On the Complexity of Comparing Evolutionary Trees

Jotun Hein* Tao Jiang† Lusheng Wang‡ Kaizhong Zhang§

Abstract

We study the computational complexity and approximation of several problems arising in the comparison of evolutionary trees. It is shown that the maximum agreement subtree (MAST) problem for three trees with unbounded degree cannot be approximated within ratio $2^{\log^{\delta} n}$ in polynomial time for any $\delta < 1$, unless NP $\subseteq$ DTIME$[2^{\mathrm{polylog}\ n}]$, and MAST with edge contractions for two binary trees is NP-hard. This answers two open questions posed in [1]. For the maximum refinement subtree (MRST) problem involving two trees, we show that it is polynomial-time solvable when both trees have bounded degree and is NP-hard when one of the trees can have an arbitrary degree. Finally, we consider the problem of optimally transforming a tree into another by transferring subtrees around. It is shown that computing the subtree-transfer distance is NP-hard and an approximation algorithm with performance ratio 3 is given.

Key words: Evolutionary tree, phylogeny, compatibility, recombination, computational complexity, approximation algorithm.

*Address: Institute for Genetics and Ecology, Aarhus University 8000 C, Denmark. Email: jotun@hardy.pop.bio.aau.dk

†Supported in part by NSERC Research Grant OGP0046613 and MRC/NSERC CGAT Grant GO-12278. Address: Department of Computer Science, McMaster University, Hamilton, Ont. L8S 4K1, Canada. Email: jiang@maccs.mcmaster.ca

‡Supported in part by NSERC Research Grant OGP0046613. Address: Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ont. L8S 4K1, Canada. Email: lwang@maccs.mcmaster.ca

§Supported in part by NSERC Research Grant OGP0046506. Address: Department of Computer Science, University of Western Ontario, London, Ont. N6A 5B7, Canada. Email: kzhang@csd.uwo.ca

1 Introduction

In the analysis of molecular evolution, the evolutionary history of a set of species is described by an *evolutionary tree* (or *phylogeny*). Let S be a set of species. An evolutionary tree T on S is a *rooted unordered* tree such that the leaves of T are uniquely labeled with the elements in S. The internal nodes are unlabeled and the order among siblings is *insignificant*. Usually we require that each internal node has at least two children. (Note that, evolutionary trees are also often viewed as *unrooted* trees in the literature. All of our results hold for the unrooted version as well.) Reconstructing the correct evolutionary tree for a set of species is one of the fundamental yet difficult problems in evolutionary genetics. Many methods have been proposed based on various criteria. However, these methods do not always produce the same answer. Therefore, it is interesting to design metrics and automatic methods for the comparison of different evolutionary trees on the same set of species. A fruitful approach is to compute a tree that can somehow express the "intersection" of these evolutionary trees.

The notion of a *maximum agreement subtree* (MAST) was first proposed by Finden and Gordon [6]. Given an evolutionary tree T on set S and a subset $A \subseteq S$, the *restriction* of T on A, denoted $T|A$, is an evolutionary tree on set A obtained from T by eliminating the species outside A and the internal nodes with only single child. The latter operation, called *forced contraction*, is illustrated in Figure 1(a). For any two evolutionary trees T_1 and T_2 on set S, an *agreement subtree* (AST) of T_1 and T_2 is a tree T such that for some $A \subseteq S$ $T = T_1|A = T_2|A$. We call $A \subseteq S$ the set of the *agreed* species and $S - A$ the set of the *disagreed* species. A maximum agreement subtree (MAST) of T_1 and T_2 is an AST with the largest number of leaves (*i.e.* the largest number of species have been agreed upon). The notion of AST and MAST can be easily extended to more than two evolutionary trees on the same set of species [1].

The first polynomial-time algorithm for MAST on two trees was given by Steel and Warnow [17]. Their algorithm runs in $O(n^2)$ time for bounded-degree trees and $O(n^{4.5} \log\ n)$ for unbounded-degree trees, where n is number of species. Farach and Thorup recently improved the running time to $O(n^{1.5} \log\ n)$ for unbounded-degree trees and to $O(nc^{\sqrt{\log\ n}})$ for bounded-degree trees [4, 5]. Amir and Keselman [1] considered MAST for several trees. They showed that MAST is polynomial-time solvable for multiple bounded-degree trees and is NP-hard for three trees with unbounded degrees. An algorithm to approximate the *complement* of MAST on multiple unbounded-degree trees with ratio 4 was given. (The complement is to minimize the number of disagreed species instead of the agreed species). They also raised two questions: the approximability of MAST on multiple unbounded-degree trees and, given two trees T_1 and T_2 on set S, how to compute a tree with the largest number of *edges* which is obtainable through a sequence of *edge contractions* from both restrictions $T_1|A$ and $T_2|A$ for some subset $A \subseteq S$. An edge contraction is shown in Figure 1(b) and is also referred to as *deletion of an internal node* in tree edit [20]. Let's call the second problem maximum agreement subtree with edge contractions (MAST-EC). Here we settle these two problems by showing that MAST for three unbounded-degree trees cannot be approximated within ratio $2^{\log^\delta n}$ in polynomial time for any $\delta < 1$, unless NP $\subseteq$ DTIME$[2^{\text{polylog}\ n}]$, and MAST-EC is NP-hard.

The *refinement* of trees is another approach towards the "intersection" of trees, and was originally introduced in the study of the *compatibility* of evolutionary trees [3, 8, 18]. Tree T is said to be a refinement of trees T_1 and T_2 if both T_1 and T_2 can be derived from T through a sequence of edge contractions. Two trees are *compatible* if they have a refinement. Polynomial-time algorithms for the tree compatibility problem have been known for a long time (*e.g.* [8, 18]). It is natural to consider the optimization version of this problem for trees which are not compatible with each other, namely, given trees T_1 and T_2 on set S, find the largest subset $A \subseteq S$ such that $T_1|A$ and $T_2|A$ are compatible [19]. Let's call a refinement of $T_1|A$ and $T_2|A$ a *maximum refinement subtree* (MRST) of T_1 and T_2 and this problem the MRST problem. One can view MRST as a natural counterpart of MAST. We show that MRST can be solved in polynomial time if T_1 and T_2 have degrees bounded by some constant and it becomes NP-hard if one of the trees is allowed to have an arbitrary degree.

When recombination of DNA sequences occurs in an evolution, the history of the evolution cannot be adequately described by a single tree. A recent proposal in attempt to solve this problem is to use a list of evolutionary trees [10, 11]. Each tree corresponds to a *region* of the DNA sequences, and each tree can be obtained from the preceding tree on the list by transferring some subtrees from one place to another. Figure 1(c) shows a subtree-transfer operation, where T_2 is moved to the branch immediately above T_4. Each such operation corresponds to a recombination event. A model for reconstructing such a list of trees based on parsimony has been proposed in [10, 11], which is useful for studying the evolution of viruses, bacteria, multigene families and alleles from nuclear DNA. It is normally not relevant for species trees, since individuals from different species do per definition not mate. This is a restriction on the domain of application of this method, but the body of data where this method is relevant is still very large and includes biological questions of great importance, such as the evolution of HIV. The model can be extended to include gene conversions, which is also a very frequent evolutionary event that can be described phylogenetically as a double recombination event occurring at some distance.

The model in [10, 11] requires the calculation of the subtree-transfer distance between two trees (*i.e.* the minimum number of subtrees we need to transfer). It was left open how to compute this distance. Unfortunately, we can show that computing the distance is NP-hard. We will also give a simple approximation algorithm achieving ratio 3. It turns out that this distance is also connected to the notion of agreement between trees.

The non-approximability of MAST on multiple unbounded-degree trees is given in Section 2. Sections 3 and 4 discuss the complexity of MAST-EC and MRST. Finally, the subtree-transfer distance is considered in Section 5.

2 Non-approximability of MAST on 3 Unbounded-Degree Trees

In this section, we show that the following problem cannot be approximated within ratio $2^{\log^\delta n}$ in polynomial time for any $\delta < 1$, unless NP $\subseteq$ DTIME[$2^{\text{polylog } n}$]. Here, DTIME[$2^{\text{polylog } n}$] denotes the class of problems solvable in $O(2^{\log^c n})$ time for some constant c.

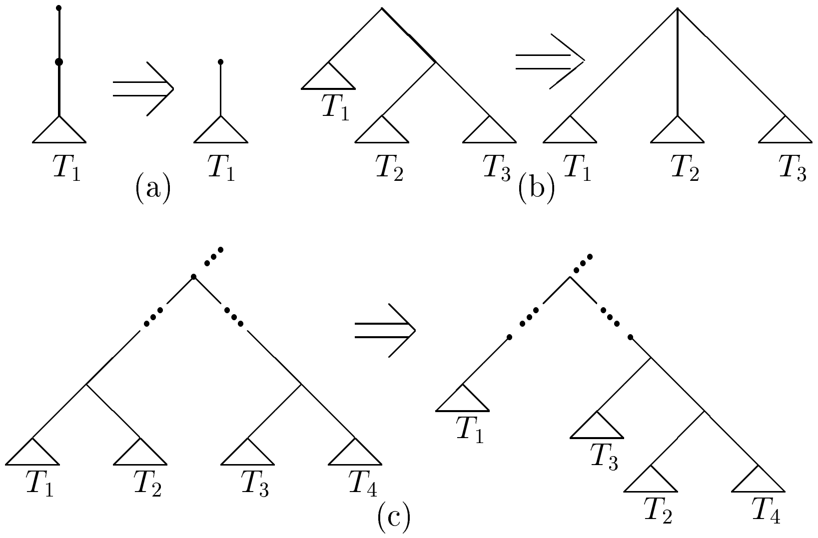

Figure 1: The operations.

Problem: MAST for Three Unbounded Degree Trees.
Instance: Three trees T_1, T_2 and T_3 of arbitrary degrees on set $S = \{s_1, s_2, \dots s_n\}$.
Goal: Find a largest subset $A \subseteq S$ such that $T_1|A = T_2|A = T_3|A$.

The idea of the proof is the self-improvement technique as used in [12]. We first prove that the problem is MAX SNP-hard. Thus it cannot be approximated within ratio $1 + \epsilon$ for some $\epsilon > 0$ unless P = NP [2]. Then we define a product of trees and show that any approximation ratio r for the problem can be improved to $r^{1/k}$. Taking an appropriate k should give the desired bound. In the following, let $c(T_1, T_2, T_3)$ denote the size of a MAST for trees T_1, T_2, T_3.

Lemma 1 *The MAST problem for three unbounded degree trees is MAX SNP-hard.*

Proof. We need to show that a MAX SNP-hard problem L-reduces to MAST for three trees with unbounded degrees. It can be easily verified that Amir and Keselman's construction for the NP-hardness [1] is in fact an L-reduction [16]. For the completeness of the paper, we include the definition of L-reduction and the construction here.

Formally, an L-reduction is defined as follows. Suppose that $\prod_1$ and $\prod_2$ are two optimization problems. We say that $\prod_1$ *L-reduces* to $\prod_2$ if there are two polynomial time algorithms f, g and constants $\alpha, \beta > 0$ such that, for any instance I of $\prod_1$, $f(I)$ forms an instance of $\prod_2$ and

1. $\mathrm{opt}(f(I)) \leq \alpha \cdot \mathrm{opt}(I)$;

2. Given any solution of $f(I)$ with value s_2, the algorithm g produces in polynomial time a solution of I with value s_1 satisfying $|s_1 - \mathrm{opt}(I)| \leq \beta \cdot |s_2 - \mathrm{opt}(f(I))|$.

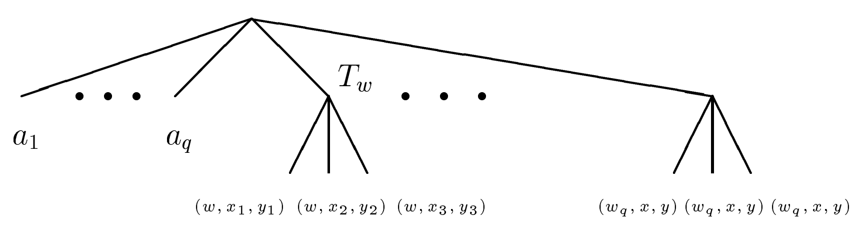

Figure 2: The tree T_1.

Amir and Keselman's reduction is from the following variant of 3-Dimensional Matching which is MAX SNP-hard [14].

Problem: MAX 3DM-B (Maximum Bounded 3-Dimensional Matching).

Instance: A set $M \subseteq W \times X \times Y$ of ordered triples where W, X and Y are disjoint. Each element of $W \cup X \cup Y$ appears in at most B triples of M.

Goal: Find the largest subset $M' \subseteq M$ such that no two elements of M' agree in any coordinate.

Given an instance of 3DM-B, $M \subseteq W \times X \times Y$, we construct three evolutionary trees T_1, T_2 and T_3. Let $|W| = |X| = |Y| = q$ and $|M| = p$. The root of each T_i has $2q + 1$ children. The first $q + 1$ children are leaves labeled with new symbols $a_1, a_2, \ldots, a_{q+1}$. Each of the last q children of T_1 is a subtree T_w corresponding to an element $w \in W$. T_w has its leaves labeled with the triples of the form (w, x, y), where $x \in X$ and $y \in Y$. T_2 and T_3 are constructed in a similar way. (See Figure 2.)

Note that, each T_w has at most B leaves, where B is an constant. The new symbols $a_1, a_2, \ldots, a_{q+1}$ ensure that the three roots of T_1, T_2 and T_3 form the root of an MAST. Since all the leaves of T_w contain the same element w in their labels, at most one of the leaves in T_w can be preserved in an AST. The same argument holds for T_x and T_y. Therefore, M has a 3-dimensional matching of size k if and only if T_1, T_2, T_3 have AST of size $k + q + 1$.

Since each element of $W \cup X \cup Y$ appears in at most B triples of M, there is always a 3-dimensional matching of size at least q/B. Therefore, $c(T_1, T_2, T_3) = \mathrm{opt}(M) + q + 1 \leq \mathrm{opt}(M) + 2q$ $\leq \mathrm{opt}(M) + 2B\mathrm{opt}(M) = (2B + 1)\mathrm{opt}(M)$.

Moreover, given an AST of T_1, T_2, T_3 of size $s_2 = q + 1 + k$, we can find in polynomial time a 3-dimensional matching of size $s_1 = k$. Observe that $\mathrm{opt}(M) - s_1 = 1 + q + \mathrm{opt}(M) - (1 + q + s_1)$ $= 1 + q + \mathrm{opt}(M) - s_2 = c(T_1, T_2, T_3) - s_2$.

This shows that the above reduction is actually an L-reduction. ■

Now, we need define the *product* of two evolutionary trees. Let T_1 and T_2 be two evolutionary trees on sets S_1 and S_2 respectively. where S_1 and S_2 are the sets of labels for these two trees. For each label $s \in S_1$, let $T_2(s)$ denote the tree obtained from T_2 by replacing each label $s' \in S_2$ with

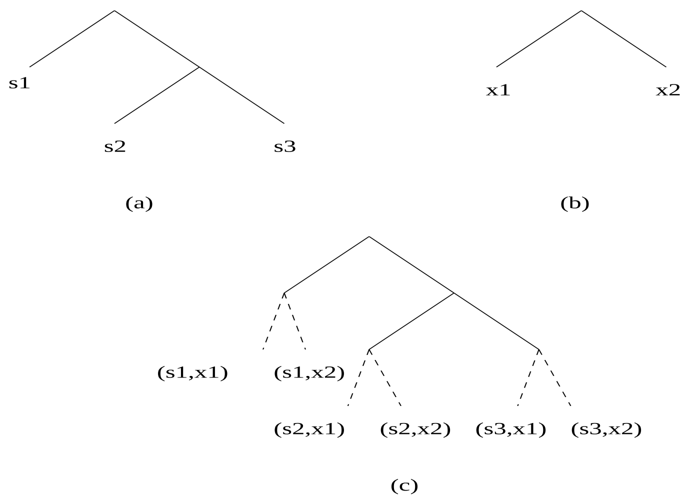

Figure 3: (a) The tree T_1. (b) The tree T_2. (c) The tree $T_1 \times T_2$.

a new label (s, s'). The product of T_1 and T_2, denoted $T_1 \times T_2$, is obtained from T_1 by replacing each leaf labeled s with the tree $T_2(s)$. (See figure 3.) For any tree T, we define $T^2 = T \times T$ and $T^{k+1} = T \times T^k$. The following lemma allows us to improve an approximation ratio for MAST by taking the product.

Lemma 2 *Let T_1, T_2 and T_3 be the three evolutionary trees. Then $c(T_1^{k+1}, T_2^{k+1}, T_3^{k+1}) \geq c(T_1, T_2, T_3) \cdot c(T_1^k, T_2^k, T_3^k)$. Moreover, given an AST of size c for $T_1^{k+1}, T_2^{k+1}, T_3^{k+1}$, we can find in polynomial time an AST of size c_1 for T_1, T_2, T_3 and an AST of size c_2 for T_1^k, T_2^k, T_3^k such that $c_1 \cdot c_2 = c$.*

Proof. Let S be the set of labels in T_1, T_2, T_3. It is easy to see that if T and T' are the MAST's for T_1, T_2, T_3, and T_1^k, T_2^k, T_3^k respectively, then $T \times T'$ is an agreement subtree for $T_1^{k+1}, T_2^{k+1}, T_3^{k+1}$. Thus, $c(T_1^{k+1}, T_2^{k+1}, T_3^{k+1}) \geq c(T_1, T_2, T_3) \cdot c(T_1^k, T_2^k, T_3^k)$. Suppose that we are given an AST T of size c for $T_1^{k+1}, T_2^{k+1}, T_3^{k+1}$. For each label $s \in S$ such that (s, s') appears in T for some $s' \in S^k$, we can identify an agreement subtree T'' of $T_{1,s}$, $T_{2,s}$, and $T_{3,s}$ in T. Let c_1 be the size of T''. Each of the leaves in T'' corresponds to an agreement subtree for T_1^k, T_2^k, T_3^k. Without loss of generality, assume that all such subtrees have the same size c_2 (otherwise we can improve c). Then, we have $c_1 \cdot c_2 = c$. ■

By the same argument as in [12], we have the following theorem.

Theorem 3 *For any constant $\delta < 1$, MAST for three unbounded-degree trees cannot be approximated within ratio $2^{\log^\delta n}$ in polynomial time, unless* NP $\subseteq$ DTIME$[2^{\text{polylog } n}]$.

Proof. Suppose that for some constant $\delta < 1$, MAST can be approximated with ratio $2^{\log^\delta n}$ in time $O(n^d)$ for some constant d. For any fixed $\epsilon > 0$, let

$$k = \left(\frac{\log^\delta n}{\log 1 + \epsilon}\right)^{\frac{1}{1-\delta}}.$$

Given an instance T_1, T_2, T_3 of MAST of size n, we can blow it up k times to obtain an instance T_1^k, T_2^k, T_3^k of size at most n^k. By the assumption, an approximate solution of T_1^k, T_2^k, T_3^k with ratio $2^{\log^\delta n^k}$ can be found in time

$$O(n^{kd}) = O(2^{kd\log\ n}) = O(2^{\text{polylog}\ n}).$$

By Lemma 2, such an approximate solution of T_1^k, T_2^k, T_3^k implies an approximate solution of T_1, T_2, T_3 with ratio

$$(2^{\log^\delta n^k})^{\frac{1}{k}} \leq 1 + \epsilon.$$

It then follows from Lemma 2 and the results in [2] that NP $\subseteq$ DTIME[$2^{\text{polylog}\ n}$]. ■

3 Agreement Subtrees with Edge Contractions

A natural extension of the agreement subtree approach is to allow the application of edge contractions in the formation of an "agreement" of the given trees. Intuitively, an edge contraction "loosens" the structure of a tree and thus increases the chance of having an agreement. For example, in the extreme case, we can contract all the edges in all trees to end up with a star. However, the obtained star contains little information about the evolutionary history. Therefore, it is desirable to contract a small number of edges yet to have a large number of the agreed species. This is the intuition behind the MAST-EC problem first proposed in [1]. Here, we show that MAST-EC on bounded-degree trees is NP-hard by a reduction from Exact Cover by 3-Sets [7].

Problem: Exact Cover by 3-Sets.

Instance: A collection C of subsets of a finite set S where every $c \in C$ contains three elements and every element $s \in S$ is contained in at most 3 subsets in C.

Goal: Find an exact covering $C' \subseteq C$ of S, *i.e.* a collection of mutually disjoint sets whose union equals S.

Theorem 4 *MAST-EC is NP-hard even if the given trees have bounded degree.*

Proof. Given an instance of Exact Cover by 3-Sets, let the set $S = \{s_1, s_2, \ldots, s_m\}$, where $m = 3q$ and $C = \{C_1, C_2, \ldots, C_n\}$, where each $C_i = \{t_{i,1}, t_{i,2}, t_{i,3}\}$, $t_{i,j} \in S$. Without loss of generality, we assume that $n > q$.

Two trees T and $\hat{T}$ are constructed as in Figure 4 and Figure 5. The top part is a binary tree whose actual structure is insignificant. (We need this part to get around the degree bound.) In order to make this part insignificant in the following calculation, introduce a large enough factor $f = 2(n+m)$. Each element $c_{i,j}$, $j = 1, 2, 3$, of C_i corresponds to a subtree $T_{i,j}$ as shown in Figure 6. In tree T, each element $s_i \in S$ corresponds to a subtree T_i as shown in Figure 7. Every triple of subtrees $T_{i,j}$ ($j = 1, 2, 3.$) is connected to the top part by a path of length $5f$, which has $5f$ internal nodes (not including the root of the subtree) and $10f$ edges (including the edges connecting $x_{i,j}$'s.). Call such a path a long chain, denoted by P_i. In tree $\hat{T}$, there are n corresponding long chains each of which contains $5f - 1$ internal nodes, and $10f - 2$ edges. (See Figure 5.)

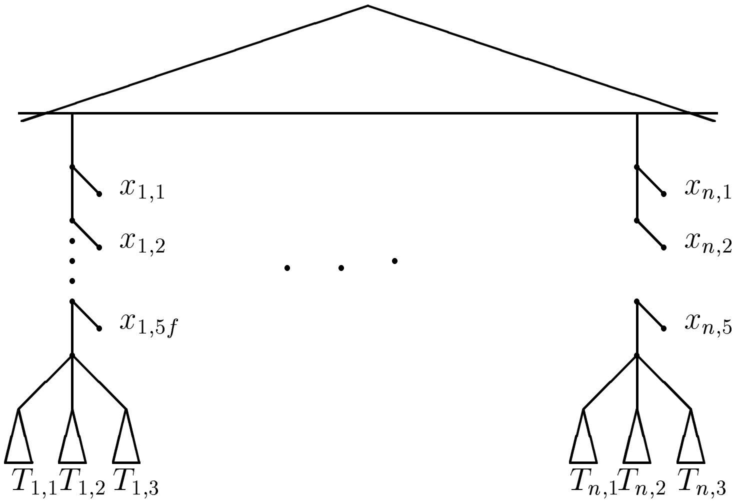

Figure 4: The tree T constructed from $C = \{C_1, C_2, \ldots, C_n\}$, where each subtree $T_{i,j}$ corresponds to a $c_{i,j} \in C_i$, $j = 1, 2, 3$.

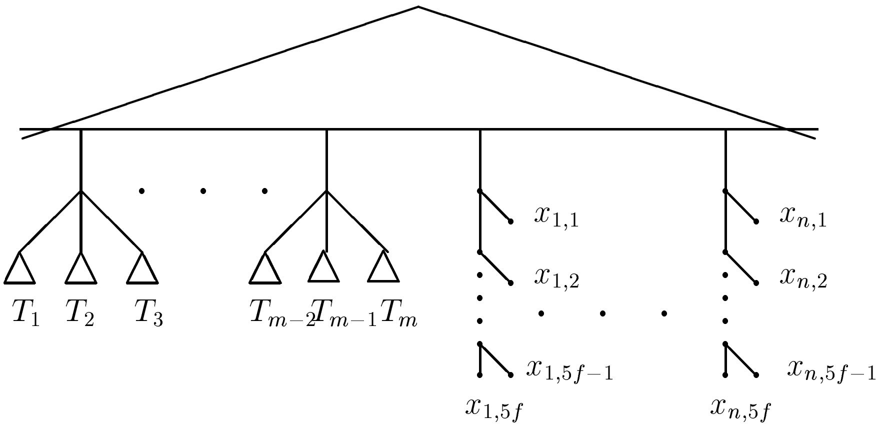

Figure 5: The tree $\hat{T}$. Each subtree T_i corresponds to an element $s_i \in S$ and each of the n chains of length $5f$ corresponds to a subset $C_i \in C$.

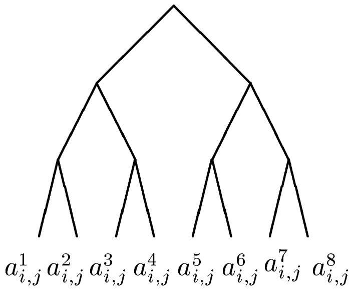

Figure 6: The subtree $T_{i,j}$ corresponding to $c_{i,j} \in C_i$ ($f = 8$ in this case).

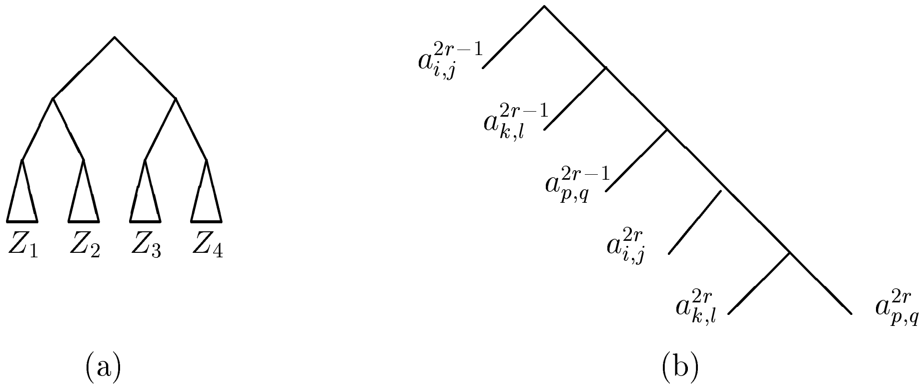

Figure 7: Suppose that $c_{i,j}$, $c_{k,l}$ and $c_{p,q}$ are the three occurrences of s_i. (a) The subtree T_i corresponding to s_i. (b) The subtree Z_r. (Again, $f = 8$.)

We will show that C has an exact cover of S if and only if there is a tree T' with at least $3(2f-2)q+(10f-2)(n-q)+5fq$ edges such that, for some subset A of labels, T' can be obtained using edge contractions from both restrictions $T|A$ and $\hat{T}|A$.

(if) Given an exact cover $\{C_{i_1}, \ldots, C_{i_q}\} \subseteq C$ of S, there is a corresponding set of subtrees $\{T_{i_l,j} | 1 \leq l \leq q,\ 1 \leq j \leq 3\}$ in T. We can obtain an agreement subtree with edge contraction (AST-EC) T' which contains the subtrees $\{T_{i_l,j} | 1 \leq l \leq q,\ 1 \leq j \leq 3\}$ and the long chains P_r ($r \in \{1, 2, \ldots, n\} - \{i_1, i_2, \ldots, i_q\}$). Note that, each $T_{i_l,j}$ contributes $2f-2$ edges, and each P_r contributes $10f-2$ edges. Moreover, we eliminate the internal nodes of the q long chains $P_{i_1}, \ldots, P_{i_q}$, each of which contains $5f$ labels and each label contributes one edge in the AST-EC T'. Thus, T' has at least $3(2f-2)q+(10f-2)(n-q)+5fq$ edges.

(only if) Suppose we have an AST-EC T' of size at least $3(2f-2)q+(10f-2)(n-q)+5fq$. Note that, the constructions of $T_{i,j}$'s in T and T_i's in $\hat{T}$ imply that without decreasing the number of edges, we can modify T' such that if a label in a $T_{i,j}$ appears in T', then the whole subtree $T_{i,j}$ appears in T'. Furthermore, in order to keep $T_{i,j}$ in T', we have to eliminate $5f$ internal nodes in the long chain P_i. On the other hand, keeping a long chain P_i in T' means that we cannot keep any of T_{ij} ($j = 1, 2, 3$) in T'. Keeping a long chain will contribute $10f-2$ edges. This means that T' has to have some T_{ij} otherwise the number of edges in T' would be at most $(10f-2)n+n+m$ which is less than $3(2f-2)q+5fq+(10f-2)(n-q)$. If T_{ij} is in T' the other two subtrees $T_{ik}(k = \{1, 2, 3\} - \{j\})$ have to be in T' in order to increase the number of edges in T'. Keeping a tripe will contribute $3(2f-2)+5f$ edges. Thus in order to obtain $3(2f-2)q+5fq+(10f-2)(n-q)$ edges, we have to preserve q tripes in T', which should give an exact cover of S. ■

A variant of MAST-EC is to construct a tree maximizing the the number of internal nodes instead of edges. Unfortunately, a simple modification of the above proves that this variant is also NP-hard.

4 Maximum Refinement Subtrees

The MRST problem can be reformulated as another problem, called the *alignment of trees*, recently introduced in [13]. Informally, an alignment $\mathcal{A}$ of trees T_1 and T_2 is obtained by first inserting new unlabeled nodes into T_1 and T_2 such that the two resulting trees T_1' and T_2' have the same structure, *i.e.* they are identical if the labels are ignored, and then *overlaying* T_1' on T_2'. Here inserting a new node u under an existing internal node v means we make u the parent of some children of v and then u a child of v [20]. Thus, each node in the alignment $\mathcal{A}$ is labeled with a pair of (possibly null) symbols, one from T_1' and the other from T_2'. A score scheme is defined for each pair of labels. The *value* of the alignment $\mathcal{A}$ is the sum of the scores of all pairs of opposing labels. An optimal alignment is one that maximizes the value over all possible alignments. The notion of alignment of trees was originally proposed as an alternative way of measuring the similarity of trees representing RNA secondary structures [13].

To formulate MRST as an alignment of trees problem, we need define the scores as follows: 1 for an identical pair of (non-null) labels and 0 otherwise. It is easy to see that in any alignment $\mathcal{A}$

of T_1 and T_2, the set of nodes with identical pairs of opposing (non-null) labels induces a refinement subtree T of T_1 and T_2. The value of $\mathcal{A}$ is exactly the size of the subset of labels appearing in T. Therefore, an optimal alignment of T_1 and T_2 gives an MRST of T_1 and T_2.

In [13], a polynomial-time algorithm to align (unordered) trees with bounded degree is given. We hence have the following theorem.

Theorem 5 *MRST can be computed in polynomial time if both trees are of bounded degree.*

On the other hand, alignment of trees is NP-hard when one of the trees is allowed to have an arbitrary degree [13]. A slight modification of the proof of this result gives the next theorem. A weaker result can be found in [9].

Theorem 6 *MRST is NP-hard if one of the given trees can have an arbitrary degree.*

Proof. The reduction is again from Exact Cover by 3-Sets. Let $S = \{s_1, \ldots, s_m\}$ and $C = \{C_1, \ldots, C_n\}$, where $C_i = \{c_{i,1}, c_{i,2}, c_{i,3}\}$, be an instance of this problem. Assume $m = 3q$.

We construct two trees as shown in Figures 8 and 9. Note that, each $s_i \in S$ appears at most three times in C_i's. For each $i = 1, \ldots n$ and $j = 1, 2, 3$, let $a_{i,j}$ and $a'_{i,j}$ be two labels corresponding to the $c_{i,j}$. The set of labels for the two constructed trees is $\{a_{i,j}, a'_{i,j} | i = 1, \ldots, n,\ j = 1, 2, 3\} \cup \{s_{i,j} | i = 1, \ldots, n,\ j = 1, 2, 3, 4, 5\}$. The tree T is basically a chain of n subtrees T_1, T_2, ..., T_n. (See Figure 8 (a).) Each T_i contains a stem of 5 labels as its upper segment and a subtree of 6 labels as its lower segment. (See Figure 8 (b).) The lower segment of T_i corresponds to C_i in C. The tree $\hat{T}$ contains m subtrees $Z_1, \ldots, Z_m$ and n stems of 5 labels as shown in Figure 9. Each Z_i in $\hat{T}$ contains 2, 4, or 6 labels corresponding to the one, two, or three occurrences of $s_i \in S$.

Now, we want to show that C contains an exact cover of S if and only if T and $\hat{T}$ have a refinement with $6q + 5(n - q)$ labels. Note that, for each subtree T_i, either its entire upper segment or its lower segment can be included in any refinement of T and $\hat{T}$. Including the upper segment contributes 5 labels, whereas including the lower segment contributes 6 labels. Thus, an MRST of T and $\hat{T}$ includes as many lower segments as possible. Therefore, we can conclude that C contains an exact cover of S if and only if T and $\hat{T}$ have a refinement with $6q + 5(n - q)$ labels. ■

5 The Subtree-Transfer Distance

When *recombination* of DNA sequences occurs in an evolution, two sequences meet and generate a new sequence, consisting of genetic material taken left of the recombination point from the first sequence and right of the point from the second sequence [15, 10, 11]. Figure 10 (a) demonstrates a recombination. From a phylogenetic viewpoint, before the recombination, the ancestral material on the present sequence was located on two sequences, one having all the material to the left of the recombination point and another having all the material to the right of the breaking point.

In Figure 10 (b), it is assumed that a recombination has occurred in the generation of some ancestor r of s_2. Thus, the evolutionary history can no longer be described by a single tree. The

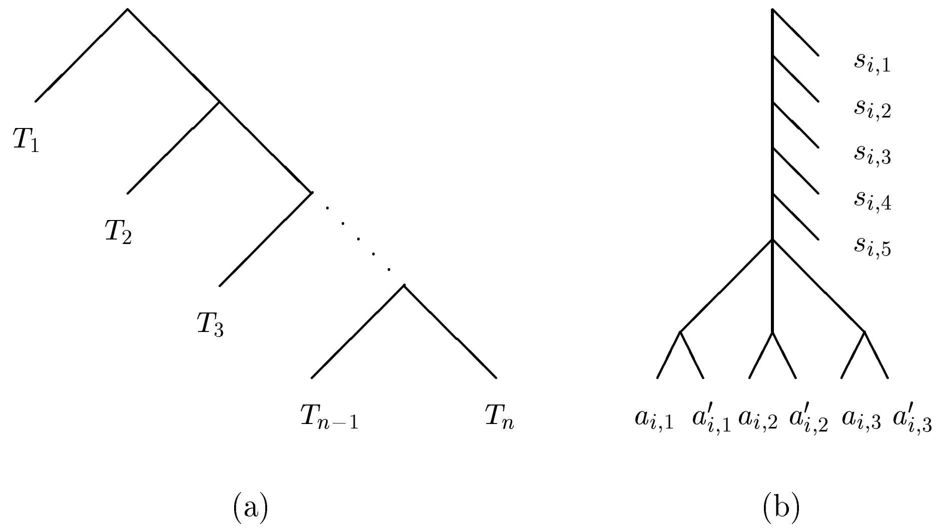

Figure 8: (a) The tree T. (b) The subtree T_i.

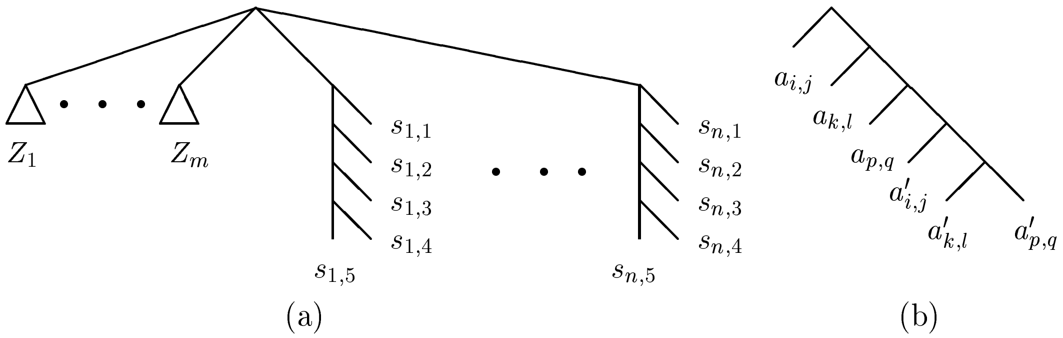

Figure 9: Suppose that $c_{i,j}$, $c_{k,l}$ and $c_{p,q}$ are the three occurrences of s_i. (a) The tree $\hat{T}$. (b) The subtree Z_i.

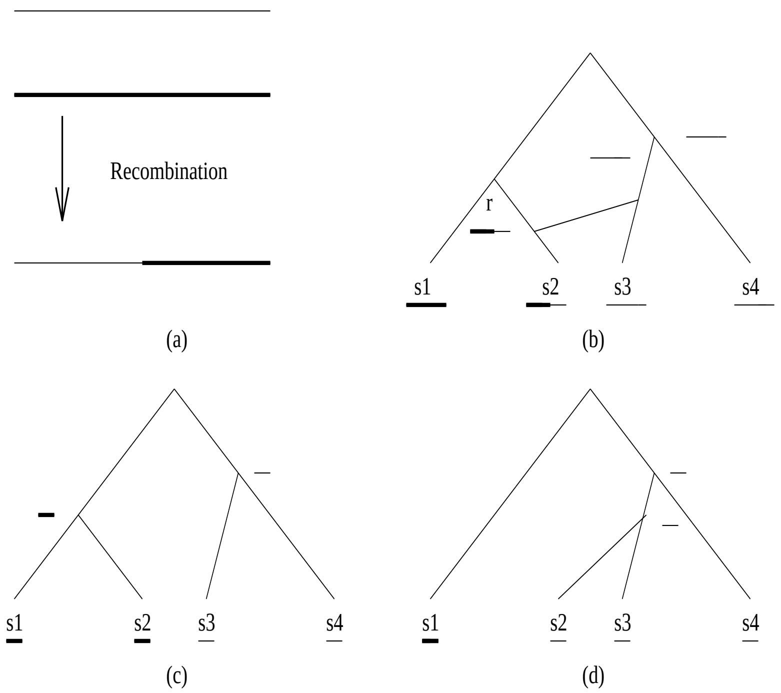

Figure 10: (a) A recombination. (b) A recombination occurs at an ancestor r of s_2. (c) The evolutionary tree corresponding to the left region. (d) The evolutionary tree corresponding to the right region.

recombination event partitions the sequences into two *neighboring* regions. The history for the left region could be described by the evolutionary tree in Figure 10 (c), which is obtained by going back in time above the recombination following the left-going edge, while the history for the right region could be described by the tree in Figure 10 (d), which is obtained by following the right-going edge. The recombination makes the two evolutionary trees describing neighboring regions differ. However, two neighbor trees cannot be arbitrarily different, one must be obtainable from the other by a subtree-transfer operation. When more than one recombination occurs, one can describe an evolutionary history using a list of evolutionary trees, each corresponds to some region of the sequences and each can be obtained by several subtree-transfer operations from its predecessor [11]. The computation of subtree-transfer distance is useful in reconstructing such a list of trees based on parsimony [10, 11].

In this section, we show that computing the subtree-transfer distance between two evolutionary trees is NP-hard and give an approximation algorithm with performance ratio 3. Before we prove the results, it is again convenient to reformulate the problem. Let T_1 and T_2 be two evolutionary trees on set S. An *agreement forest* of T_1 and T_2 is any forest which can be obtained from both T_1 and T_2 by cutting k edges (in each tree) for some k and applying forced contractions in each resulting component trees. Define the size of a forest as the number of components it contains. Then the *maximum agreement forest* (MAF) problem is to find an agreement forest with the *smallest* size. The following lemma shows that MAF is really equivalent to computing the subtree-transfer distance.

Lemma 7 *The size of a MAF of T_1 and T_2 is one more than their subtree-transfer distance.*

The lemma can be proven by a simple induction on the number of leaves. Intuitively, the lemma says that the transfer operations can be broken down into two stages: first we cut off the subtrees to be transferred from the rest in T_1 (not worrying where to put them), then we assemble them appropriately to obtain T_2. This separation will simplify the proofs.

5.1 The NP-hardness

Theorem 8 *It is NP-hard to compute the subtree-transfer distance between two binary trees.*

Proof. The reduction is again from Exact Cover by 3-Sets. Let $S = \{s_1, s_2, \ldots s_m\}$ be a set and $C_1, \ldots, C_n$ be an instance of this problem. Assume $m = 3q$.

The tree T_1 is formed by inserting n subtrees $A_1, \ldots, A_n$ into a chain containing $2n + 2m$ leaves $x_1, \ldots, x_{2n}, y_1, \ldots, y_{2m}$ uniformly. (See Figure 11(a).) Each A_i corresponds to $C_i = \{c_{i,1}, c_{i,2}, c_{i,3}\}$, and has 9 leaves as shown in Figure 11(b). Suppose that $c_{j,j'}$, $c_{k,k'}$ and $c_{l,l'}$ are the three occurrences of an $s_i \in S$ in C. Then in T_2, we have a subtree B_i as shown in Figure 12(a). For each C_i, we also have a subtree D_i in T_2 as shown in Figure 12(b). The subtrees are arranged as a linear chain as shown in Figure 12(c).

Note that, each adjacent pair of subtrees A_i and A_{i+1} in T_1 is separated by a chain of length 2 which also appears in T_2. Thus, to form a MAF of T_1 and T_2, our best strategy is clearly to cut

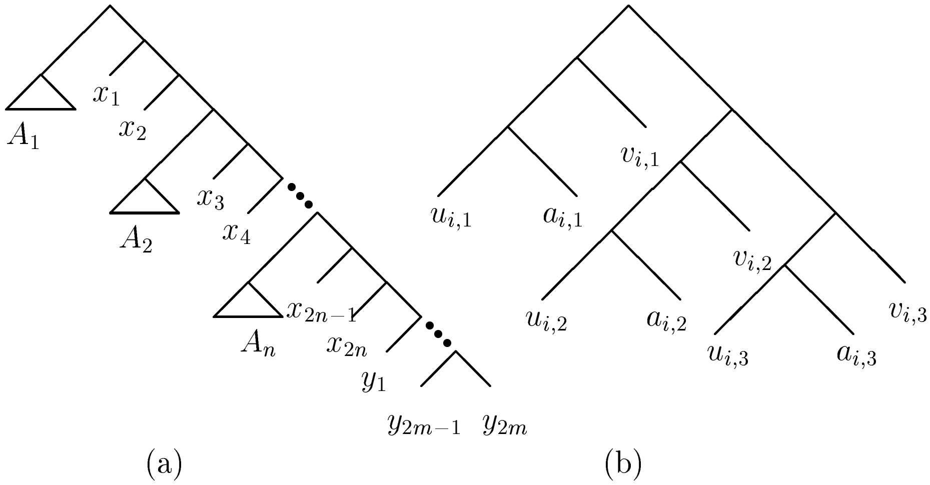

Figure 11: (a) The tree T_1. (b) The subtree A_i.

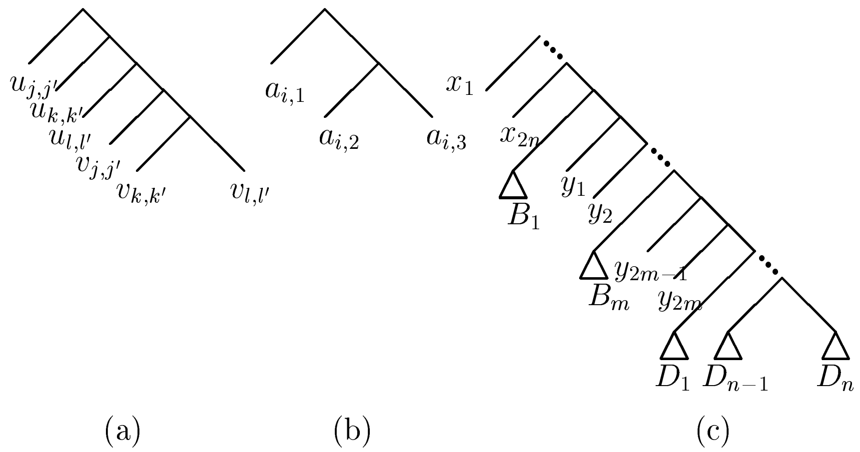

Figure 12: (a) The subtree B_i. (b) The subtree D_i. (c) The tree T_2.

off $A_1, A_2, \ldots, A_n$ in T_1 and similarly cut off $B_1, B_2, \ldots, B_m$ in T_2. This then forces us to cut off $D_1, D_2, \ldots, D_n$ in T_2. Now in each A_i, we can either cut off the leaves $u_{i,1}, v_{i,1}, u_{i,2}, v_{i,2}, u_{i,3}, v_{i,3}$ to form a subtree containing three leaves $a_{i,1}, a_{i,2}, a_{i,3}$ (yielding $6+1=7$ components totally), or we can cut off $a_{i,1}$, $a_{i,2}$, and $a_{i,3}$. In the second case, we will be forced to also cut links between the three subtrees containing leaves $\{u_{i,1}, v_{i,1}\}$, $\{u_{i,2}, v_{i,2}\}$ and $\{u_{i,3}, v_{i,3}\}$ respectively, as the B_i's are already separated. Hence in this case the best we can hope for is $3+3=6$ components (if we can keep all three 2-leaf subtrees in the agreement forest).

We now formally show that C has an exact cover of S if and only if T_1 and T_2 have an agreement forest of size $1+6q+7(n-q) = 7n-q+1$.

(if) Suppose we are given an exact cover $\{C_{i_1}, C_{i_2}, \ldots, C_{i_q}\} \subseteq C$ of S. For each A_{i_l} ($l = 1, 2, \ldots q$) in T_1, we cut off $a_{i_l,1}$, $a_{i_l,2}$, $a_{i_l,3}$, and the three subtrees containing leaves $\{u_{i_l,1}, v_{i_l,1}\}$, $\{u_{i_l,2}, v_{i_l,2}\}$ and $\{u_{i_l,3}, v_{i_l,3}\}$. Correspondingly, we can obtain the 6 components from T_2. Note that, each B_i in T_2 can contribute at most one subtree containing $\{u_{i_l,j}, v_{i_l,j}\}$ ($j = 1, 2, 3$). In this way, each A_{i_l} creates 6 components in the agreement forest. For each of the other $n-q$ subtrees A_r ($i \in \{1, 2, \ldots n\} - \{i_1, \ldots i_q\}$), we can cut off the leaves $u_{r,1}, v_{r,1}, u_{r,2}, v_{r,2}, u_{r,3}, v_{r,3}$ and the subtree containing the three leaves $a_{r,1}, a_{r,2}, a_{r,3}$. Correspondingly, we can obtain the 7 components from T_2. Therefore, the total number of components in the agreement forest is $1+6q+7(n-q) = 7n-q+1$.

(only if) Suppose we have an agreement forest of size $1+6q+7(n-q) = 7n-q+1$. From the previous discussion, we know that

1. Each A_i in T_1 contributes at least 6 components.
2. Among the n subtrees A_i's, at most q A_i's can contribute 6 components each. Moreover, these A_i's correspond to disjoint C_i's in C.

Thus, the agreement forest of size $7n-q+1$ should give us an exact cover of S. ■

5.2 An Approximation Algorithm of Ratio 3

Our basic idea is to deal with a pair of sibling leaves a, b in the first tree T_1 at a time. If the pair a and b are siblings in the second tree T_2, we replace this pair with a new leaf labeled by (a, b) in both trees. Otherwise, we will cut T_2 until a and b become siblings or separated. Eventually both trees will be cut into the same forest. Five cases need be considered. Figure 13 illustrate the first four cases. The last case (Case (v)) is that a and b are also siblings in T_2.

The approximation algorithm is given in Figure 14. The variable N records the number of components (or the number of cuts plus 1).

Theorem 9 *The approximation ratio of the algorithm in Figure 14 is 3, i.e., it always produces an agreement forest of size at most three times the size of a MAF for T_1 and T_2.*

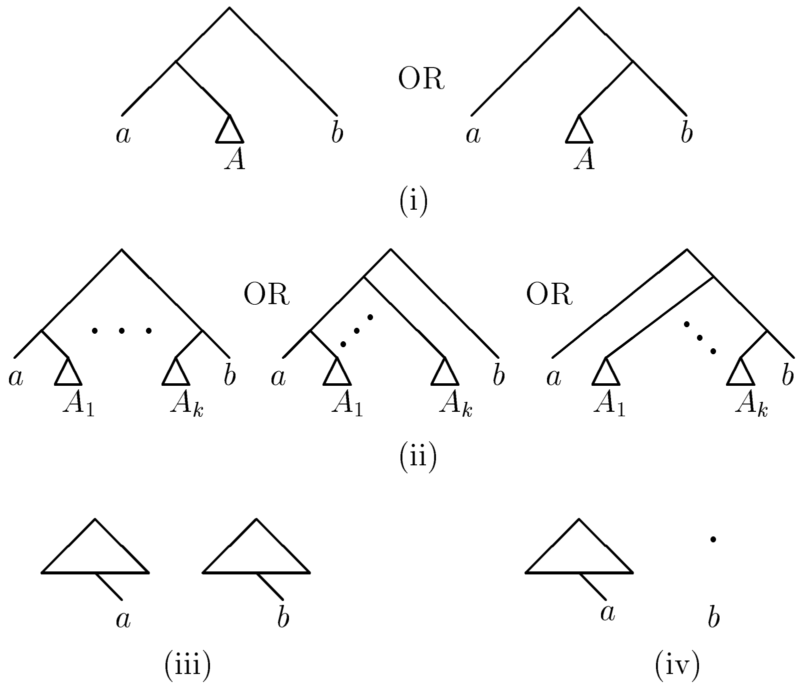

Figure 13: The first four cases of a and b in T_2.

Input: T_1 and T_2.
0. $N := 1$;
1. For a pair of sibling leaves a, b in T_1,
consider how they appear in T_2 and cut the trees:
Case (i): Cut off the middle subtree A in T_2; $N := N + 1$;
Case (ii): Cut off a and b in both T_1 and T_2; $N := N + 2$;
Case (iii): Cut off a and b in both T_1 and T_2; $N := N + 2$;
Case (iv): Cut off b in T_1;
Case (v): Replace this pair with a new leaf labeled (a, b) in both T_1 and T_2;
2. If some component in the forest for T_1 has size larger than 1, repeat Step 1.
Output: The forest and N.

Figure 14: The approximation algorithm of ratio 3.

Proof. (Sketch) We consider the number of edges cut in an agreement forest and show that the above algorithm cuts at most three times as many the edges cut by a MAF. To establish the approximation bound, the basic idea is to consider a MAF and charge the edges cut by the algorithm to the edges cut by the MAF, and make sure that each MAF edge is charged at most three edges. For convenience, we will refer to an edge according to its lower end.

We need a lemma which establishes that the algorithm is always optimal in Case (i).

Lemma 10 *There exists a MAF F which cuts all the edges cut by the algorithm in Case (i).*

We only have to consider Cases (i)-(iii) because Case (iv) repeats an old cut. Let's fix the MAF F. We charge the edges as follows. In Case (i), each edge cut is simply charged to itself. Case (ii) is the most interesting. If at least one of the edges above a and b are cut by F, then we simply charge these two edges cut by the algorithms to the "correct" edge(s) cut by F. So each edge is charged with a cost of at most 2. Otherwise all the edges above the subtrees $A_1, \ldots, A_k$ $(k \geq 2)$ must be cut by F and we charge the two edges above a and b to these edges (each charged a cost of $\frac{2}{k} \leq 1$). Note that, only edges cut in Case (i) create components of size bigger than 1. Thus, if Case (iii) occurs, then a and b must belong to different components in F and hence F must cut at least one of the edges above a and b in T_1 (and thus in T_2 as well) to disconnect them. So we can charge the two edges above a and b to the "correct" one(s) cut by F (each charged a cost of at most 2 edges).

It is easy to see that each edge cut by F is charged at most twice. Moreover, if an edge is charged twice, the first time it is charged must be in the second subcase of Case (ii). The second time can be in either Case (i), Case (iii), or the first subcase of Case (ii). Hence, each such edge is charged a cost of at most $1 + 2 = 3$ edges cut by the algorithm. That is, the algorithm cuts at most three times as many edges cut by F. ■.

It will be interesting to improve the approximation ratio. On the other hand, the NP-hardness proof can be easily strengthened to work for MAX SNP-hardness. Thus there is no hope for a polynomial-time approximation scheme for this problem.

Acknowledgement. We sincerely thank the referees for many helpful comments and suggestions.

References

[1] A. Amir and D. Keselman, Maximum agreement subtree in a set of evolutionary trees - metrics and efficient algorithms, *Proc. 35th IEEE Symp. Found. Comp. Sci.*, 1994.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and hardness of approximation problems, *Proc. 33rd IEEE Symp. Found. Comp. Sci.*, 14-23, 1992.

[3] G. Estabrook, C. Johnson and F. McMorris, A mathematical foundation for the analysis of cladistic character compatibility, *Math. Biosci.* 29, 181-187, 1976.

[4] M. Farach and M. Thorup, Fast comparison of evolutionary trees, *Proc. 5th Annual ACM-SIAM Symp. Discrete Algorithms,* 1994.

[5] M. Farach and M. Thorup, Optimal evolutionary tree comparison by sparse dynamic programming, *Proc. 35th IEEE Symp. Found. Comp. Sci.*, 1994.

[6] C. Finden and A. Gordon, Obtaining common pruned trees, *Journal of Classification* 2, 255-276, 1985.

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

[8] D. Gusfield, Efficient algorithms for inferring evolutionary trees, *Networks* 21, 19-28, 1991.

[9] A.M. Hamel and M.A. Steel, Finding a maximum compatible tree is NP-hard, *Technical Report* No. 114, October 1994, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

[10] J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, *Math. Biosci.* 98, 185-200, 1990

[11] J. Hein, A heuristic method to reconstruct the history of sequences subject to recombination, *Journal Molecular Evolution* 36, 396-405, 1993.

[12] T. Jiang and M. Li, On the approximation of shortest common supersequences and longest common subsequences, *SIAM J. Comput.* 24, 1122-1139, 1995.

[13] T. Jiang, L. Wang and K. Zhang, Alignment of trees - an alternative to tree edit, *Theoretical Computer Science* 143-1, 137-148, 1995.

[14] V. Kann, Maximum bounded 3-dimensional matching is MAX SNP-complete, *Information Processing Letters* 37, 27-35, 1991.

[15] J. Kececioglu and D. Gusfield, Reconstructing a history of recombinations from a set of sequences, *Proc. 5th Annual ACM-SIAM Symp. Discrete Algorithms,* 1994.

[16] C.H. Papadimitriou and M. Yannakakis, Optimization, Approximation, and complexity classes, *Journal of Computer and System Sciences* 43, 425-440, 1991.

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

[18] T. Warnow, Tree compatibility and inferring evolutionary history, *J. of Algorithms* 16, 388-407, 1994.

[19] T. Warnow, Private communication, 1994.

[20] K. Zhang and D. Shasha, Simple fast algorithms for the editing distance between trees and related problems, *SIAM J. Comput.* 18, 1245-1262, 1989.