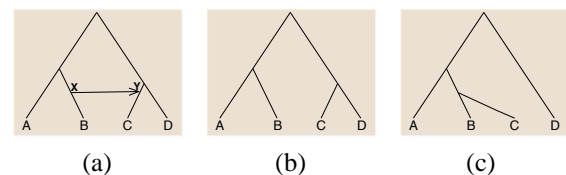


Fig. 1. (a) A phylogenetic network with a single HGT even from X to Y . (b) The underlying organismal (species) tree. (c) The tree of a horizontally transferred gene.

cant evolutionary mechanism in plants, fish, and other groups of species [17]. HGT plays a major role in bacterial genome diversification [2, 4], and is a significant mechanism by which bacteria develop resistance to antibiotics [5]. To facilitate evolutionary analyses of these groups of organisms, developing accurate criteria for reconstructing phylogenetic networks and efficient algorithms for inference based on these criteria are imperative. A large number of publications have been introduced in recent years about various aspects of phylogenetic networks; see [16, 18] for detailed surveys.

Maximum Parsimony (MP) is one of the most widely used criteria for phylogenetic tree analysis. It is based on a minimum-evolution principle, compares well to other accurate criteria, and has a host of efficient algorithms for solving problems based on it [6, 8]. In 1990, Hein observed that the criterion could be extended to detect recombination [10, 11]. He observed that each individual site in a set of sequences

labeling a network evolves down a tree contained in the network (e.g., the trees in Figs. 1(b) and 1(c) are contained in the network shown in Fig. 1(a)). Following this observation, Nakhleh *et al.* formulated the parsimony criterion for inferring and evaluating phylogenetic networks [20]. The HGT reconstruction problem seeks an optimal set of edges whose addition to a given species tree results in an optimal network that explains the given gene data. In the context of parsimony, we refer to this problem as the *fixed-tree MP phylogenetic network* problem, or FTMPNN. Solving this problem entails scoring the parsimony of a phylogenetic network leaf-labeled by a set of sequences; we refer to this problem as the *parsimony score of phylogenetic network* problem, or PSPN. Nakhleh *et al.* used a straightforward algorithm (exponential in the number of reticulation edges) for solving the PSPN problem, and exhaustively searched *all* networks for solving the FTMPNN problem. Further, they left open the question of the computational complexity of these problems.

In this work, we prove that the PSPN problem is NP-hard, and hard to approximate. However, on the positive side, we give an efficient algorithm for the problem, and bound its running time to prove it is fixed parameter tractable [3]. The algorithm has very good performance in practice, as we show, and was integrated as part of efficient heuristics for the FTMPNN problem. Further, we devise new heuristics for the FTMPNN problem, and show through experiments on biological as well as synthetic data, that the heuristics are efficient in practice, while maintaining a high accuracy. The biological dataset we analyze include the *rbcL* gene in plastids, cyanobacteria, and proteobacteria. The evolution of this gene is believed to include a set of horizontal gene transfer events [1].

A large body of work has been introduced in recent years to address phylogenetic network reconstruction and evaluation. In general, three categories of non-treelike models have been addressed, all of which have been introduced under the umbrella concept of phylogenetic networks. However, major differences exist among the three categories. *Splits networks* are graphical models that capture incompatibilities in the data due to various factors, not necessarily HGT or hybrid speciation. *Phylogenetic networks* are the extension of phylogenetic trees to enable the modeling of reticulation events, such as HGT and hybrid speciation (these are also called *reticulate networks* in [13]). The third category is that of *recombination networks*, which are used to model the evolution of haplotypes and genes at the population level. See [16, 18] for detailed surveys of the various phylogenetic network models and methodologies. Phylogenetic networks that we address in this work belong to the second category.

2 PARSIMONY OF NETWORKS

2.1 Preliminaries and Definitions

Let $T = (V, E)$ be a tree, where V and E are the *tree nodes* and *tree edges*, respectively, and let $L(T)$ denote its leaf set.

Further, let X be a set of taxa (species). Then, T is a phylogenetic tree over X if there is a bijection between X and $L(T)$. Henceforth, we will identify the taxa set with the leaves they are mapped to, and let $[n] = \{1, \dots, n\}$ denote the set of leaf-labels. A tree T is said to be *rooted* if the set of edges E is directed and there is a single distinguished internal vertex r with in-degree 0. We denote by T_v the subtree rooted at v induced by the tree edges. A function $\lambda : [n] \rightarrow \{0, 1, \dots, \Sigma - 1\}$ is called a *state assignment function* over the alphabet Σ for T . We say that function $\hat{\lambda} : V(T) \rightarrow \{0, 1, \dots, \Sigma - 1\}$ is an extension of λ on T if it agrees with λ on the leaves of T . In a similar way, we define a function $\lambda^k : [n] \rightarrow \{0, 1, \dots, \Sigma - 1\}^k$ and an extension $\hat{\lambda}^k : V(T) \rightarrow \{0, 1, \dots, \Sigma - 1\}^k$. The latter function is called a *labeling* of T . We write $\hat{\lambda}^k(v) = s$ to denote that sequence s is the label of the vertex v . Every position $1 \leq i \leq k$ denotes a site in the sequence. Given a labeling $\hat{\lambda}^k$, let $d_e(\hat{\lambda}^k)$ denote the hamming distance between the two sequences labeling the two endpoints of the edge $e \in E(T)$.

A phylogenetic network $N = N(T) = (V', E')$ over the taxa set X is derived from T by splitting two edges $e, e' \in E$, adding a new vertex in each of them, and joining the two new vertices with a directed *reticulation edge*. A tree edge can take part in more than one reticulation event. Phylogenetic networks must satisfy additional temporal constraints [19]. Finally, we denote by $T(N)$ the set of all trees contained inside network N . Each such tree is obtained by the following two steps: (1) for each node of in-degree 2, remove one of the incoming edges, and then (2) for every node x of in-degree and out-degree 1, whose parent is u and child is v , remove node x and its two adjacent edges, and add a new edge from u to v . For a network N and a node $v \in V(N)$, let N_v denote the graph induced by the nodes reachable from v .

2.2 Parsimony of Phylogenetic Networks

We begin by reviewing the parsimony criterion for phylogenetic trees.

Problem 1. *Parsimony Score of Phylogenetic Trees (PSPT)*
Input: A 3-tuple (S, T, λ^k) , where T is a phylogenetic tree and λ^k is the labeling of $L(T)$ by the sequences in S .

Output: The extension $\hat{\lambda}^k$ that minimizes the expression $\sum_{e \in E(T)} d_e(\hat{\lambda}^k)$.

We define the parsimony score for (S, T, λ^k) , $\text{pars}(S, T, \lambda^k)$, as the value of this sum, and $\text{pars}(S, T, \lambda^k, i)$ as the value of this sum for site i only. In other words, $\text{pars}(S, T, \lambda^k) = \sum_{1 \leq i \leq k} \text{pars}(S, T, \lambda^k, i)$. Problem 1 has a polynomial time dynamic programming type algorithm originally devised for binary characters and binary trees by Fitch [6], and later extended to arbitrary degree trees and multi-state characters by Sankoff [23]. The algorithm finds an optimal assignment (i.e., $\hat{\lambda}^k$) for each site separately.

Since Fitch's algorithm is a basic building block in this paper, we hereby describe it. As mentioned above, the input to the problem is a tree T and a single character $C = \lambda^1$. The

algorithm finds $\text{pars}(\{1, 0\}, T, C)$, the optimal assignment to internal nodes of T , in two phases: (1) assigning values to internal nodes in a bottom-up fashion, and (2) eliminating the values determined in the previous phase in a top-down fashion. Specifically, phase (1) proceeds as follows: for a node v with children v_1 and v_2 whose values $A(v_1)$ and $A(v_2)$ have been determined,

$$A(v) = \begin{cases} A(v_1) \cap A(v_2) & \text{if } A(v_1) \cap A(v_2) \neq \emptyset \\ A(v_1) \cup A(v_2) & \text{otherwise.} \end{cases}$$

Phase (2) proceeds as follows: for a node v whose parent $f(v)$ has already been processed:

$$B(v) = \begin{cases} \sigma \in A(v) \cap A(f(v)) & \text{if } A(v) \cap A(f(v)) \neq \emptyset \\ \sigma \in A(v) & \text{otherwise.} \end{cases}$$

The algorithm above applies for binary trees. A straightforward extension to arbitrary k -degree trees is achieved if in phase (2) at each node v , $B(v)$ is a state σ which is in majority among all $A(v_i)$ for all children i and the ancestor of v . We now prove a lemma that will be useful later.

Lemma 1. *Let T be a tree and C a single character over the alphabet Σ . Let x be the number of internal nodes v s.t. $|A(v)| > 1$ by applying Fitch's algorithm on (T, C) . Then x is less than twice S^* —the parsimony score of T over C .*

As explained in Section 1 and illustrated in Fig. 1, when an HGT event occurs, the evolutionary history of the complete genomes of the organisms is modeled by a phylogenetic network. Nevertheless, the evolutionary history of every site in these genomes is modeled by one of the phylogenetic trees inside the network. This gives rise to the following definition of the parsimony score of phylogenetic networks (as was introduced by [10, 11] and formalized by [20]).

Definition 1. *Parsimony Score of Phylogenetic Networks (PSPN)*

Input: A 3-tuple (S, N, λ^k) , where N is a phylogenetic network and λ^k is the labeling of $L(N)$ by the sequences in S .

Output: The extension $\hat{\lambda}^k$ that minimizes the expression $\sum_{1 \leq i \leq k} [\min_{T \in \mathcal{T}(N)} \text{pars}(S, T, \lambda^k, i)]$.

The parsimony score of a network is the value of this sum. In the next section, we prove that the PSPN problem is NP-hard. Notice that based on Definition 1, the parsimony of each site is computed independently of the other sites, and hence we focus on the case of a single site.

3 THE PSPN PROBLEM

3.1 Hardness of the Problem

In the same spirit of MP heuristics for phylogenetic trees, a crucial part of heuristics for solving the MP problem on phylogenetic networks involves solving the PSPN problem. The decision version of the problem for the case of a single binary site is defined as follows.

Problem 2. (PSPN1)

Input: A phylogenetic network $N = N(T) = (V', E')$ with binary labeling of length 1, and an integer P

Question: Is the MP score of the network $\leq P$?

We prove the hardness of the PSPN1 problem by a reduction from the Maximum 2-Satisfiability (max-2-sat) problem [7], which is formally defined as follows.

Problem 3. Maximum 2-Satisfiability (max-2-sat)

Input: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 2$, and a positive integer $K \leq |C|$.

Question: Is there a truth assignment for U that simultaneously satisfies at least K of the clauses in C ?

We start with a lemma which will be used in our main proof. Let a “True-True” denote a clause that has no negated literals, “True-False” denote a clause that has exactly one negated literal, and “False-False” denote a clause in which both literals are negated. For each of these three types of clauses, we generate subnetworks as shown in Figs. 2(a), 2(b), and 2(c).

Lemma 2. (1) *An optimal parsimony score of 3 for a “True-True” network is obtained by labeling $x = 1, y = 1$, or both. Otherwise, the best parsimony score is 4.*

(2) *An optimal parsimony score of 3 for a “True-False” network is obtained by labeling $x = 0, y = 1$, or both. Otherwise, the best parsimony score is 4.*

(3) *An optimal parsimony score of 3 for a “False-False” network is obtained by labeling $x = 0, y = 0$, or both. Otherwise, the best parsimony score is 4.*

We are now in position to prove the main theorem.

Theorem 1. *The PSPN1 problem is NP-hard.*

Lemma 3. *Max-2-sat is hard even for inputs where each variable is restricted to appear at most 12 times.*

Corollary 1. *The PSPN1 problem is NP-hard even for networks of bounded degrees, where each node has at most 12 children.*

If $\text{gap-max-2sat}[P1, P2]$ (see [12] for the definition of gap problems) is NP-hard, then by our reduction $\text{gap-PSPN1}[4*|C| - P2 + |U|, 4*|C| - P1 + |U|]$ is NP-hard. By [9] there is a constant ζ such that there is no polynomial time algorithm for max-2-sat with performance ratio better than ζ . Thus there is such a constant also for PSPN1.

Corollary 2. *There is a constant ζ' such that there is no polynomial time algorithm for PSPN1 with performance ratio better than ζ' .*

Corollary 3. *The PSPN1 problem is hard to approximate even for networks of bounded degrees, where each node has at most 20 children.*

This result follows from the fact that gap-max-3sat , when every variable appears 5 times, is hard.

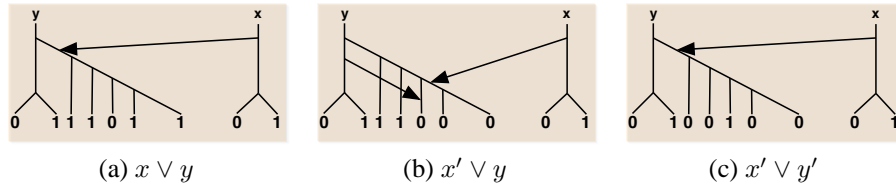


Fig. 2. Part of the reduction from max-2-sat to PSPN1.

3.2 An Improved FPT Algorithm

Definition 2. A reticulation edge $(u \rightarrow v)$ is called a lowest reticulation edge (or just a lowest edge) if there is no reticulation edge adjacent to any node in either T_u or T_v .

Lemma 4. For every phylogenetic network, there exists a lowest edge.

This lemma follows from the fact that phylogenetic networks are acyclic and satisfy additional temporal constraints [19].

Corollary 4. Let $(u \rightarrow v)$ be a lowest edge. Then both N_u and N_v are trees.

The algorithm Net2Trees of [20] enumerates all the 2^B possible trees contained inside a given network with B reticulation edges, and calculates the parsimony score of each tree by running Fitch’s algorithm [6] in $O(n|\Sigma|)$ time. The optimal score among all trees contained inside the network is then returned. The total running time is $2^B \cdot O(n|\Sigma|)$, which, for a fixed B is polynomial. However, a very simple example demonstrates that this running time can be unnecessarily extremely high. Consider a site with a single observed state. Obviously, the underlying tree yields the optimal assignment with score zero. In contrast the naïve algorithm of [20] will run in time exponential in B (and in n in a worst-case scenario).

We now present our improvement to the algorithm from [20] for computing the optimal score of a network N . By Lemma 4, there exists a lowest edge $e = (u \rightarrow v)$ in N and by Corollary 4 the subnetworks reachable from both endpoints u and v are trees. Therefore we can compute $A(u)$ and $A(v)$ by Fitch algorithm. The following lemma is fundamental for the algorithm correctness:

Lemma 5. Let $e = (u \rightarrow v)$ be a lowest edge in a network N for which $A(u)$ and $A(v)$ have already been computed. Also assume the resulting tree contains e . Then,

1. If $A(u) \subseteq A(v)$, there will be no mutation on e .
2. If $A(u) \cap A(v) = \emptyset$, there will be mutation on e .

In the cases which are not covered by Lemma 5, we say that $u \rightarrow v$ is *uncertain*. Lemma 5 gives rise to the recursive algorithm, PSPN(N), for computing the optimal score of N , as outlined in Fig. 3.

The correctness of the algorithm is implied by the construction and Lemma 5. A reticulation edge $(u \rightarrow v)$ is

PSPN($N=(V',E')$)

1. If N is not a tree
 - a. Find a lowest reticulation edge $e = (u \rightarrow v)$ in N ;
 - b. Let e' be the edge between v and its ancestral node on the tree edge;
 - c. By Fitch’s algorithm, compute the optimal assignment A of u and v ;
 - d. If $A(u) \cap A(v) = \emptyset$ then
 $opt = PSPN(V', E' \setminus e)$;
 - e. else if $A(u) \subseteq A(v)$ then
 $opt = PSPN(V', E' \setminus e')$;
 - f. else
 - (1) $opt = PSPN(V', E' \setminus e)$;
 - (2) $opt' = PSPN(V', E' \setminus e')$;
 - (3) if $opt' < opt$ then $opt \leftarrow opt'$;
 - g. return opt ;
2. else
 - a. Let T be the resulting tree;
 - b. return $Fitch(T)$;

Fig. 3. The improved FPT algorithm for the PSPN problem.

automatically taken into the tree only if it yields no mutation and automatically rejected from the tree if it necessarily leads to a mutation. In all other cases u is uncertain, and both cases are considered.

It is clear that the algorithm recurs only on reticulation edges $(u \rightarrow v)$ where u is uncertain. Necessarily, $|A(u)| > 1$ and by Lemma 1, the number of such nodes is at most twice the optimal score of N .

Theorem 2. The running time of the improved FPT algorithm is $O(n \cdot 2^{opt(N)})$, where n is the number of nodes in the network and $opt(N)$ is the optimal parsimony score of the site (under consideration) on the network.

4 THE FTMPN PROBLEM

Finally, we consider the *fixed-tree MP on phylogenetic networks* (FTMPN) problem [20]. In this problem, given an organismal (species) tree, the objective is to compute an additional set of edges whose addition to the tree yields a phylogenetic network that explains the horizontal gene transfer events which occurred during the evolutionary history of the sequences. This problem arises in situations when the underlying organismal tree is known. For example, Lerat et al. [15] reported a well-supported organismal phylogeny reconstructed from about 100 “core” genes in γ -Proteobacteria. Completing this phylogenetic tree into a network based on the whole

genomes of these organisms amounts to detecting HGT events that occurred in the γ -Proteobacteria group.

Since the actual number of the HGT events as well as their locations are not known, parsimony is used as the optimality criterion for the search. [20] showed that solving this problem accurately detects the HGT events in a sequence dataset. However, since their goal was to study the quality of the approach rather than the efficiency of computing it, they had a brute-force implementation that took almost ten hours on datasets with only two HGT events. Since this is infeasible in practice, we devise simple, yet efficient and accurate, heuristics for solving the FTMPN problem and demonstrate, through simulations, its excellent accuracy.

The preliminary results in [20] showed the optimal phylogenetic networks with k reticulation edges could always be obtained from the optimal phylogenetic networks with $k - 1$ reticulation edges. Based on this observation, we implemented a branch and bound heuristic (B&B) in which at each step of the search only “best” networks are retained. Further, to find the optimal phylogenetic networks with k reticulation edges, we conduct search based only on the optimal ones with $k - 1$ reticulation edges. This cuts the time significantly, while maintaining excellent accuracy (in terms of the optimality of the score computed by the heuristic compared to that of the model network), as will be shown in the next section.

To gain further improvements in time, we extended the B&B heuristic by inspecting Hamming distances on the tree edges; we call this heuristic B&B(Hamming). This heuristic divides the sequences (that label the nodes of the species tree) into blocks. Then, the heuristic applies Fitch’s algorithm and labels the internal nodes of the tree. Next, for each edge, it computes the Hamming distance for each of the blocks, and normalizes it by the average Hamming distance over all blocks along the same edge. Finally, for each edge we compute the difference between the maximum and minimum values of normalized Hamming distances over all blocks, and use this value as a criterion for finding candidate edges. Finally, the search for tree edges among which to add HGT events is done in the (reduced) space of candidate edges. The rationale behind this approach is that for DNA segments that were horizontally transferred (rather than inherited down the species tree), the parsimony score on the species tree should be higher than that of segments that evolved down the species tree. The reason for this is that the species tree does not model the evolution of horizontally transferred DNA segments, and hence that tree should not be a “good” model for these segments (which translates into high parsimony scores). The B&B(Hamming) achieves further reductions in time and still, while maintaining the accuracy of the search.

5 EMPIRICAL PERFORMANCE

5.1 Data

For the biological data, we considered a 15-taxon dataset of plastids, cyanobacteria, and proteobacteria, which is

a subset of the dataset considered by Delwiche and Palmer [1] and for which multiple HGT events were conjectured by the authors. The 15-taxon *rbcL* dataset consists of two sequences from the proteobacteria group, two from cyanobacteria, one from green plastids, one from red plastids, one cyanophora, and four Form II *rbcL* sequences. For this dataset, we obtained the species tree which was reported in [1], and analyzed the *rbcL* gene of these 15 organisms. The gene dataset consists of 15 aligned amino acid sequences, each of length 532 (the alignment is available from <http://www.life.umd.edu/labs/delwiche/alignments/rbcLgb7-95.distrib.txt>).

For the synthetic data, we used the following protocol to generate them. We used the *r8s* tool [22] to generate a random birth-death phylogenetic tree on 20 taxa. The *r8s* tool generates molecular clock trees; we deviated the tree from this hypothesis by multiplying each edge in the tree by a number randomly drawn from an exponential distribution. The expected evolutionary diameter (longest path between any two leaves in the tree) is 0.2. We then generated five model phylogenetic networks by adding 1, 2, 3, 4, and 5 reticulation edges (simulating HGT events) to the model tree. For each of the five phylogenetic networks, we used the Seq-gen tool [21] to evolve 26 datasets of DNA sequences of length 1500 down the organismal tree and DNA sequences of length 500 down the other tree contained inside the network (the one that exhibits all HGT events). Both sequence datasets were evolved under the K2P+ γ model of evolution, with shape parameter 1 [14]. Finally, we concatenated the two datasets.

5.2 Methods

To analyze the data, we have implemented the B&B as well as the B&B(Hamming) heuristics for solving the FTMPN problem. As these two heuristics entail scoring the parsimony of a phylogenetic network, we have implemented the naïve algorithm, introduced in [20] and referred to as “FPT” in the results section, as well as the new improved one, described in Section 3.2 and referred to as “Improved FPT” in the results section, and compared their performance in terms of time.

In our analysis, we aimed to investigate two main questions: (1) How do the new heuristics for solving the FTMPN problem perform with respect to identifying the correct number of HGT events as well as their actual locations on the organismal trees? (2) How does the Improved FPT algorithm perform, in terms of actual running time, compared to the FPT algorithm?

In both the biological as well as synthetic data, the organismal trees were known. For the biological data, we compared our results against the HGT events conjectured by the authors, and for the simulated data, we compared our results against the (known) correct solutions.

5.3 Results and Analysis

Biological data. Fig. 4(a) shows the parsimony scores of the most parsimonious networks with different number of horizontal gene transfer edges. We computed the weighted

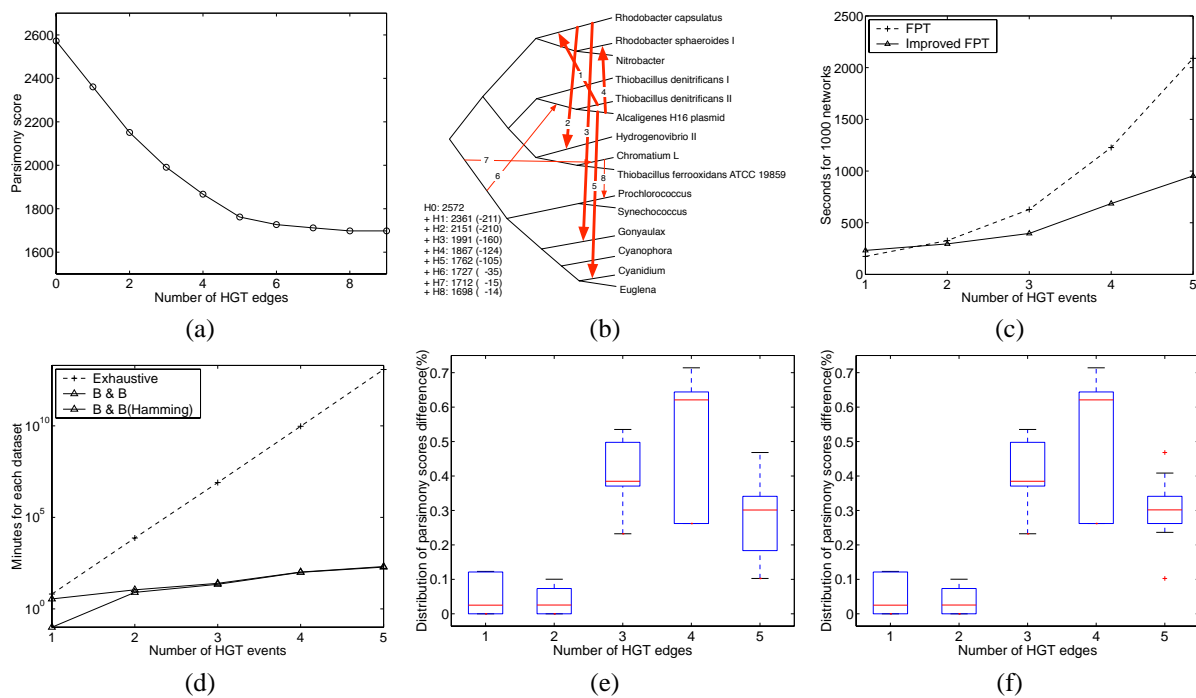


Fig. 4. (a) and (b) show results on the biological data set, whereas (c)—(f) show results on the synthetic data sets. (a) The improvement in the parsimony score as more HGT edges are added to the 15-taxon organismal tree. (b) The phylogenetic network obtained by adding the HGT edges that led to the best improvement in the parsimony score. (c) The actual computational time (averaged over all 26 runs) of the naïve FPT algorithm of [20] and our improved FPT algorithm for solving the PSPN problem, as a function of the number of HGT edges in the network. (d) actual computational time (averaged over all 26 runs) taken by the heuristics and the exhaustive search method (on a log scale); these times were taken to solve the FTMPN problem using the three methods. (e) and (f): percent difference in parsimony scores (for all runs) between optimal networks computed by the heuristics and the model networks, shown with whisker-and-box plots.

parsimony scores, using five different amino acid substitution matrices: PAM120, PAM250, BLOSUM45, BLOSUM62, and IDENTITY. The results based on these five matrices were almost identical, and due to space constraints we show only the results obtained using the IDENTITY matrix. Fig. 4(a) shows clearly that parsimony scores drop dramatically when the first 5 or 6 potential HGT edges are added to the species tree. The decrease then becomes insignificant, and no decrease at all is achieved after adding the eighth edge. The edges that resulted in the optimal decrease are shown by directed edges positioned on the species tree in Fig. 4(b), with each of the edges representing a potential transfer of the *rbcL* gene (the numbers associated with the directed edges represent the order in which they were added). Fig. 4(b) also listed the parsimony score of the most parsimonious network after adding each of the HGT edges. Row “+*Hi*” corresponds to the network after adding i^{th} HGT edge into the existing network, while “*H0*” represents the original species tree. The score changes from the previous networks are given inside the parentheses. It is clear that among the 15 taxa, the first five HGT edges are significant, while the others do not result in a significant improvement in the parsimony score, if any at all. The first three HGT edges group the Form II Rubisco together, and separates them from the rest (Form I Rubisco). The other two

HGT edges are placed between *Cyanidium*, a Red Plastid, and one of the proteobacteria, and between *Alcaligenes H16 plasmid* and *Rhodospirillum rubrum I*. These two HGT edges place the two proteobacteria close to the red plastid. These five HGT edges were the ones conjectured by the authors in [1].

Synthetic data. In the first set of experiments, we compared the performances (in terms of actual computational time) of the naïve FPT algorithm [20] and our new improved FPT algorithm. The results are summarized in Fig. 4(c). The results show that except for the case of a single HGT event, the improved FPT algorithm becomes much faster than the naïve one as the number of HGT edges increases. In particular, for the case of 5 HGT events, the improvement is larger than a factor of 2. More importantly, as the number of HGT events increases, the improvement becomes much more significant (indicated by the widening gap between the two curves in Fig. 4(c)). In the second set of experiments, we studied the performance of the two heuristics B&B and B&B(Hamming) for solving the FTMPN problem. We compared the times taken by these heuristics to the time the exhaustive search of [20] would take; we had to estimate this latter time, since it would take probably years to perform an exhaustive search on all networks with more than two HGT events. Further, we compared the parsimony scores of the optimal networks computed by these

heuristics, PSI , to the parsimony score of the model network, PSM , by the formula: $(PSM - PSI)/PSM$ %. This is the value referred to as *parsimony score difference*(%) in Fig. 4(e) and 4(f). Fig. 4(d) shows drastic improvements in the time achieved by the two heuristics. The exact times in minutes using the B&B heuristic for the networks with 1, 2, 3, 4, and 5 HGT events are 3.5, 11, 25, 103, and 206, respectively. The exact times in minutes using the B&B(Hamming) heuristic for the networks with 1, 2, 3, 4, and 5 HGT events are 0.1, 8, 22, 100, 192, respectively. On the other hand, the estimated time in minutes using an exhaustive search in the network space are 6.5 , 7.3×10^3 , 8.0×10^6 , 9.5×10^9 , and 12.0×10^{12} , respectively. Equally important, the improvement was achieved while maintaining high accuracy in the parsimony scores computed, which is reflected in the negligible score differences plotted in Figs. 4(e) and 4(f). The Figs. show that the parsimony scores of the networks inferred by the heuristics fall within 0.7% of the parsimony scores of the model networks. This is a very high accuracy.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the parsimony criterion for phylogenetic networks. We settled the computational complexity question of the PSPN problem, proving it is NP-hard and hard to approximate. Further, we devised an efficient algorithm for solving the problem. We also designed efficient heuristics for the FTMPNN problem. We implemented our methods and tested them on biological as well as synthetic data. Our results are very promising and provide a significant contribution towards bringing the tools and methodologies for reconstructing and evaluating phylogenetic networks on a par with those for phylogenetic trees. In this paper, we provided the first experimental analysis of the parsimony criterion for networks on both synthetic as well as biological data. For future work, we intend to analyze others groups of prokaryotic organisms, establish the complexity of the FTMPNN problem, and design efficient solutions for it.

7 ACKNOWLEDGMENTS

This work was supported in part by the Rice Terascale Cluster funded by NSF under grant EIA-0216467, Intel, and HP. Sagi Snir was supported in part by NSF grant CCR-0105533.

REFERENCES

- [1]C. F. Delwiche and J. D. Palmer. Rampant horizontal transfer and duplication of rubisco genes in eubacteria and plastids. *Mol. Biol. Evol.*, 13(6), 1996.
- [2]W.F. Doolittle, Y. Boucher, C.L. Nesbo, C.J. Douady, J.O. Andersson, and A.J. Roger. How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Phil. Trans. R. Soc. Lond. B. Biol. Sci.*, 358:39–57, 2003.
- [3]R.G. Downey and M.R. Fellows. Fixed parameter tractability and completeness I: basic theory. *SIAM Journal of Computing*, 24:873–921, 1995.
- [4]J.A. Eisen. Assessing evolutionary relationships among microbes from whole-genome analysis. *Curr. Opin. Microbiol.*, 3:475–480, 2000.
- [5]I.T. Paulsen *et al.* Role of mobile DNA in the evolution of Vacuolysin-resistant *Enterococcus faecalis*. *Science*, 299(5615):2071–2074, 2003.
- [6]W. Fitch. Toward defining the course of evolution: minimum change for a specified tree topology. *Syst. Zool.*, 20:406–416, 1971.
- [7]M. R. Garey and D. S. Johnson. *Computer and Intractability*. Bell Telephone Laboratories, incorporated, 1979.
- [8]D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19–28, 1991.
- [9]J. Hastad. Some optimal inapproximability results. *STOC97*, pages 1–10, 1997.
- [10]J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98:185–200, 1990.
- [11]J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, 36:396–405, 1993.
- [12]D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.
- [13]D. H. Huson and D. Bryant. Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol.*, 23(2):254–267, 2006.
- [14]M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980.
- [15]E. Lerat, V. Daubin, and N.A. Moran. From gene trees to organismal phylogeny in prokaryotes: The case of the γ -proteobacteria. *PLoS Biology*, 1(1):1–9, 2003.
- [16]C.R. Linder, B.M.E. Moret, L. Nakhleh, and T. Warnow. Network (reticulate) evolution: biology, models, and algorithms. In *The Ninth Pacific Symposium on Biocomputing (PSB)*, 2004. A tutorial.
- [17]C.R. Linder and L.H. Rieseberg. Reconstructing patterns of reticulate evolution in plants. *American Journal of Botany*, 91:1700–1708, 2004.
- [18]V. Makarenkov, D. Kevorkov, and P. Legendre. Phylogenetic network reconstruction approaches. *Applied Mycology and Biotechnology (Genes, Genomics and Bioinformatics)*, 6, 2005. To appear.
- [19]B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme. Phylogenetic networks: modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):13–23, 2004.
- [20]L. Nakhleh, G. Jin, F. Zhao, and J. Mellor-Crummey. Reconstructing phylogenetic networks using maximum parsimony. *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, 393:440–442, Jun 2005.
- [21]A. Rambaut and N. C. Grassly. Seq-gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comp. Appl. Biosci.*, 13:235–238, 1997.
- [22]M. Sanderson. *r8s* software package. Available from <http://loco.ucdavis.edu/r8s/r8s.html>.
- [23]D. Sankoff. Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28:35–42, 1975.