SIAM J. Comput.
Vol. 30, No. 3, pp. 729–752

FROM GENE TREES TO SPECIES TREES[*]

BIN MA[†], MING LI[‡], AND LOUXIN ZHANG[§]

Abstract. This paper studies various algorithmic issues in reconstructing a species tree from gene trees under the duplication and the mutation cost model. This is a fundamental problem in computational molecular biology. Our main results are as follows.

1. A linear time algorithm is presented for computing all the losses in duplications associated with the least common ancestor mapping from a gene tree to a species tree. This answers a problem raised recently by Eulenstein, Mirkin, and Vingron [*J. Comput. Bio.*, 5 (1998), pp. 135–148].
2. The complexity of finding an optimal species tree from gene trees is studied. The problem is proved to be NP-hard for the duplication cost and for the mutation cost. Further, the concept of reconciled trees was introduced by Goodman et al. and formalized by Page for visualizing the relationship between gene and species trees. We show that constructing an optimal reconciled tree for gene trees is also NP-hard. Finally, we consider a general reconstruction problem and show it to be NP-hard even for the well-known nearest neighbor interchange distance.
3. A new and efficiently computable metric is defined based on the duplication cost. We show that the problem of finding an optimal species tree from gene trees is NP-hard under this new metric but it can be approximated within factor 2 in polynomial time. Using this approximation result, we propose a heuristic method for finding a species tree from gene trees with uniquely labeled leaves under the duplication cost. Our experimental tests demonstrate that when the number of species is larger than 15 and gene trees are close to each other, our heuristic method is significantly better than the existing program in Page's GeneTree 1.0 that starts the search from a random tree.

Key words. gene trees, species trees, NP-hardness, algorithms

AMS subject classifications. 68Q, 68W, 92B

PII. S0097539798343362

1. Introduction.

As DNA sequences become easier to obtain, in the field of evolutionary moleculor biology emphasis has been placed on constructing gene trees and, from the gene trees, reconstructing evolutionary trees for species (called *species trees*) [9, 11, 20]. The current strategy for reconstructing species trees is based on the separate consideration of distinct gene families represented by homologous sequences; these homologous sequences are assumed to evolve in the same way as species. However, because of the presence of paralogy, sorting of ancestral polymorphism, and horizontal transfer, gene trees and species trees are often inconsistent [21, 25, 31, 33] and a "correct" species tree may simply not exist. Hence, a fundamental problem that arises is how to reconcile different, sometimes contradictory, gene trees into a

[*] Received by the editors August 11, 1998; accepted for publication October 29, 1999; published electronically July 13, 2000. An extended abstract of this work was presented in the *Second Annual International Conference on Computational Molecular Biology*, New York, 1998, pp. 182–191.

http://www.siam.org/journals/sicomp/30-3/34336.html

[†] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong (csbma@cityu.edu.hk). The research of this author was supported in part by HK RGC grants 9040297 and 9040352 and CityU Strategic grant 7000693.

[‡] Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (mli@math.uwaterloo.ca). The research of this author was supported in part by NSERC Operating grant OGP0046506, a CGAT grant, and a Steacie Fellowship. Part of this work was done at City University of Hong Kong.

[§] BioInformatics Center, Kent Ridge Digital Labs, 21 Heng Mui Keng Terrace, Singapore 119613 (lxzhang@krdl.org.sg). Current address: Department of Mathematics, National University of Singapore, Singapore 117543.

species tree [10]. This problem has been studied extensively for the last two decades. Several similarity/dissimilarity measures for gene trees and species trees have been proposed and efficient comparison methods have also been investigated. See, for example, [8, 14, 15, 18, 32].

This paper studies the problem of reconciling different gene trees into a species tree under two well-known duplication-based similarity measures. These measures were proposed by Goodman et al. [14], Page [23], and Guigó, Muchnik, and Smith [15]. Genes have *gene trees* because of gene replication. As a gene copy at a locus in the whole genome replicates and its copies are passed onto offsprings, branch points are generated. Since the gene copy has a single ancestral copy, the resulting history is a branching tree. Gene divergence causes all the inconsistencies among different gene trees. Such divergence can be the result of either speciation or duplication events [22]. If the common ancestry of two genes can be tracked back to a speciation event, then they are said to be related by *orthology*; if it is tracked back to a duplication event, then they are related by *paralogy* [10]. Taking orthology and paralogy into account, Goodman et al. proposed a similarity measure for annotating species trees with duplications, gene losses, and nucleotide replacements [14]. Later, Page developed a method based only on duplications for interpreting inconsistency between vertebrate globin gene trees and the species tree that is constructed from morphological data [23]; Guigó, Muchnik, and Smith elaborated the idea for locating the gene duplications in eukaryotic history [15].

A *species tree* can be defined as the pattern of branching of species lineages via the process of speciation. When species are split by speciation, the gene copies within species likewise are split into separate bundles of descent. Thus, gene trees are contained within species trees. However, a gene tree may disagree with the containing species tree because of the reasons mentioned above. The duplication and mutation costs were defined using a least common ancestor (LCA) mapping M from gene trees to a species tree. Assume that only genes from each contemporary species are presented in gene trees. In a gene tree, leaves denote the genes from contemporary species; internal nodes are considered as ancestral genes. We may think that an ancestral gene is uniquely determined by the subset of contemporary genes descending from it in the gene tree. Similarly, in a species tree, an internal node is considered as an ancient species (which might not exist today) and is determined by the contemporary species descending from it. We may denote a contemporary species and the genes from that species by the same label. The mapping M from a gene tree to a species tree just maps a contemporary gene to the corresponding species, and an ancestral one to the most recent species which contains that gene (as a subset). Hence, we call it the *LCA mapping* in this paper. When the gene and species trees are inconsistent, it maps an ancestral gene g, and its child gene $c(g)$ to the same ancient species. In this case, we say that a *duplication* happens at g. Furthermore, roughly speaking, the number of gene losses associated with g is defined as the total number of ancient species between $M(g)$ and $M(c(g))$ for all children $c(g)$. To measure the similarity between a gene and species trees, Page defined the duplication cost as the number of duplications, and Guigó, Muchnik, and Smith defined the mutation cost as the sum of the number of duplications and the number of gene losses [15]. The mutation cost is not only efficiently computable, as shown in [4] and [34] independently (see also [5]), but also biologically meaningful [18]. Reconstructing a global species tree is based on the parsimonious criterion of minimizing the concerned cost between the gene trees and the species tree. In their paper [15], Guigó, Muchnik, and Smith developed a heuristic method for the problem using a nearest neighbor interchange (NNI) search algorithm and applied it to infer the most likely phylogenetic relationship among 16 major eukaryotic taxa from the sequences of 53 different genes. In spite of having several serious flaws, their work demonstrated the potential of these measures in studies of genome evolution.

The contributions of this paper are in three aspects. First, we study the properties of the LCA mapping as well as the duplication and mutation costs. In particular, we prove a less obvious fact that the duplication cost satisfies the triangle inequality (Lemma 5.1) and study the relation between the duplication cost and the best-known NNI distance. We also present a linear time algorithm for computing all the losses in all duplications (section 3). Secondly, the complexity of reconstructing a species tree from gene trees is investigated. We prove that the problem is NP-hard under the duplication cost and under the mutation cost. The concept of a reconciled tree was introduced by Goodman et al. [14] for studying hemoglobin gene phylogeny, where there were significant discrepancies between gene and organismal phylogenies; later it was formalized by Page [23] as a means of describing historical associations such as those between genes and species. We prove that finding the best reconciled tree from a gene tree is NP-hard. We also consider a general reconstruction problem and prove it to be NP-hard even for the NNI distance. These results justify the necessity of developing heuristic methods and experimental research for reconstructing species trees [15, 24]. Finally, we give a heuristic method for reconstructing species trees. To this end, we propose a new and efficiently computable metric, satisfying the metric axioms, based on the duplication cost. Under this new metric, we show that the problem of reconstructing a species tree from gene trees is NP-hard but can be approximated within factor 2 in polynomial time. Using this approximation result, we present a new heuristic method for reconstructing species trees from uniquely leaf-labeled gene trees under the duplication cost.

The rest of the paper is divided into six sections. In section 2, we define the concepts of gene duplications and losses, review the duplication cost and the mutation cost and their basic properties, and formalize three problems of reconstructing a species tree from gene trees. In section 3, we present a linear time algorithm for computing all the losses between a gene tree and a species tree. In section 4, we prove that the problems defined in section 2 are NP-hard. In section 5, a new metric is proposed based on the duplication cost. We prove that, under the new metric, reconstructing a species tree from gene trees is NP-hard but can be approximated within factor 2 in polynomial time. Then a new heuristic method for reconstructing species trees is proposed. Experimental results are given to demonstrate that our new heuristic works quite well. In section 6, we consider a general reconstruction problem and prove it to be NP-hard even for the popular NNI distance. In section 7, we discuss further research and open questions.

We refer the reader to [2, 13] for textbooks on NP-completeness and approximation algorithms.

2. Comparing gene and species trees: Duplications and losses.

In this section we first define the gene trees and species trees. We then introduce the two duplication-based measures for comparing gene and species trees: the duplication and mutation costs. For their biological meaning, we refer the reader to [14, 15, 23].

2.1. Species trees and gene trees.

For a set I of N biological taxa, the model for their evolutionary history is a rooted full binary tree T where there are N leaves each uniquely labeled by a taxon in I and $N-1$ unlabeled internal nodes. Here the term "full" means that each internal node has exactly two children. Such a tree is

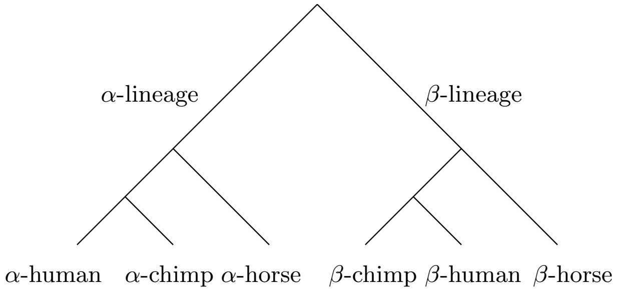


FIG. 1. *A gene tree based on α-hemoglobin and β-hemoglobin* [4].

called a *species tree*. In a species tree, we treat an internal node as a subset (called a *cluster*) which includes as its members its subordinate species represented by the leaves below it. Thus, the evolutionary relation "m is a descendant of n" is expressed using set-theoretic notation as "$m \subset n$."

The model for gene relationship is a rooted full binary tree with labeled leaves. Usually, a gene tree is constructed from a collection of genes each having several copies appearing in the studied species. For example, the gene family of hemoglobin genes in vertebrates contains α-hemoglobin and β-hemoglobin. A gene tree based on these two genes is illustrated in Figure 1 for human, chimpanzee, and horse. We use the species to label the genes appearing in it. Thus, the labels in a gene tree may not be unique. An internal node g corresponds to a multiset $\{x_1^{i_1}, x_2^{i_2}, \ldots, x_m^{i_m}\}$, where i_j is the number of its subordinate leaves labeled with x_j. The *cluster* of g is simply the set

$$S_g = \{x_1, x_2, \ldots, x_m\}.$$

Finally, we use $L(T)$ to denote the set of leaf labels in a species or gene tree T.

2.2. Gene duplications and the duplication cost. Given a gene tree G and a species tree S such that $L(G) \subseteq L(S)$. For any node $g \in G$, we define $M(g)$ to be the LCA of g in S, i.e., the smallest node $s \in S$ such that $S_g \subseteq s$. Here we used the term "smallest" to mean "farthest from the root." This correspondence M, first considered by Goodman et al. [14], is referred to as a mapping of G into S by Page [23]. We call M the *LCA mapping* from G to S. Obviously, if $g' \subset g$, then $M(g') \subseteq M(g)$, and any leaf is mapped onto a leaf with the same label. For an internal node g, we use $c(g)$ (sometimes $a(g)$ and $b(g)$) to denote a child of g and $G(g)$ the subtree rooted at g.

DEFINITION 2.1. *Let g be an internal node of G. If $M(c(g)) = M(g)$ for some child $c(g)$ of g, then we say $G(g)$ and $S(M(g))$ are root-inconsistent or a duplication happens at g.*

The total number $t_{dup}(G, S)$ of duplications happening in G under the LCA mapping M is proposed as a measure for the similarity between G and S [14, 23]. We call such a measure the *duplication cost*.

A subset A of (internal or leaf) nodes in a species tree S is *disjoint* if $x \cap y = \phi$ for any $x, y \in A$. For a disjoint subset A in S, the *restriction* of S on A is the smallest subtree of S containing A as its leaf set, denoted by $R_S(A)$. The *homomorphic subtree* $S|_A$ of S induced by A is a tree obtained from $R_S(A)$ by contracting all degree 2 nodes except the root. These concepts are illustrated in Figure 2. We state, without proofs, the following facts which will be used implicitly in the rest of this paper.

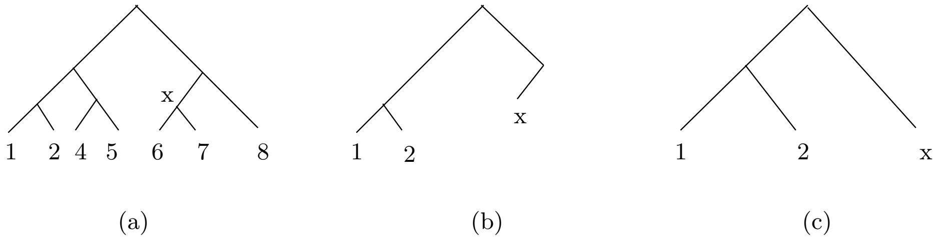


Fig. 2. (a) *A species tree S;* (b) *the restriction subtree $R_S(A)$ for $A = \{1,2,x\}$;* (c) *the homomorphic subtree $S|_A$ induced by A.*

Proposition 2.2. *Let G be a gene tree and S a species tree. Then $t_{dup}(G,S) = 0$ if and only if G is identical to $S|_{L(G)}$.*

Proposition 2.3. *Let g be the root of G with children $a(g)$ and $b(g)$ and let s be the root of S with children $a(s)$ and $b(s)$. Then, if a duplication happens at g under the LCA mapping from G to S, then $t_{dup}(G,S) = 1 + t_{dup}(a(g),S) + t_{dup}(b(g),S)$.*

Furthermore, the duplication cost also satisfies the triangle inequality, which will be proved in Lemma 5.1 in section 5. Under the duplication cost, the problem of finding the "best" species tree from a set of known gene trees can be formulated as the following minimization problem.

Optimal Species Tree I (OST I).

Instance: n gene trees $G_1, G_2, \ldots, G_n$.

Question: Find a species tree S with the minimum duplication cost $\sum_{i=1}^{n} t_{dup}(G_i, S)$.

One can easily convert the above optimization problem into its *decision version* by having an extra integer c as input and requiring the minimum duplication cost to be less than c. This comment applies to all other optimization problems in this paper.

2.3. Gene losses and the mutation cost.

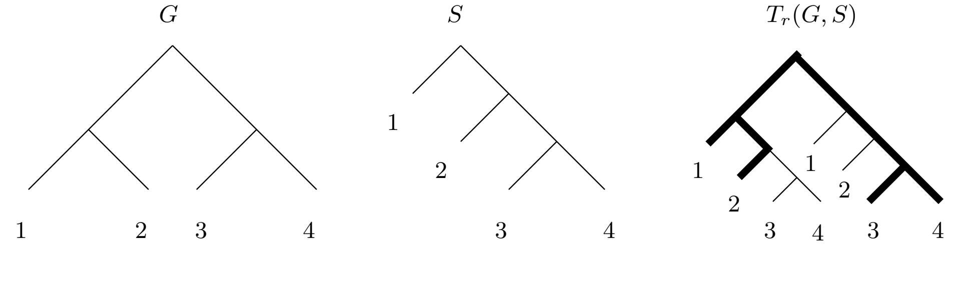
After defining the duplication cost, we now introduce the mutation cost. We first define the *number of gene losses* associated with the LCA mapping M from G to S. Since $L(G) \subseteq L(S)$, $S|_{L(G)}$ is well defined and M induces an LCA mapping M' from G to $S|_{L(G)}$. Let g and g' be two nodes in $S|_{L(G)}$ such that $g \subseteq g'$. Define

$$d(g,g') = |\{h \in S|_{L(G)} \mid M'(g) \subset h \subset M'(g')\}|.$$

Let $a(g)$ and $b(g)$ denote the two children of g. The *number of losses* l_g associated to g is

$$l_g = \begin{cases} 0 & \text{if } M'(g) = M'(a(g)) = M'(b(g)); \\ d(a(g),g) + 1 & \text{if } M'(a(g)) \subset M'(g) \;\&\; M'(g) = M'(b(g)); \\ d(a(g),g) + d(b(g),g) & \text{if } M'(a(g)) \subset M'(g) \;\&\; M'(b(g)) \subset M'(g). \end{cases}$$

Note that our definition of $l(g)$ is a *generalization* of the one defined by Guigó, Muchnik, and Smith [15]. When $L(G) = L(S)$ and gene tree G is also uniquely labeled, our definition is identical to the one defined in [15]. The *mutation cost* is defined as the sum of t_{dup} and the total number of losses $l(G,S) = \sum_{g \in G} l_g$. This measure turns out to be identical to a biologically meaningful measure defined in Mirkin, Muchnik, and Smith [18] when G has the same number of uniquely labeled leaves as S. The problem of finding the "best" species tree from a set of known gene trees under this measure is formulated as the following.



(a) (b) (c)

FIG. 3. (a) *A gene tree G;* (b) *a species tree S;* (c) *the reconciled tree $T_r(G,S)$ of G with respect to S.*

Optimal Species Tree II (OST II).
INSTANCE: n gene trees $G_1, G_2, \ldots, G_n$.
QUESTION: Find a species tree S with the minimum mutation cost $\sum_{i=1}^{n}(t_{dup}(G_i,S)+l(G_i,S))$.

2.4. Reconciled trees. For visualizing the relationship between gene and species trees, we use a third tree called the *reconciled tree* [14]. The reconciled tree has two important properties. The first property is that the observed gene tree is a subtree of the reconciled tree. The second property is that the clusters of the reconciled tree are all clusters of the species tree. Formally, the reconciled tree is defined as follows.

Let T' and T'' be two rooted trees; we use $T' \triangle T''$ to denote the rooted tree T obtained by adding a node r as the root and connecting r to $r(T')$ and $r(T'')$ so that T' and T'' are two subtrees rooted at the children of r. Further, let t be an internal node in T'; then $T'|_{\,t\rightarrow T''}$ denotes the tree formed by replacing the subtree rooted at t with T''. Similarly, $T'|_{\,t\rightarrow T_1,\, t'\rightarrow T_2}$ can be defined for disjoint nodes t and t'.

For a gene tree G rooted at g and a species tree S rooted at s such that $L(G) \subseteq L(S)$, let M be the LCA mapping from G to S and let $s' = M(a(g))$ and $s'' = M(b(g))$. The *reconciled tree* $R = R(G,S)$ of G with respect to S is defined as

$$(1) \qquad R = \begin{cases} R(G(a(g)),S) \triangle R(G(b(g)),S) & \text{if } s' = s'' = s, \\ S|_{\,s'\rightarrow R(G(a(g)),\,S(s'))} \triangle R(G(b(g)),S) & \text{if } s' \subseteq a(s),\ s'' = s, \\ S|_{\,s'\rightarrow R(G(a(g)),S(s')),\, s''\rightarrow R(G(b(g)),S(s''))} & \text{if } s' \subseteq a(s),\ s'' \subseteq b(s), \\ S|_{\,a(s)\rightarrow R(G,S(a(s)))} & \text{if } M(g) \subseteq a(s). \end{cases}$$

Such a concept is illustrated in Figure 3. An efficient algorithm was presented in [23] for computing a reconciled tree given a set of gene trees and a species trees. It is easy to see that the reconciled tree $R(G,S)$ satisfies the following three properties, of which the first two are mentioned above:

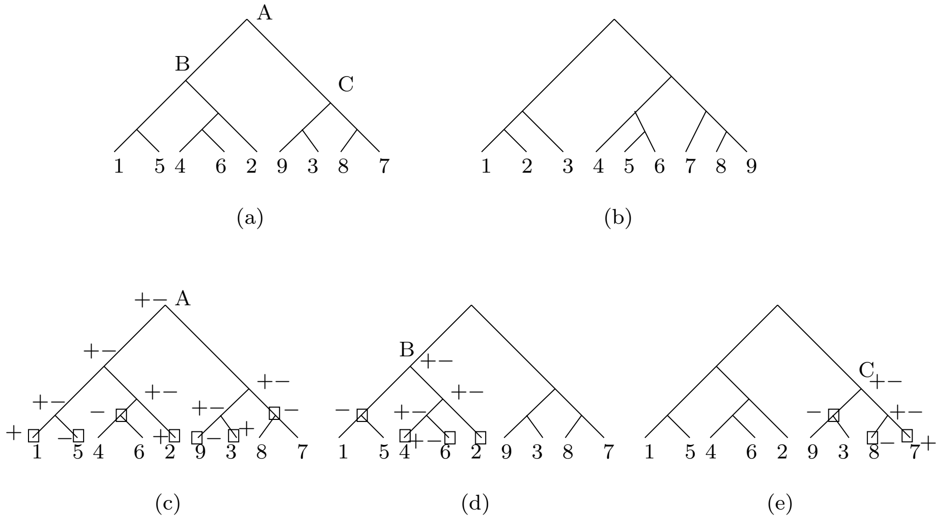
1. It contains G as a subtree, i.e., there is a subset L of leaves such that $R(G,S)|_L$ is isomorphic to G.
2. All clusters are in S, where a cluster is defined as a subset of species below an internal node in S (see subsection 2.1).
3. For any two children $a(g)$ and $b(g)$ of a node $g \in R(G,S)$, $a(g) \cap b(g) = \phi$, or $a(g) = b(g) = g$.

Actually, Page also defined the reconciled tree $R(G,S)$ as the smallest tree satisfying the above properties. However, these two definitions are not obviously equivalent. A rigorous proof of this equivalence is needed and unknown. Reconstructing a species tree from a gene tree can be formulated as the following.

Optimal Species Tree III (OST III).
INSTANCE: A gene tree G.
QUESTION: Find a species tree S with the minimum duplication cost $t_{dup}(T_r(G,S),S)$.

3. Computing all loss events. When comparing a gene tree and a species tree, one may need to know both mutation cost and all "loss events" (to be defined). It is an open problem to compute all loss events efficiently [5]. In this section, we will develop a linear time algorithm to solve the problem. In the rest of this section, we assume that both gene tree G and species tree S are uniquely leaf labeled and $L(G) = L(S)$. We first introduce the concept of the gene loss events.

Let $u \in G$ and a duplication d_u occur at u. Recall that $S(M(u))$ denotes the subtree of S below $M(u)$. A node $v \in S(M(u))$ is *mixed* in the duplication d_u if $v \cap c(u) \neq \phi$ for any child $c(u)$ of u; it is *speciated* if $v \cap a(u) \neq \phi$ but $v \cap b(u) = \phi$ or vice versa; it is *gapped* if $v \cap c(u) = \phi$ for any $c(u)$. Finally, we say that a *loss event* occurs at a maximal speciated/gapped node in d_u. Note that a unique loss event occurs at some node on the path from $M(u)$ to any leaf in S. Figure 4 presents a mapping from a gene tree G (in (b)) to a species tree S (in (a)). Three duplications occur at nodes $r(g)$, $\{4,5,6\}$, and $\{7,8,9\}$ that are shown in (c), (d), and (e), respectively, where mixed nodes are labeled with "$+-$," speciated nodes with "$+$" or "$-$" depending on which intersection is empty, and gapped nodes are not labeled. All 14 loss events are marked by square boxes.



(a) (b) (c) (d) (e)

FIG. 4. *Duplications between a species tree* (a) *and a gene tree* (b).

Formally, the problem of computing all the loss events is formulated as follows. Given a gene tree G and a species tree S such that $L(G) = L(S)$, to find for each duplication d occurring at a node $g \in G$, the subtree $S(M^{-1}(g))$ of S with all the loss events as its leaves. For example, for the gene tree and species tree illustrated in Figure 4, the output is the three subtrees shown in Figure 5.

Note that the species tree S and gene tree G are rooted. We first impose an arbitrary ordering on the children of each node and produce an in-order traversal of G and S, respectively. Recall that in an in-order traversal, $i < j$ if and only if i is in

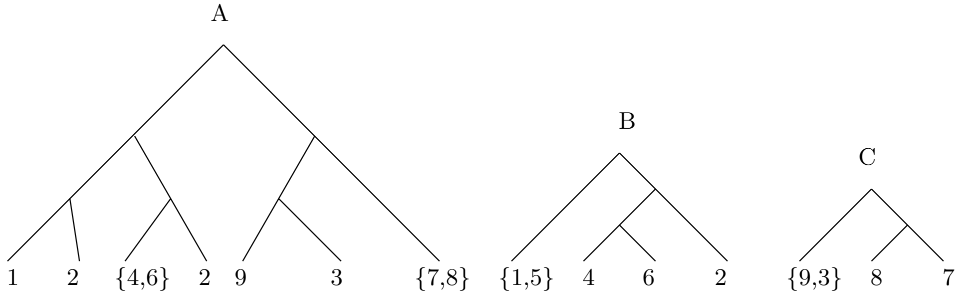


Fig. 5. *Output from computing all the loss events.*

the left subtree of j [2]. Without loss of generality, we may assume that each node of S is labeled by a number $k \leq 2n-1$, which is called the *in-order number* of the node. Preprocess the tree S in $O(n)$ steps so that an LCA query can be answered in constant time [16, 28]. Using this preprocessing, we can also compute the LCA mapping from G to S in linear time [34].

We store M in G as follows. To each node x in G, we associate a pair of $\langle i, j\rangle$, where i is its in-order number while j is the in-order number of $M(x)$ in S.

Definition 3.1 (see [34]). *Let g be an internal node of G. It is said to be type-*1 *under the LCA mapping $M : G \rightarrow S$ if $M(a(g)) \subset M(g)$ and $M(b(g)) \subset M(g)$; it is type-*2 *if $M(a(g)) \subset M(g)$ and $M(b(g)) = M(g)$ or vice versa; it is type-*3 *if $M(a(g)) = M(b(g)) = M(g)$. Recall that $a(g)$ and $b(g)$ denote the children of g.*

To each node $y \in S$, we also assign an ordered pair $\langle i, n_{23}\rangle$, where i is its in-order number and n_{23} is the number of type-2 or type-3 nodes in G that is mapped to y. Observe that duplications occur at type-2 or type-3 nodes.

For a type-1 node g_1, $M(a(g_1))$ and $M(b(g_1))$ are distinct from $M(g_1)$. The unique path from $M(a(g_1))$ to $M(b(g_1))$ through $M(g_1)$ is called an *arc* in the mapping M from G to T. For our purpose, we say that such an arc *starts* at $M(g_1)$. We also say that such an arc *passes* through any intermediate nodes between $M(a(g_1))$ and $M(g_1)$ and between $M(b(g_1))$ and $M(g_1)$. For a type-2 node g_2, assume $M(a(g_2)) \subset M(g_2)$ and $M(b(g_2)) = M(g_2)$. The unique path from $M(g_2)$ to its descendant $M(a(g_2))$ is called an *arc* in the mapping M from G to T,[1] *starting* at $M(g_2)$. Such an arc *passes* through all intermediate nodes between $M(a(g_2))$ and $M(g_2)$. To each node y in S we associate a (linked) list $A(y)$ of all arcs passing or starting from y and two integers s_y and p_y, where s_y is the number of arcs starting at y, and p_y is the number of arcs passing y.

Proposition 3.2. *The $A(y)$'s, s_y's, and p_y's can be computed in $O(l+2n)$ time, where l denotes the total number of loss events.*

Proof. We use breadth-first search starting from the root on the gene tree G. For each node $x \in G$, if $M(x) \neq M(a(x))$ and $M(x) \neq M(b(x))$, then $M(a(x)), M(b(x))$ is below $M(x)$ in S, we use the in-order numbers of $M(x)$, $M(a(x))$, and $M(b(x))$ to travel down from $M(x)$ to $M(a(x))$ and $M(b(x))$ in S, and add the arc $(a(x), b(x))$ to the list $A(y)$ and update s_y and p_y for each node y on the arc. If x is a type-2 node, let $M(x) \neq M(a(x))$ but let $M(x) = M(b(x))$. Then, we use the in-order numbers of $M(x)$, $M(a(x))$ to travel down from $M(x)$ to $M(a(x))$, during which we add the arc $(x, a(x))$ to the list $A(y)$ and update s_y and p_y for each node y on the arc.

[1] In [34], this is called a path.

Now we analyze the time complexity. For each node x, we take $O(d(M(a(x)), M(b(x))))$ in total to update the linked lists $A(y)$, the starting numbers s_y, and passing numbers p_y of nodes y on the arc. Thus, the algorithm takes

$$\begin{aligned} t &= \sum_{x\in G-L(G)} d(M(a(x)), M(b(x))) \\ &= \sum_{y\in S-L(S)} |A(y)| \\ &= n-1+\sum_{y\in Mixed(S)} |A(y)| \\ &\leq n+l+t_{dup} \\ &\leq 2n+l, \end{aligned}$$

where the third equality follows from the fact that the number of duplications in which a node y is mixed is equal to $s_y + p_y - 1$ [34], and the first inequality is based on the fact that for each duplication, the number of losses is equal to one plus the number of mixed nodes [34]. This concludes the proof. □

PROPOSITION 3.3. *Let x be an internal node in the species trees S; then a loss event occurs at x in some duplication if and only if $s_{p(x)} + p_{p(y)} - p_x - s_x + n_{23} > 0$, where n_{23} is the number of type-2 or type-3 nodes mapped to x under the LCA mapping M.*

Proof. There are exactly $s_{p(x)} + p_{p(y)} - 1$ duplications in which the parent $p(x)$ of x is a mixed node [34]. On the other hand, there are $s_x + p_x - 1$ duplications in which x is a mixed node. Further, n_{23} of these duplications occur at x. Thus, there are exactly $s_{p(x)} + p_{p(y)} - s_x - p_x + n_{23}$ duplications in which $p(x)$ is a mixed node but x is a speciation, i.e., a loss event occurs at x if $s_{p(x)} + p_{p(y)} - p_x - s_x + n_{23} > 0$. This concludes the proof. □

Thus, by Proposition 3.3, we can list all the nodes at which a loss event occurs in $O(n)$ steps by traveling down the species tree S. Moreover, we need to find out for each loss event which duplication causes it. Recall that $A(y)$ denotes the set of all arcs that pass or start at y for each node $y \in S$. Let

$$A(y) = \{(x_i, x'_i) \mid i \leq m\}$$

and let $A^{-1}(y) = \{M^{-1}x_i, M^{-1}x'_i \mid i \leq m\}$. The following proposition is a combination of Claim 1 and Claim 2 in the proof of Proposition 3.4 in [34].

PROPOSITION 3.4. *The homomorphic subtree $G|_{A_y^{-1}}$ contains all the duplication nodes z in which y is a mixed node.*

By Proposition 3.4, we have the following.

PROPOSITION 3.5. *An in-order traversal of $G|_{A_y^{-1}}$ can be computed in $O(|A^{-1}(y)|)$ steps.*

Proof. Let $|A^{-1}(y)| = k$. Then $k = O(m)$. Radix sort $A^{-1}(y)$ in $O(m)$ steps. Let $z_1, z_2, \ldots, z_k$ be the in-order list of leaves of $G|_{A_y^{-1}}$. Let $z'_{2j-1} = z_j$ and $z'_{2j} = \mathrm{LCA}(z_j, z_{j+1})$. Then $z'_1, z'_2, \ldots, z'_{2m-1}$ is the in-order traversal of $G|_{A_y^{-1}}$. □

Let $\mathrm{Dup}(G, S)$ denote the set of all duplications occurring under the LCA mapping from G to S. For each duplication $d \in \mathrm{Dup}(G, S)$, let $\mathrm{Loss}(d)$ denote the set of nodes of S on which a loss event occurs in d. For each node $y \in S_{loss} = \cup\mathrm{Loss}(d)$ on which a loss event occurs, let

$$S_y = \{d \in \mathrm{Dup(G)} \mid y \in \mathrm{Loss}(d)\}. \tag{2}$$

Since for a duplication d, a loss event occurs at a node y if and only if the parent $p(y)$ of y is a mixed node in d, but y is a speciation node. Then, by Proposition 3.4, an in-order traversal of S_y can be obtained from difference between the in-order traversal of $G|_{A^{-1}(p(y))}$ and $G|_{A^{-1}(y)}$, which takes at most $O(s_{p(y)}+p_{p_y}+s_y+p_y)$ steps. Therefore, we have the following.

PROPOSITION 3.6. *The S_y's for all nodes y can be computed in $O(l+n)$ steps.*

Proof. Do a breadth-first search for loss nodes in S. For each loss node y, we find the in-order traversals of $G|_{A^{-1}(y)}$ and $G|_{A^{-1}(p(y))}$ and then find S_y from them as described above. The complexity is

$$t = \sum_{y \in S_{loss}} (O(s_{p(y)} + p_{p(y)} + s_y + p_y)) \leq O(l + 2n).$$

This concludes the proof. □

Recall that, for each duplication d, we use Loss(d) to denote the set of nodes on which a loss event occurs in d and let

$$L_d = \{y \in S \mid y \in \text{Loss}(d)\}. \tag{3}$$

Then, from all S_y constructed above, we can derive all L_d as follows.

PROPOSITION 3.7. *All L_d can be computed in $O(l)$ steps. Thus, all loss subtrees can be constructed in $O(l)$ steps.*

Proof. Radix sort S_{loss} and let

$$y_1, y_2, \ldots, y_c$$

be the in-order list of nodes in S_{loss}. For a duplication, we will keep L_d in a linked list which is denoted by the same symbol. Let there be m duplications $d_1, d_2, \ldots, d_m$. Initially, L_{d_i} is empty for every i. Then, we examine $S_{y_1}, S_{y_2}, \ldots, S_{y_c}$ in order one by one. First, for each $d \in S_{y_1}$, we insert y_1 in L_d. In general, after $S_{y_1}, S_{y_2}, \ldots, S_{y_i}$ have been examined, we search $S_{y_{i+1}}$ in the same way: for each $d \in S_{y_i}$, we insert y_{i+1} in L_d. After all S_{y_i} have been examined, L_d stores all the nodes on which a loss event occurs in duplication d. Actually, following the construction carefully, one will see that L_d is an in-order traversal. Therefore, one can easily construct the loss subtree for duplication d from L_d. The time bound is obvious. □

We have proved the following theorem.

THEOREM 3.8. *Given a gene tree G and a species tree S, Algorithm* A *constructs the loss subtrees in $O(n+l)$ time.*

```
ALGORITHM A.
INPUT: A gene tree $G$ and species tree $S$.
1. Impose an arbitrary ordering on the children of each node and
     produce an in-order traversal of $G$ and $S$, respectively.
     Assume $i_x$ denotes the in-order number of $x$ for $x \in G, S$.
2. Compute the LCA mapping $M : G \to S$. To each $x \in G$ assign
     a pair $< i_x, i_{M(x)} >$; To each $y \in S$ assign a pair $< i_y, n_{23} >$, where
     $n_{23}$ is the number of type-2 or type-3 nodes in $G$ that are mapped to $y$.
3. Compute the set $S_{loss}$ of all the nodes in which a loss event occurs
     using Proposition 3.3.
4. For each $y \in S_{loss}$, compute the set $S_y$ that is defined in equation (2).
5. For each duplication $d$, use the sets $S_y$ to compute $L_d$ that is defined
     in equation (3).
6. Reconstruct all the loss subtrees from the $L_d$'s.
```

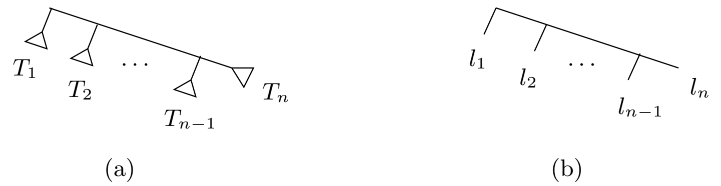


FIG. 6. (a) *The tree* $L[T_1, T_2, \ldots, T_n]$ *and* (b) *a line tree.*

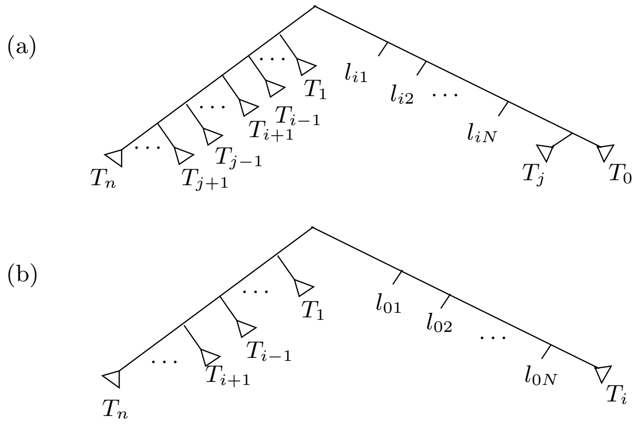


FIG. 7. (a) *The gene tree* G_{ij} *is constructed from the edge* (v_i, v_j). (b) *The gene tree* G_i *is constructed from the node* v_i.

4. The complexity of finding optimal species trees.

4.1. Optimal species tree I. Given n trees $T_1, T_2, \ldots, T_n$, we use $L[T_1, T_2, \ldots, T_n]$ to denote the tree T shown in Figure 6(a). When T_i is a single labeled node, the resulting tree is just a *line tree* as in Figure 6(b).

THEOREM 4.1. *The decision version of OST* I *is NP-complete.*

Proof. The problem is trivially in NP. To prove its NP-hardness, we reduce the independent set problem to OST I. Recall that the independent set problem is as follows: given a graph $G = (V, E)$ and an integer $d \leq |V|$, decide if G contains an independent set of size d, i.e., a subset of $V' \subseteq V$ such that $|V'| = d$ and no two nodes in V' are joined by an edge in E. Given an instance $G = (V, E)$ of the independent set problem, where $V = \{v_1, v_2, \ldots, v_n\}$, we construct a corresponding instance of OST I as follows.

Let $N = 5n^3$. For each v_i, we introduce N labels l_{ip}, $1 \leq p \leq N$, and a line tree $T_i = L[l_{i1}, l_{i2}, \ldots, l_{iN}]$. We also introduce extra N labels l_{0p}, $1 \leq p \leq N$, and a line tree $T_0 = L[l_{01}, l_{02}, \ldots, l_{0N}]$. For each pair (i, j) $(1 \leq i \neq j \leq n)$ such that $(v_i, v_j) \in E$, we define a tree G_{ij} with leaves labeled by $A = \{l_{ip} \mid 0 \leq i \leq n, 1 \leq p \leq N\}$ as shown in Figure 7(a). In G_{ij}, the left subtree is formed by connecting all T_p's $(p > 1)$ except for T_i and T_j by a line tree. Note that G_{ij} and G_{ji} have different right subtrees. Hence, we use two trees G_{ij} and G_{ji} to encode an edge (i, j). Finally, for each $v_i \in V$, we define a tree G_i with leaves labeled by A as shown in Figure 7(b). The left subtree of G_i is formed by connecting all T_p's except for T_i by a line tree, and the right subtree is a line tree with leaves l_{0k} $(1 \leq k \leq N)$ and l_{ik} $(1 \leq k \leq N)$ from left to right. Overall, we encode an edge (v_i, v_j) by two trees G_{ij} and G_{ji} and a node v_i by one tree G_i. Obviously, such a construction can be carried out in polynomial time. The NP-hardness of OST I derives from the following lemma. $\square$

LEMMA 4.2. *The graph G contains an independent set of size d if and only if there is a species tree S for all the gene trees G_{ij} and G_i constructed above with the duplication cost $c < (|E| + n - d + \frac{1}{2})N$.*

Proof. ($\Rightarrow$) Assume that G contains an independent set K of size d. Without loss of generality, we assume $V(K) = \{v_1, v_2, \dots, v_d\}$. Then, we define a species tree S as

$$S = L[l_{n1}, \dots, l_{nN}, \dots, l_{(d+1)1}, \dots, l_{(d+1)N}, l_{01}, \dots, l_{0N}, l_{d1}, \dots, l_{dN}, \dots, l_{11}, \dots, l_{1N}].$$

For each $i \le d$, $t_{dup}(G_i, S) = n - 1$. For each $i > d$, $t_{dup}(G_i, S) = N + n - 1$. Further, for any $(v_i, v_j) \in E$, either $i > d$ or $j > d$, and so

$$N \le t_{dup}(G_{ij}, S) + t_{dup}(G_{ji}, S) \le N + 2n. \tag{4}$$

Thus, the duplication cost c of S is

$$\begin{aligned} \sum_{(v_i,v_j)\in E} &(t_{dup}(G_{ij}, S) + t_{dup}(G_{ji}, S)) + \sum_{1\le i\le n} t_{dup}(G_i, S) \\ &\le |E|(N + 2n) + (n - d)(N + n - 1) + d(n - 1) \\ &\le (|E| + n - d)N + 2n^3 \\ &< \left(|E| + n - d + \frac{1}{2}\right) N. \end{aligned}$$

($\Leftarrow$) We prove the converse by contradiction. Suppose that the optimal duplication cost is c for gene trees G_{ij} and G_i. Denote $A_i = \{l_{ip} \mid 1 \le p \le N\}$. Let S be an optimal species tree. Then one can define a total order $\prec$ on $\{A_i | 1 \le i \le n\}$ such that $LCA(A_i) \subset LCA(A_j)$ implies $A_i \prec A_j$. Suppose $A_{i_n} \prec A_{i_{n-1}} \prec \cdots \prec A_{i_0}$ is such a total order; then we define a line tree S' as

$$S' = L[l_{i_0 1}, \dots, l_{i_0 N}, l_{i_1 1}, \dots, l_{i_1 N}, \dots, l_{i_n 1}, \dots l_{i_n N}].$$

Let S' have duplication cost c'. Then we have the following two facts.

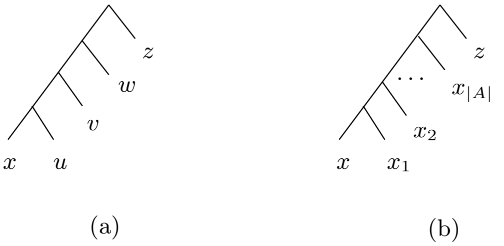
Fact 1. $c' \le 2\,n^3 + c$.

Proof. Since $S'|_{A_i} = T_i$, no duplication happens at all subtrees T_i $(0 \le i \le n)$ in each gene tree $G_{i'j'}$ and $G_{i'}$. On the other hand, let u be any internal node on a right subtree of G_{ij}. If u is a parent of some l_{ip} $(0 \le p \le N - 1)$ and u is not a duplication node in the mapping from G_{ij} to S, then it is easy to see $LCA(A_j) \subset LCA(A_i)$ and $LCA(A_0) \subset LCA(A_i)$. Thus $A_j \prec A_i$ and $A_0 \prec A_i$. Therefore, u is not a duplication node in the mapping from G_{ij} to S'. Note that two exceptions are the parent and the brother of l_{iN}. Similarly, for each internal node $x \in G_i$ that is a parent of l_{0p} $(1 \le p \le N - 1)$, it will not be a duplication node for S' if it is not for S. Thus, the duplication cost for S' on all the right subtrees of gene trees is at most the cost for S plus $2n^2$.

Since there are at most $n' = n^2 + n(n - 1)(n - 2)$ extra internal nodes on the left subtrees of gene trees that have not been considered above, we have that $c' \le 2n^2 + n' + c \le 2n^3 + c$. This finishes the proof of Fact 1. $\square$

Fact 2. $c' \ge (|E| + n - d + 1)N$.

Proof. Let $E_< = \{(v_i, v_j) \in E \mid \text{parent}_{S'}(l_{i1}), \text{parent}_{S'}(l_{j1}) \subset \text{parent}_{S'}(l_{01})\}$ and $V_< = \{v_i \in V | \text{parent}_{S'}(l_{i1}) \subset \text{parent}_{S'}(l_{01})\}$. If $G = (V, E)$ does not contain



(a)

(b)

FIG. 8. *Rooted line trees.*

an independent set of size d, then $|E_<| - |V_<| \geq 1 - d$. In fact, this is trivial if $|V_<| < d$. Otherwise, let the largest independent set of restriction subgraph $G|_{V_<}$ be K'. Then $|K'| \leq d - 1$. Since K' is largest, for any node $v \in V_< - K'$, $(v, v') \in E$ for some $v' \in K'$. This implies that $|E_<| \geq |V_<| - |K'| \geq |V_<| - d + 1$ or, equivalently, $|E_<| - |V_<| \geq 1 - d$ when $|V_<| \geq d$.

It is easy to verify that, for any i, j, if $(v_i, v_j) \in E_<$, then

$$t_{dup}(G_{ij}, S') + t_{dup}(G_{ji}, S') \geq 2N, \tag{5}$$

and if $v_i \notin V_<$, then

$$t_{dup}(G_i, S') \geq N. \tag{6}$$

By formulae (4), (5), and (6), we have

$$\begin{aligned} c' &\geq \sum_{v_i \in V - V_<} t_{dup}(G_i, S') + \sum_{(v_i, v_j) \in E - E_<} (t_{dup}(G_{ij}, S') + t_{dup}(G_{ji}, S')) \\ &\quad + \sum_{(v_i, v_j) \in E_<} (t_{dup}(G_{ij}, S') + t_{dup}(G_{ji}, S')) \\ &\geq N\,(n - |V_<|) + N\,(|E| - |E_<|) + 2\,N\,|E_<| \\ &\geq (|E| + n + |E_<| - |V_<|)N \\ &\geq (|E| + n - d + 1)N. \end{aligned}$$

Thus, Fact 2 is proved. □

Combining Fact 1 and Fact 2, we have that $c > (|E| + n - d + \frac{1}{2})N$, a contradiction. Thus, we finish the proof of the lemma. □

Remark 1. We have actually proved that OST I is NP-hard even for all gene trees with the same uniquely labeled leaves. Such a stronger conclusion will be used to prove that OST III is NP-hard in section 4.3.

Remark 2. Based on the above remark, we can also prove that the decision version of OST I remains NP-complete even for one gene tree that are not uniquely leaf-labeled. The proof of this result can be found in the proof of Theorem 4.7.

4.2. Optimal species tree II. Let C be a set of full binary trees G with leaves uniquely labeled by $L(G)$, and let T be a full binary tree with leaves uniquely labeled by $\sum_{G \in C} L(G)$. We say that C is *compatible* with T if for every $G \in C$, the homomorphic subtree $T|_{L(G)}$ of T induced by $L(G)$ is G, and it is *compatible* if it is compatible with some tree. Finally, recall that $L[z, w, v, u, x]$ denotes a rooted line tree with 5 leaves z, w, v, u, x as shown in Figure 8(a).

LEMMA 4.3. *If a collection C of* 5*-leave rooted line trees $L[y, w_i, v_i, u_i, x]$, $1 \leq i \leq k$, is compatible, then it is compatible with a rooted line tree $L[y, x_n, x_{n-1}, \ldots, x_1, x]$, where $\{x_1, x_2, \ldots, x_n\} = \cup_{i=1}^{k} \{u_i, v_i, w_i\}$.*

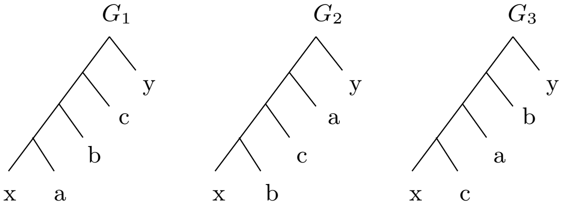


FIG. 9. *Three trees correspond to an ordered triple* (a, b, c).

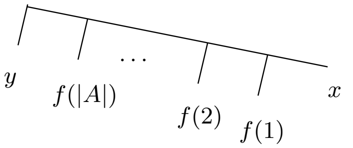


FIG. 10. *The species tree constructed from a cyclic ordering* f.

Proof. Choose a label z not in $\{x, y\}$ and $\cup_{i=1}^{k}\{u_i, v_i, w_i\}$. For each $t = L[y, w_i, v_i, u_i, x]$, we add an edge between z and the root so that the resulting tree t^z is an unrooted full binary tree in which each internal node has degree 3. It is not difficult to see that t^z is defined by the following set of quartets [29]:

$$Q(t^z) = \{xu_i|v_iz, xv_i|w_iz, xu_i|yz, xv_i|yz, xw_i|yz\}.$$

Suppose that C is compatible with a rooted full binary tree T; then $C^z = \{t^z \mid t \in C\}$ is compatible with T^z, and thus quartet set $\cup_{t\in C} Q(t^z)$ is compatible with T^z. By a lemma in [29], $\cup_{t\in C} Q(t^z)$ is compatible with a line tree $L[x, u_1, u_2, \dots, u_{|A|}, y, z]$. This implies that C is compatible with the binary tree rooted at the internal node that is jointed with z (after the removal of z), which has the form shown in Figure 8(b). ∎

THEOREM 4.4. *The decision version of OST* II *is NP-complete.*

Proof. The problem is obviously in NP. To prove its NP-hardness, we now describe a transformation from the cyclic ordering problem [13] to OST II. The cyclic ordering problem is defined as follows.

> INSTANCE: A finite set A, and a collection C of ordered triples (a, b, c) of distinct elements from A.
>
> QUESTION: Is there a one-to-one function $f : A \to \{1, 2, \dots, |A|\}$ such that, for each $(a, b, c) \in C$, we have either $f(a) < f(b) < f(c)$ or $f(b) < f(c) < f(a)$ or $f(c) < f(a) < f(b)$?

The problem is proved to be NP-complete in [12].

Suppose an instance (A, C) of the cyclic ordering problem is given. We construct for each ordered triple $\pi = (a, b, c) \in C$ three gene trees $G_1^\pi = L[y, c, b, a, x]$, $G_2^\pi = L[y, a, c, b, x]$, and $G_3^\pi = L[y, b, a, c, x]$ as shown in Figure 9, where x and y are two new labels fixed for all triples in C. Now, we consider a collection $G(C) = \{G_i^\pi \mid 1 \le i \le 3, \pi \in C\}$ of $3|C|$ gene trees. Obviously, such a construction can be carried out in polynomial time.

We claim that there is a species tree, with leaves $A \cup \{x, y\}$, which has mutation cost at most $14|C|$ if and only if A has a cyclic ordering.

Suppose a cyclic ordering f exists. Let $f(i)$ denote the ith smallest element in A and let $S = L[y, f(|A|), \dots, f(2), f(1), x]$ as in Figure 10.

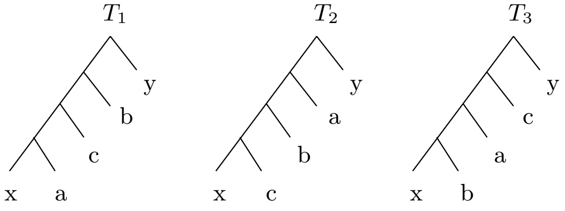


FIG. 11. *Three trees in the first column in Table* 1.

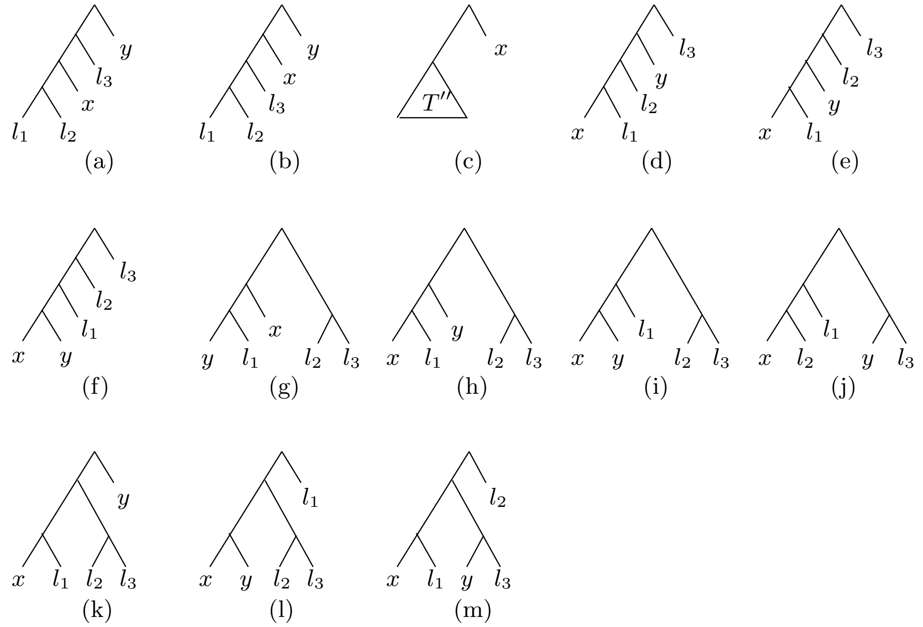

y l_3 x l_1 l_2
(a)
y x l_3 l_1 l_2
(b)
x T''
(c)
l_3 y l_2 x l_1
(d)
l_3 l_2 y x l_1
(e)
l_3 l_2 l_1 x y
(f)
y l_1 x l_2 l_3
(g)
x l_1 y l_2 l_3
(h)
x y l_1 l_2 l_3
(i)
x l_2 l_1 y l_3
(j)
y x l_1 l_2 l_3
(k)
l_1 x y l_2 l_3
(l)
l_2 x l_1 y l_3
(m)

FIG. 12. *Cases* (a)–(m) *in the proof of Claim* 1.

For a triple $\pi = (a, b, c) \in C$, without loss of generality, we may assume that $f(a) < f(b) < f(c)$. Then G_1^π is the homomorphic subtree of S on $\{x, a, b, c, y\}$. Thus, $c(G_1, S) = 0$, $c(G_2, S) = 5$, and $c(G_3, S) = 9$. Hence, the total mutation cost over all $3|C|$ gene trees is $14|C|$.

Conversely, suppose that T is a species tree with leaves $A \cup \{x, y\}$ and with mutation cost at most $14|C|$. Then we have the following fact.

Fact. For any $\pi = (a, b, c) \in C$, the homomorphic subtree of T on $\{x, a, b, c, y\}$ is G_1, G_2, or G_3 as shown in Figure 9.

Proof. The homomorphic subtree T' of T on $\{x, a, b, c, y\}$ is a full binary tree with five labeled leaves. Assume that it is not one of G_1^π, G_2^π, or G_3^π. All possible homomorphic subtrees are illustrated in Figure 11 and Figure 12 and a case-by-case analysis of the mutation cost of G_1, G_2, and G_3 with T is shown in Table 1.

Hence, T has mutation cost at least $14|C| + 1$. This is a contradiction. Thus we conclude the fact. □

By Lemma 4.3, there exists a line tree such that for each triple $\pi = (a, b, c)$, the homomorphic subtree on $\{x, y, a, b, c\}$ is one of the gene trees $G_1^\pi, G_2^\pi, G_3^\pi$. It is not difficult to see that such a line tree induces a cyclic ordering. This concludes the proof of Theorem 4.4. □

Table 1
Case-by-case analysis of duplications.

Case	T_i	(a)	(b)	(c)	(d)
Cost	17	18	27	42, 45	29,32
Case	(e)	(f)	(g)	(h)	(i)
Cost	31,34	32,35	35	34	35
Case	(j)	(k)	(l)	(m)	
Cost	26,29	20	33	28, 29,32	

4.3. Optimal species tree III. To prove the hardness result, we need to establish Lemma 4.5 and Lemma 4.6, which are derived from the definition of reconciled trees. Recall that for a node g in a gene tree G, $G(g)$ denotes the subtree of G rooted at g.

Lemma 4.5. *Given a gene tree G and a species tree S, let T_r be the reconciled tree of G with respect to S, and let g be an internal node in G. If g is mapped to $t \in T_r$ when G is considered as a subtree of T_r, then $T_r(t)$ is the reconciled tree of $G(g)$ with respect to $S(t)$.*

Proof. The lemma follows from the definition of reconciled trees. □

Lemma 4.6. *Let T_r be the reconciled tree of G with respect to S. Then $t_{dup}(T_r, S) = t_{dup}(G, S)$.*

Proof. We prove this lemma by induction on the number of leaves in G. It is obviously true for a gene tree G that has only three leaves. Now assume that G has at least four leaves. Let t be the root of T_r with children $a(t)$ and $b(t)$, let g be the root of G with children $a(g)$ and $b(g)$, and let s be the root of S with children $a(s)$ and $b(s)$. We consider the following cases.

Case 1. $a(t) \cap b(t) = \phi$.

Note that $t = s$ and $a(t)$ and $b(t)$ are two clusters in S. Further, by the definition of reconciled trees, $a(t) \neq t$, and $b(t) \neq t$. Thus, t is not a duplication node under the LCA mapping from T_r to S. On the other hand, since G is identical to $T_r|_{L(G)}$, we have that $a(g) \subset a(t), b(g) \subset b(t)$ or $a(g) subset b(t)$, $b(g) \subset a(t)$. Let $a(g)$ and $b(g)$ be mapped to t_1 and t_2, respectively, when G is considered as a subtree of T_r. By Lemma 4.5, $T_r(t_1) = T_r(G(a(g)), S(t_1))$ and $T_r(t_2) = T_r(G(b(g)), S(t_2))$. By induction, $t_{dup}(T_r(t_1), S(t_1)) = t_{dup}(G(a(g)), S(t_1))$ and $t_{dup}(T_r(t_2), S(t_2)) = t_{dup}(G(b(g)), S(t_2))$. Since $a(g) \subseteq t_1$ and $b(g) \subseteq t_2$, g is not a duplication node under the LCA mapping from G to S. Thus,

$$\begin{aligned} t_{dup}(G,S) &= t_{dup}(G(a(g)), S(a(s))) + t_{dup}(G(b(g)), S(b(s))) \\ &= t_{dup}(T_r(a(t)), S(a(s))) + t_{dup}(T_r(b(t)), S(b(s))) \\ &= t_{dup}(T_r, S). \end{aligned}$$

Case 2. $a(t) = b(t)$.

Then $a(t) = b(t) = t = s$. Thus a duplication happens at t under the LCA mapping from T_r to S. Since $a(t) = b(t)$, then either $a(g)$ is mapped to $a(t)$ or $b(g)$ is mapped to $b(t)$. Otherwise, $a(t)$ or $b(t)$ contains G as a subtree, which contradicts the fact that T_r is the reconciled tree of G with respect to S. Without loss of generality, we may assume that the former is true. Let $b(g)$ be mapped to t'. Note that $t' \subseteq b(t) = s$. Under the LCA mapping from G to S, $a(g)$ is mapped to s. Thus, by induction,

$$\begin{aligned} t_{dup}(T_r, S) &= 1 + t_{dup}(T_r(a(t)), S) + t_{dup}(T_r(b(t)), S) \\ &= 1 + t_{dup}(a(g), S) + t_{dup}(G(b(g)), S(t')) \\ &= t_{dup}(G, S). \end{aligned}$$

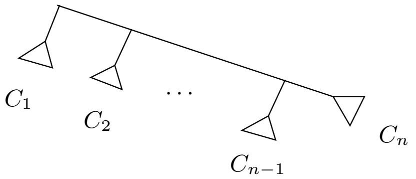


FIG. 13. *Connection of m gene trees in a right line tree.*

This proves Lemma 4.6. □

By Lemma 4.6, the problem OST III is a special case of the problem OST I in which each instance has only one gene tree. Unfortunately, such a problem is still NP-hard when a given gene tree is not a uniquely leaf-labeled tree.

THEOREM 4.7. *The decision version of OST* III *is NP-complete.*

Proof. Again, the problem is obviously in NP. To prove its NP-hardness, by Lemma 4.6, we need only to prove the following problem to be NP-hard:

> *Given a gene tree, find a species tree S with the minimum duplication cost $t_{dup}(G, S)$.*

Given a class C of m gene trees with the same n uniquely labeled leaves, we construct a gene G by connecting all gene trees in C through a right line tree as shown in Figure 13. Since all gene trees in C have the same labeled leaves, we have that for any species tree S,

$$t_{dup}(G, S) = m - 1 + \sum_{1 \le i \le m} t_{dup}(G_i, S).$$

This finishes the reduction from an NP-hard problem OST I to the problem given above (see Remark 1 after Theorem 4.1). □

5. A heuristic method.

We have proved that the problem of reconstructing an optimal species tree from gene trees is NP-hard. Therefore, there is unlikely an efficient algorithm for the problem. In this section, we will develop a heuristic method for it in a special case when all gene trees are uniquely leaf-labeled. Throughout this section, we will assume that trees are uniquely leaf-labeled without explicitly mentioning it.

5.1. A new metric.

In this section, we introduce a new metric for measuring the similarity of two rooted full binary trees with uniquely labeled leaves based on the concept of duplications. Given two rooted full binary trees T_1 and T_2, in which each internal node has at least two children, we define the LCA mapping M from T_1 to T_2 as in section 2. We say a *duplication* happens at $x \in T_1$ under M if and only if for some child $c(x)$ of x, $M(c(x)) = M(x)$. We also use $t_{dup}(T_1, T_2)$ to denote the number of duplications occurring under the mapping M.

Let T be a rooted full binary tree. For any internal edge $e = (u, v)$, the *contraction tree* of T at e is the resulting tree after the removal of e and combining u and v into a new node p such that p is adjacent to all the neighbors of both u and v.

LEMMA 5.1. *The duplication cost satisfies the triangle inequality, i.e., $t_{dup}(T_1, T_3) \le t_{dup}(T_1, T_2) + t_{dup}(T_2, T_3)$ for any three rooted full binary trees T_1, T_2, and T_3 with same uniquely labeled leaves.*

Proof. Let M_{ij} denote the LCA mapping from T_i to T_j. Now let T_1' be the resulting tree from T_1 by contracting all edges (u, v) such that $M_{12}(u) = M_{12}(v)$. Furthermore, let M_{12}' be the mapping from T_1' to T_2. We prove the following facts.

Fact 1. For any $m \in T_1'$, $M_{12}'(m) = m$. Thus, $t_{dup}(T_1', T_2) = 0$.

Proof. It follows from the definition of T_1'. □

Fact 2. $t_{dup}(T_1, T_3) \le t_{dup}(T_1, T_2) + t_{dup}(T_2, T_3)$ if $t_{dup}(T_1', T_3) \le t_{dup}(T_2, T_3)$.

Proof. Under the mapping M_{13}, a duplication happens at a node $n \in T_1$ if and only if $M_{13}(n) = M_{13}(c(n))$ for some child $c(n)$ of n. Let D denote the set of such duplication nodes in T_1 under M_{13}. We divide D into two disjoint subsets:

$$D_1 = \{n \in D \mid M_{12}(n) = M_{12}(c(n))\}$$

and

$$D_2 = \{n \in D \mid M_{12}(n) \neq M_{12}(c(n))\}.$$

Obviously, $|D_1| \le t_{dup}(T_1, T_2)$ since any node in D_1 is also a duplication node under M_{12}. Furthermore, by the definition of T_1', $|D_2| \le t_{dup}(T_1', T_3)$ since any node in D_2 is a duplication under the LCA mapping from T_1' to T_3. Hence, $t_{dup}(T_1, T_3) = |D_1| + |D_2| \le t_{dup}(T_1, T_2) + t_{dup}(T_1', T_3) \le t_{dup}(T_1, T_2) + t_{dup}(T_2, T_3)$ if $t_{dup}(T_1', T_3) \le t_{dup}(T_2, T_3)$. This concludes the proof of Fact 2. □

Let $M_{12}'(n) = p$ and $M_{12}'(c(n)) = q$. Then, by Fact 1, $n = p$ and $c(n) = q$. If $M_{13}(n) = M_{13}(c(n))$, then all nodes in the path from $M_{23}(p)$ and $M_{23}(q)$ are mapped to the same node in T_3. This implies that $t_{dup}(T_1', T_3) \le t_{dup}(T_2, T_3)$ and so, by Fact 2, $t_{dup}(T_1, T_3) \le t_{dup}(T_1, T_2) + t_{dup}(T_2, T_3)$. This finishes the proof of Lemma 5.1. □

Now we define a new similarity measure between two rooted full binary trees as

$$d(T_1, T_2) = \frac{t_{dup}(T_1, T_2) + t_{dup}(T_2, T_1)}{2}.$$

Since the duplication cost is computable in linear time [34], the measure $d(.,.)$ is also efficiently computable. Further, it satisfies the three metric axioms.

PROPOSITION 5.2. *For any three full binary trees T_1, T_2, and T_3 with the same uniquely labeled leaves, $d(.,.)$ satisfies the following properties:*

(1) $d(T_1, T_2) = 0$ *if and only if* $T_1 = T_2$*;*

(2) $d(T_1, T_3) \le d(T_1, T_2) + d(T_2, T_3)$ *for any* T_2*;*

(3) $d(T_1, T_2) = d(T_2, T_1)$.

In what follows, we call $d(.,.)$ the *symmetric duplication cost*. Interestingly, the symmetric duplication cost is closely related to the NNI distance for full binary trees, which was introduced independently in [19] and [27]. An NNI operation swaps two subtrees that are separated by an internal edge (u, v) as illustrated in Figure 14. The *NNI distance*, $D_{NNI}(T_1, T_2)$, between two full binary trees T_1 and T_2 is defined as the minimum number of NNI operations required to transform one tree into the other.

PROPOSITION 5.3. *For any species trees T_1 and T_2, $d(T_1, T_2) \le D_{NNI}(T_1, T_2)$.*

Proof. Suppose T_1 is converted into T_2 by one NNI operation. Then, we can easily verify that $d(T_1, T_2) = 1$. Thus, $d(T_1, T_2) \le D_{NNI}(T_1, T_2)$. Since $d(.,.)$ satisfies the triangle inequality, the result holds in general also. □

We now prove the following NP-completeness result.

THEOREM 5.4. *The decision version of finding an optimal species tree from a set of gene trees is NP-complete under the symmetric duplication cost.*

Proof. Obviously, it is in NP. In section 4.1, we have shown that OST I is NP-complete even for all gene trees with the same uniquely labeled leaves. Moreover, we may even assume that the duplication cost between any two gene trees is at least 2.

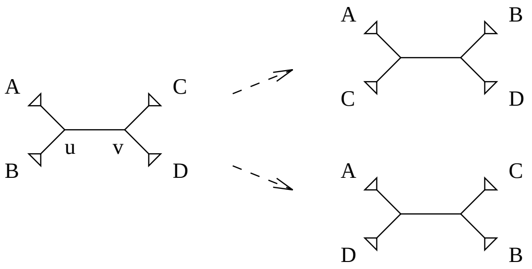


FIG. 14. *The two possible NNI operations on an internal edge* (u, v).

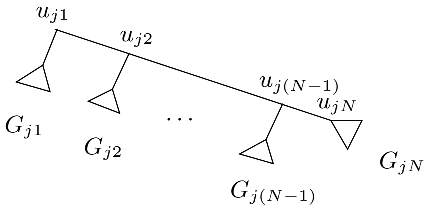


FIG. 15. *The tree* G'_j *of Theorem* 5.4.

We reduce this special case of OST I to the problem of finding an optimal species tree from gene trees under the symmetric duplication cost. Given an instance of OST I $I_1 = \{G_1, G_2, \ldots, G_n\}$ where each G_i is leaf uniquely labeled, and for any $i \neq j$, $L(G_i) = L(G_j)$ and $t_{dup}(G_i, G_j) \geq 2$. Assume that the leaf label set is $L = \{l_1, l_2, \ldots, l_m\}$ for each gene tree. For any species tree S with leaf label set L, if $S \neq G_i$ for any i, $1 \leq i \leq n$, then $t_{dup}(S, G_i) \geq 1$ for any i. If $S = G_i$ for some i in the range from 1 to n, then $t_{dup}(S, G_j) \geq 2$ for any $j \neq i$. Thus, for any species tree S with leaf label set L,

$$\sum_{j=1}^{n} t_{dup}(S, G_j) \geq n. \tag{7}$$

Let $N = 3n^2$. We introduce mN new labels l_{ik}, $1 \leq i \leq m$ and $1 \leq k \leq N$. For any j $(1 \leq j \leq n)$ and k $(1 \leq k \leq N)$, we construct G_{jk} from G_j by replacing the leaf l_i by l_{ik} for every i, $1 \leq i \leq m$. Let G'_j be the tree $L(G_{j1}, G_{j2}, \ldots, G_{jN})$ defined in Figure 15. Note that G'_j is a tree with mN labeled leaves. Finally, let $I_2 = \{G'_1, G'_2, \ldots, G'_n\}$. In order to finish the reduction, we now prove that I_1 has a solution with cost at most d if and only if I_2 has a solution with cost less than $(\frac{d+n}{2} + \frac{1}{4})N$.

Suppose S is a solution for I_1 with cost d. Let S' be the tree obtained from S by replacing each leaf l_i by a line tree $L(l_{i1}, l_{i2}, \ldots, l_{iN})$. Then it is easy to see that $t_{dup}(G_{jk}, S') = t_{dup}(G_j, S)$. Thus, $t_{dup}(G'_j, S') \leq t_{dup}(G_j, S)N + N$. Furthermore, since $t_{dup}(L(l_{i1}, l_{i2}, \ldots, l_{iN}), G'_k) = 0$, and S' has $n - 1$ internal nodes that are not in $L(l_{i1}, l_{i2}, \ldots, l_{iN})$ for any i, $t_{dup}(S', G'_j) \leq n-1$. Therefore, the symmetric duplication cost of the solution S' for I_2 is

$$\sum_{i=1}^{n} \frac{t_{dup}(S', G'_j) + t_{dup}(G'_j, S')}{2} \leq \frac{1}{2}(dN + nN + n(n-1)) < \left(\frac{d+n}{2} + \frac{1}{4}\right)N.$$

Conversely, assume that the optimal solution for I_1 has duplication cost at least

$d+1$. Suppose S is a solution of I_2. For any $1 \leq k \leq N$, let $A_k = \{l_{ik} | 1 \leq i \leq m\}$, $S_k = S|_{A_k}$, and let u_k be the LCA of u_{jk} in S, where u_{jk}'s are the nodes in G'_j as shown in Figure 15. Note that u_k does not depend on the choice of j. Obviously, $u_N \subseteq \cdots \subseteq u_2 \subseteq u_1$. Assume that there are h indices k's ($1 \leq k \leq N-1$) satisfying $u_{k+1} \subset u_k$. Let these indices be $k_1, k_2, \ldots, k_h$. Then for any $k \neq k_t$ ($t = 1, 2, \ldots, h$), $1 \leq k \leq N-1$, u_{jk} is a duplication node in the mapping from G'_j to S. Hence, we have that

$$t_{dup}(G'_j, S) \geq \sum_{k=1}^{N} t_{dup}(G_{jk}, S) + N - 1 - h. \tag{8}$$

We use h_j to denote the number of duplications that occur on one of the nodes u_{k_1}, u_{k_2}, ..., u_{k_h} under the LCA mapping from S to G'_j. Let j' be the index that minimizes h_j over all j from 1 to n and let $h' = h - h_{j'}$. Assume that $u_{r_1}, u_{r_2}, \ldots, u_{r_{h'}}$ are the h' nonduplication nodes in the mapping from S to $G'_{j'}$. We have that $\{r_1, r_2, \ldots, r_{h'}\} \subseteq \{k_1, k_2, \ldots, k_h\}$ and $r_1 < r_2 < \cdots < r_{h'}$. Let $A = \bigcup_{t=1}^{h'} A_{r_t}$; then it is easy to verify that in the tree $S|_A$, for any $1 \leq s \leq h'$, $A_{r_s} \cap LCA(\bigcup_{t=s+1}^{h'} A_{r_t}) = \emptyset$. Thus,

$$t_{dup}(S, G'_j) \geq h_j + t_{dup}(S|_A, G'_j) \geq h - h' + \sum_{t=1}^{h'} t_{dup}(S_{r_t}, G_{jr_t}). \tag{9}$$

Combining formulae (7), (8), and (9), we have

$$\begin{aligned}
&\sum_{j=1}^{n} \left[t_{dup}(G'_j, S) + t_{dup}(S, G'_j)\right] \\
&\quad \geq \sum_{k=1}^{N}\sum_{j=1}^{n} t_{dup}(G_{jk}, S) + n(N - 1 - h') + \sum_{t=1}^{h'}\sum_{j=1}^{n} t_{dup}(S_{r_t}, G_{jr_t}) \\
&\quad \geq (d + n + 1)N - n.
\end{aligned}$$

Thus we know that for any solution of I_2, the cost is at least $\frac{(d+n+1)N-n}{2} \geq (\frac{d+n}{2} + \frac{1}{4})N$. □

5.2. A heuristic method for finding species trees. Although finding an optimal species tree from gene trees is NP-hard for the symmetric duplication cost $d(.,.)$, we have the following approximation result.

THEOREM 5.5. *There is a polynomial-time approximation of ratio* 2 *to the problem of finding an optimal species tree from gene trees with the symmetric duplication cost* $d(.,.)$.

Proof. Given an input of n gene trees $G_1, G_2, \ldots, G_n$, we compute $\sum_{i \neq j}^{n} d(G_i, G_j)$ for each $j \leq n$ and output G_j with the minimum cost $\sum_{i \neq j}^{n} d(G_i, G_j)$ as the species tree. We now prove that the output species tree has at most twice the optimal symmetric duplication cost. Assume that G_1 is the output and S is an optimal species tree. Then

$$\begin{aligned}
\sum_{i \leq n} d(G_i, G_1) &\leq \frac{\sum_{i \leq n}\sum_{j \leq n} d(G_i, G_j)}{n} \\
&\leq \frac{\sum_{i \leq n}\sum_{j \leq n} (d(G_i, S) + d(G_j, S))}{n} \\
&\leq 2\sum_{i \leq n} d(G_i, S).
\end{aligned}$$

This proves Theorem 5.5. □

In general, the optimal species tree for a set of gene trees under the symmetric duplication cost is different from ones under the duplication and mutation costs. However, these trees should be quite similar to each other intuitively. Hence, based on Theorem 5.5, we propose the following heuristic method for the problem.

Search Paradigm

```
Input: Gene tree $G_1, G_2, \ldots, G_n$.
1. Find a gene tree $T' = T_k$ with the minimum
   symmetric duplication cost $\sum_{i \leq n} d(T_i, T_k)$.
2. Search for the optimal species tree starting from
   $T'$ using NNI, CP, or alternate NNI and CP.
```

Here cut and paste (CP) is also known as subtree pruning and regrafting [30]. According to the experimental research conducted by Page and Charleston [24], the best choice seems to be alternating between the NNI and CP method in step 2 of our heuristic method.

We have extensively tested our heuristic method and compared it with the algorithm that starts the search from a random tree. The latter was implemented in Page's package GeneTree Version 1.0. When running Page's algorithm, we start from a random tree and search near-optimal species trees using the method of alternating NNI and CP. We also use the method of alternating NNI and CP to do the search in our algorithm. When there are less than 10 species, and gene trees in each data set are chosen randomly, both algorithms perform well. They produce quickly species trees with optimal duplication costs. However, when there are over 15 species, and gene trees in a data set are closely related, which is usually true for practical molecular data, our algorithm performs much better. We have conducted 22 tests. We generated a set of gene trees as follows: (a) Generate a random tree R using an algorithm of Rémy [1, 26]; (b) repeatedly generate a tree by randomly choosing up to 10 NNI operations and applying these operations on R. The results are listed in Table 2 except for three unfinished tests in which the algorithm of searching from a random tree took over one hour and was stopped before finishing, but our algorithm finished within half a minute. Our algorithm found species trees with better duplication costs in all the cases and took much fewer CP and NNI operations (and hence much less time) to get the solution. We used a Pentium MMX-233 personal computer. In each of the 22 tests, our algorithm finished in less than half a minute, while the search-from-a-random-tree algorithm took more than one hour for 6 tests.

6. A general reconstructing problem.

There is a large family of genes each having several distinct copies in the studied species. In order to derive a species tree that truly reflects the evolution of species, one needs full knowledge about which copies of the gene are comparable. This is usually impossible until a careful study of the species has been done. However, one may have different confidences in different genes. Hence, it is natural to propose the following general problem. We use I^+ to denote the set of positive integers and let m be any similarity measure between gene and species trees.

General Optimal Species Tree (GOST).

INSTANCE: A set of n gene trees $G_1, G_2, \ldots, G_n$, to each tree a confidence value $c_i \in I^+$ is associated.

QUESTION: Find a species tree S with the minimum cost $\sum_{i=1}^{n} c_i m(G_i, S)$.

Table 2

R = *the algorithm that starts the search with a random tree,* SP = *our search paradigm. The last column contains the numbers of NNI/CP operations used by the two algorithms.*

Data sets	Species	Gene trees	Alg.	Optimal trees	Dup. cost	NNI/CP operations
1	15	5	SP R	14	18	10628 37872
2	15	5	SP R	4	21	2848 53394
3	15	10	SP R	1	34	716 147540
4	15	10	SP R	1	32	1466 17228
5	17	5	SP R	3	16	2780 102758
6	17	5	SP R	2 999	17 21	994 872660
7	17	5	SP R	9 6	17	9221 42356
8	17	5	SP R	7	18	7310 105370
9	17	5	SP R	14 999	18 20	12638 878552
10	18	5	SP R	6 999	19 23	7490 1059318
11	18	5	SP R	6	19	7289 113472
12	18	5	SP R	2	20	1684 16462
13	18	5	SP R	6	16	6430 33968
14	18	5	SP R	1	17	1068 98694
15	19	7	SP R	1	22 23	1216 84170
16	19	7	SP R	2	26	2374 15880
17	20	5	SP R	7 999	13 15	8898 1326562
18	20	7	SP R	1 999	31 36	1430 1384018
19	20	8	SP R	4 999	38 43	6656 1323186

Clearly, GOST is NP-hard under the duplication cost and the mutation cost. For the NNI distance, the same conclusion also holds.

Theorem 6.1. *The decision version of GOST is NP-complete for the NNI distance.*

Proof. We reduce the problem of computing NNI distance between two trees (see [3]) to GOST. Given are two binary trees T_1 and T_2 with n leaves. By applying an NNI operation to T_1, there are as many as $2n - 2$ different resulting trees. Let T_3 be such a tree, i.e., $d_{NNI}(T_3, T_1) = 1$. We consider the following instance I of GOST:

$$I = \{T_1, T_2, T_3, c_1 = 2, c_2 = 2, c_3 = 1\}.$$

Let S be an optimal species tree for I. Then one can easily verify that $S = T_3$ if and only if $d_{NNI}(T_1, T_2) = d_{NNI}(T_1, T_3) + d_{NNI}(T_3, T_2)$. Note that the NNI distance

$d_{NNI}(T_1, T_2)$ is at most $n \log n + 2n$ [17]. If GOST is solved in polynomial time, we can compute $d_{NNI}(T_1, T_2)$ using an efficient search as follows. For each T_3 such that $d_{NNI}(T_1, T_3) = 1$, compute the optimal species tree S for the instance I defined above. If $S = T_3$, then we use T_3 to replace T_1, compute $d_{NNI}(T_3, T_2)$ inductively, and output $1 + d_{NNI}(T_3, T_2)$. This finishes the reduction and hence the proof. □

Note that the approximation algorithm in Theorem 5.5 cannot be generalized to GOST. Therefore, it is challenging to develop polynomial-time algorithms with good approximation factors for GOST under the various similarity measures.

7. Further research. Further studies on our topics in relation with parametric complexity classes have been carried out recently by Fellows et al. We refer the reader to [6, 7]. We end the paper with a list of open questions.

1. Is the definition of reconciled tree in section 2.4 identical to the one defined by Page (the smallest tree satisfying the three properties listed in section 2.4)?
2. Study the complexity of approximating the problems OST I, OST II, and OST III. Is it possible to develop efficient polynomial-time approximation algorithms for these problems?
3. Develop efficient polynomial-time approximation algorithms for GOST defined in section 6 under the various measures studied here.

REFERENCES

[1] L. Alonso and R. Schott, *Random Generation of Trees*, Kluwer Academic Publishers, Boston, 1995.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[3] B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang, *On distance between phylogenetic trees*, in Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, SIAM, Philadelphia, PA, 1997, pp. 427–436.

[4] O. Eulenstein and M. Vingron, *On the Equivalence of Two Tree Mapping Measures*, Arbeitspapiere der GMD, 936, Bonn, Germany, 1996.

[5] O. Eulenstein, B. Mirkin, and M. Vingron, *Duplication-based measures of difference between gene and species trees*, J. Comput. Bio., 5 (1998), pp. 135–148.

[6] M. Fellows, M. Hallett, C. Korostensky, and U. Stege, *Analogs and duals of the MAST problem for sequences and trees*, in Proceedings of European Symposium on Algorithms (ESA'98), Venice, Italy, 1998, Lecture Notes in Comput. Sci. 1461, Springer-Verlag, Berlin, 1998, pp. 103–114.

[7] M. Fellows, M. Hallett, and U. Stege, *On the multiple gene duplication problem*, in Proceedings of the 9th International Symposium on Algorithms and Computation (ISSAC'98), Taejon, Korea, 1998, Lecture Notes in Comput. Sci. 1533, Springer-Verlag, Berlin, 1998, pp. 347–356.

[8] M. Farach and M. Thorup, *Fast comparison of evolutionary trees*, in Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms, 1994, Arlington, VA, SIAM, Philadelphia, PA, pp. 481–488.

[9] J. Felsenstein, *Phylogenies from molecular sequences: Inference and reliability*, Ann. Review Genet., 22 (1988), pp. 521–561.

[10] W. Fitch, *Distinguishing homologous and analogous proteins*, Syst. Zool., 19 (1970), pp. 99–113.

[11] W. Fitch and E. Margoliash, *Construction of phylogenetic trees*, Science, 155 (1967), pp. 279–284.

[12] Z. Galil and N. Megiddo, *Cyclic ordering is NP-complete*, Theoret. Comput. Sci., 5 (1977), pp. 179–182.

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

[14] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda, *Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences*, Syst. Zool., 28 (1979), pp. 132–163.

[15] R. Guigó, I. Muchnik, and T. Smith, *Reconstruction of ancient molecular phylogeny*, Mol. Phy. and Evol., 6 (1996), pp. 189–213.

[16] D. Harel and R. E. Tarjan, *Fast algorithms for finding nearest common ancestors*, SIAM J. Comput., 13 (1984), pp. 338–355.

[17] M. Li, J. Tromp, and L. Zhang, *Some notes on the nearest neighbor interchange distance*, J. Theoret. Bio., 182 (1996), pp. 463–467.

[18] B. Mirkin, I. Muchnik, and T. Smith, *A biologically meaningful model for comparing molecular phylogenies*, J. Comput. Bio., 2 (1995), pp. 493–507.

[19] G.W. Moore, M. Goodman, and J. Barnabas, *An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets*, J. Theoret. Bio., 38 (1973), pp. 423–457.

[20] M. Nei, *Molecular Evolutionary Genetics*, Columbia University Press, New York, 1987.

[21] J. E. Neigel and J. C. Avise, *Phylogenetic relationship of mitochondrial DNA under various demographic models of speciation*, in Evolutionary Processes and Theory, Academic Press, Orlando, FL, 1986, pp. 515–534.

[22] S. Ohno, *Evolution by Gene Duplication*, Springer-Verlag, Berlin, 1970.

[23] R. Page, *Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas*, Syst. Bio., 43 (1994), pp. 58–77.

[24] R. Page and M. Charleston, *From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem*, Mol. Phy. and Evol., 7 (1997), pp. 231–240.

[25] P. Pamilo and M. Nei, *Relationship between gene trees and species trees*, Mol. Bio. Evol., 5 (1988), pp. 568–583.

[26] J. L. Rémy, *Un procédé itératif de dénombrement d'arbres binaires et son application á leur génération aléatoire*, R.A.I.R.O. Informatique Théorique, 19 (1985), pp. 179–195.

[27] D. F. Robinson, *Comparison of labeled trees with valency trees*, J. Combin. Theory Ser. B, 11 (1971), pp. 105–119.

[28] B. Schieber and U. Vishkin, *On finding lowest common ancestors: Simplification and parallelization*, SIAM J. Comput., 17 (1988), pp. 1253–1262.

[29] M. Steel, *The complexity of reconstructing trees from qualitative characters and subtrees*, J. Classification, 9 (1992), pp. 91–116.

[30] D. Swofford and G. Olsen, *Phylogeny reconstruction*, in Molecular Systematics, D. M. Hillis et al., eds., Sinauer Associates, Sunderland, MA, 1990, pp. 411–501.

[31] N. Takahata, *Gene genealogy in three related populations: Consistency probability between gene and population trees*, Genetics, 122 (1989), pp. 957–966.

[32] M. Waterman and T. Smith, *On the similarity of dendrograms*, J. Theoret. Bio., 73 (1978), pp. 789–800.

[33] C.-I. Wu, *Inference of species phylogeny in relation to segregation of ancient polymorphisms*, Genetics, 127 (1991), pp. 429–435.

[34] L. Zhang, *On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies*, J. Comput. Bio., 4 (1997), pp. 177–188.