International Journal of Foundations of Computer Science

SUBTREE TRANSFER DISTANCE FOR DEGREE-D PHYLOGENIES *

WING-KAI HON TAK-WAH LAM SIU-MING YIU†
Department of Computer Science
The University of Hong Kong, Hong Kong

and

MING-YANG KAO‡
Department of Computer Science, Northwestern University
Evanston, Illinois 60201, USA

and

WING-KIN SUNG§
Department of Computer Science
National University of Singapore, Singapore

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

The subtree transfer (STT) distance is one of the distance metric for comparing phylogenies. Previous work on computing the STT distance considered degree-3 trees only. In this paper, we give an approximation algorithm for the STT distance for degree-d trees with arbitrary d and with generalized STT operations. Also, some NP-hardness results related to STT distance are presented.

Keywords: Evolutionary trees; approximation algorithms; NP-hardness

1. Introduction

Phylogenies are rooted trees whose leaves are labeled with distinct species. They are used to represent the evolutionary history for the species. This paper is concerned with a metric called *subtree transfer* (STT) distance [5, 6] for measuring the distance between two phylogenies. The STT distance measures the minimum total

*A preliminary version appeared in part in Proceedings of the 11th International Conference on Algorithms and Computation, pages 527–538, 2000.

†Email: `{wkhon,twlam,smyiu}@cs.hku.hk`. Research supported in part by the Hong Kong RGC Grant HKU-7042/02E.

‡Email: `kao@cs.northwestern.edu`. Research supported in part by NSF Grant EIA-0112934.

§Email: `ksung@comp.nus.edu.sg`

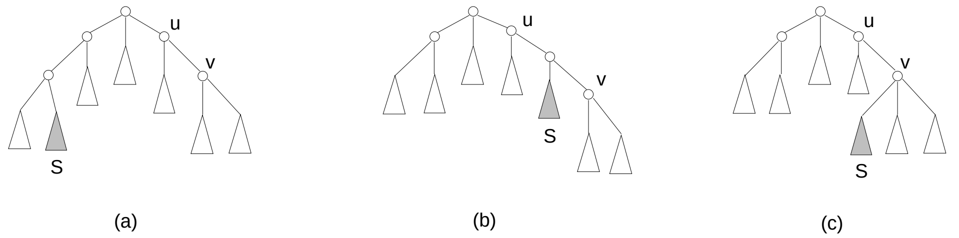

Figure 1: Examples of restricted-STT and STT operations. Tree (a) is transformed to tree (b) by a restricted-STT operation, while tree(a) is transformed to tree (c) by an STT operation.

cost of STT operations to transform one phylogeny to the other, where an STT operation detaches a subtree from a node and re-attaches it to another part of the tree.

The true phylogeny for a set of species is normally unique, since during the evolution process, the parent of a species is usually unique. However, there can be exceptions, and the study of STT distance is motivated by the interest in *recombination events* that create such exceptions. A recombination event often occurs among closely related species, such as a specific class of viruses, so that a species is derived from two parent species. That is, the genetic material of the child species is inherited from both parent species. Consequently, the evolutionary history can induce more than one true phylogeny. Hein [5] has demonstrated that when there is a single recombination event, two phylogenies are derived and they differ by one STT operation. The STT distance can therefore be used as a measurement on how likely two distinct phylogenies are both true, with the discrepancy due to recombination events.

Previous work on the subtree transfer (STT) distance focuses on degree-3 trees only [7, 1]. In particular, in the course of transforming a degree-3 tree to another degree-3 tree, all intermediate trees are required to be of degree 3. In other words, the STT operation is *restricted* in the sense that the subtree detached can only be re-attached to the middle of an edge, producing a new internal node with degree 3. See Figure 1 for an example. In this paper we study the STT distance for degree-d phylogenies for any $d \geq 3$ while allowing an STT operation to re-attach the subtree to either an internal node or the middle of an edge.[a]

An STT operation is charged by how far a subtree is transferred. More specifically, depending on whether the trees are unweighted or weighted, we count respectively the number of the edges or the total weight of the edges between the nodes where detachment and re-attachment take place.[b] Define the STT (respec-

[a]The generalized STT operation allows us to model the general recombination event such that a parent species involved in the recombination can be a common ancestor of two or more species.

[b]This cost model is referred to as the linear cost model in the literature [2, 1]. Intuitively, a smaller cost indicates a closer relationship between the parent species of the corresponding recombination. Note that recombination is more likely to occur for closer parent species. In contrast, the unit cost model gives a cost 1 for each STT operation. Thus, the linear cost model is

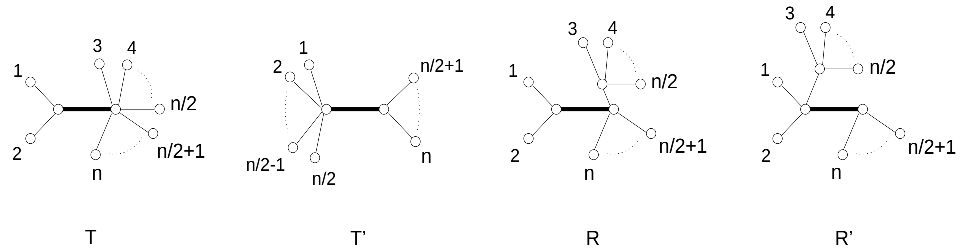

Figure 2: The unweighted STT distance between T and T' is $\Omega(n)$. Consider T and T' as weighted trees such that every internal edge has a unit weight (i.e., the highlighted edges in the figure). The weighted STT distance between T and T' is 1. In particular, the cost of transforming T to R, then to R', and finally to T' is $0 + 1 + 0 = 1$.

tively, restricted-STT) distance between two phylogenies as the minimum cost of transforming one to the other using STT (respectively, restricted-STT) operations. Unlike many other graph or tree problems, the unweighted version of the STT distance problem is not a special case of the weighted version. In particular, Figure 2 shows two phylogenies whose unweighted STT distance is $\Omega(n)$, yet if we assign a unit weight to every edge of these phylogenies, their weighted STT distance is only $O(1)$. On the other hand, the unweighted STT distance is not necessarily bigger than the weighted one; Figure 3 shows two phylogenies whose unweighted STT distance is indeed smaller than the weighted one.

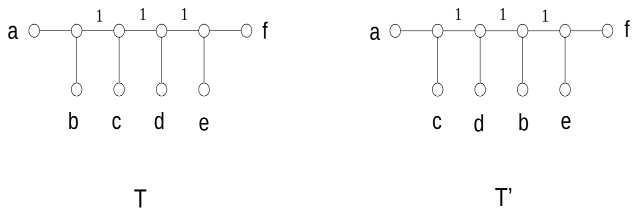

Figure 3: T and T' are degree-3 phylogenies. The unweighted STT distance between T and T' (which is 3) is smaller than the weighted STT distance (which is 4).

Consider degree-3 phylogenies. In the weighted case, we can prove that the STT distance is the same as the restricted-STT distance, and DasGupta et al. have shown that the latter can be approximated within a factor of 2 [1]. In the unweighted case, deriving a tight approximation algorithm is more difficult; the restricted-STT distance can be approximated within only a factor of $O(\log n)$ [1, 8]. This result implies an approximation algorithm for the STT distance with the same performance. However, there are examples in which the STT distance is much smaller than the restricted-STT distance. It is natural to ask whether the STT distance can be approximated within a better factor.

preferable to the unit cost model as the latter does not reflect the likelihood of the recombination events.

Consider degree-d phylogenies. First of all, it is worth mentioning that the restricted-STT distance is ∞ as a restricted-STT operation can only produce an internal node of degree 3. In the weighted case, the STT distance can be approximated by adapting the algorithm by DasGupta for degree-3 trees [1], achieving the approximation factor of 2. In the unweighted case, we give an algorithm to approximate the STT distance within a factor of $2d-4$. This result implies that for unweighted degree-3 trees, the approximation factor can be improved from $O(\log n)$ to 2 if the intermediate trees may not be necessarily degree-3 trees. The table below summarizes the approximation factors for the variants of the STT distance.

	restricted-STT (degree-3 trees)	STT (degree-d trees)
weighted	2 [1]	2
unweighted	$O(\log n)$ [1, 8]	$2d-4$

The following summarizes the contributions of the paper.

1. We show that the problem of finding the STT distance between two weighted degree-d phylogenies is equivalent to the problem of finding the restricted-STT distance between two weighted degree-3 phylogenies. This result implies the following.
   (a) The STT distance between two weighted degree-d phylogenies can be approximated within a factor of 2.
   (b) If the leaf labels of the trees are not distinct, it is NP-hard to compute the STT distance between two trees.
2. We consider the problem of finding the STT distance between two unweighted degree-d phylogenies and obtain the following.
   (a) The STT distance between two unweighted degree-d phylogenies can be approximated within a factor of $2d-4$.
   (b) If the leaf labels of the trees are not distinct, it is NP-hard to compute the STT distance between two trees.

The rest of this paper is organized as follows. Section 2 contains notations and definitions used in this paper. The result for the weighted case is given in Section 3. For the unweighted case, Section 4 gives an approximation algorithm for computing the STT distance between two unweighted phylogenies while Section 5 shows that computing STT distance between two unweighted phylogenies with possibly non-distinct labels is NP-hard.

2. Preliminaries

We first recall the definitions of STT operation, restricted-STT operation, STT distance and restricted-STT distance. Given a tree T (rooted or unrooted), a subtree transfer (STT) operation is defined as follows. We select a subtree S from T.

Suppose S is attached to a node u by an edge e. Pick another edge $e' = (v, w)$ (or an internal node t) not in S. Detach e and S and re-attach them to a newly created node x in e' (or t). If u becomes degree 2 after removing S, merge the two edges connected to u into one. In the weighted version, let $w(e)$ denote the weight of an edge e; we require that if a new node x is created, then $w((v, x)) + w((x, w)) = w(e')$; furthermore, if u is removed, the weight of the merged edge is the total weight of the two merging edges. In addition, we assume the weights of all the remaining edges do not change.

An STT operation is called *restricted* if S is always re-attached to a new node inside an edge. An STT operation is charged by how far the subtree is transferred. Precisely, the cost of an STT operation is defined as the number of edges or the total weight of edges, for unweighted and weighted version, respectively, on the shortest path from u to x (or t).

The STT distance between two trees T_1 and T_2, denoted by $\texttt{STTdist}(T_1, T_2)$, is defined as the minimum cost of transforming T_1 to a tree which is leaf-label preserving isomorphic to T_2 using STT operations. The restricted-STT distance, denoted by $\texttt{rSTTdist}(T_1, T_2)$, is defined similarly by allowing only restricted-STT operations.

Note that $\texttt{STTdist}(T_1, T_2) = \texttt{STTdist}(T_2, T_1)$. However, the corresponding equality may not hold for restricted-STT distance. For example, consider the case that T_1 is a degree-4 tree while T_2 is a degree-3 tree. It is possible to transform T_1 to T_2 using only restricted-STT operations, but it is not possible vice versa.

Finally, we define another concept called *split*. Consider a tree T and an edge e in T. Note that the removal of the edge e from T would induce a bipartition of the leaf labels. Such a bipartition is called the split of e.

3. The STT distance between two weighted degree-d phylogenies

This section shows that the problem of computing the STT distance between two weighted degree-d phylogenies (the weighted *STT-d* problem) is equivalent to the problem of computing the restricted-STT distance between two weighted degree-3 phylogenies (the weighted *rSTT*-3 problem). Since DasGupta *et al.* [1] have shown that the weighted rSTT-3 problem can be approximated within a factor of 2 and it is NP-hard when the set of leaf labels is not distinct, the same results apply to the weighted STT-d problem.

We first show that the STT-d problem can be reduced to the rSTT-3 problem. Given a weighted degree-d phylogeny X, we construct a degree-3 phylogeny T from X as follows.

Transformation from a degree-d phylogeny to a degree-3 phylogeny: For each node u of X with degree $d > 3$, let the edges that are attached to u be $e_0, e_1, \ldots, e_{d-1}$. Pick one of the edges, say $e_0 = (u, v)$. Create $d - 3$ new nodes $y_1, y_2, \ldots, y_{d-3}$ on e_0 such that y_1 is adjacent to u, y_i is adjacent to y_{i-1} for $2 \leq i \leq d-3$ and $w((y_1, u)) = 0$, $w((y_i, y_{i-1})) = 0$ for $2 \leq i \leq d-3$, $w((v, y_{d-3})) = w(e_0)$. Detach e_i and the corresponding subtree, and reattach them to node y_i for

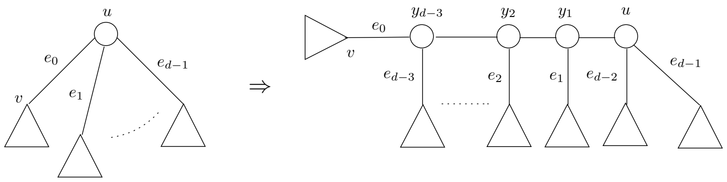

Figure 4: Degree-d node to degree-3 node.

$1 \leq i \leq d-3$. See figure 4 for an example.

Let T be the resulting tree after the transformation. Note that T is a degree-3 phylogeny and the transformation only uses restricted-STT operations of zero cost (to generate the new edges of weight zero). We call T or any tree that can be generated from X with the above process a degree-3 representation of X. Based on the above construction, we have the following facts.

Fact 1 *Let X be a weighted degree-d phylogeny. Let T and T' be two degree-3 representations of X. Then,*

- $\texttt{rSTTdist}(X, T) = 0$ *and* $\texttt{STTdist}(T, X) = 0$*, and*
- $\texttt{rSTTdist}(T, T') = 0$.

Also, STT operations on X can be 'simulated' by a sequence of restricted-STT operations on its degree-3 representation T with the same cost. More precisely, we have the following lemma.

Lemma 1 *Let X be a weighted degree-d phylogeny and X' be constructed from X by one STT operation with cost c. Let T and T' be the degree-3 representation of X and X', respectively. Then, T can be transformed to T' using a number of restricted-STT operations with the total cost c.*

Proof. Firstly, for each edge e of X, there is a unique edge e' of T such that their splits are the same. Also, for each degree-3 node u in X, there are three edges attached to them, say, e_1, e_2, and e_3. Then, there is a unique node u' in T such that the three edges attached to u' would have the same split as e_1, e_2 and e_3, respectively.

Now, we can simulate an STT operation on X, which is of cost c, by a restricted-STT operation on T as follows. Suppose that S is the subtree being moved by the STT operation on X, and S is attached to some node u by an edge e_1.

Case 1. If S and e_1 is moved and re-attached to the middle of an edge e_2, we simulate the operation in T by moving the edge e'_1 and its attached subtree to become re-attached to the middle of the edge e'_2.

Case 2. If S and e_1 is moved and re-attached to a degree-3 node v, let e' be any edge attached to v' in T. We can simulate the operation in T by moving e'_1 and its attached subtree to become re-attached to the middle of e', forming a new node x on e' with $w((x, v')) = 0$.

Case 3. Otherwise, S and e_1 must be moved and re-attached to some v of degree $d > 3$. Suppose that v is attached to the edges $f_1, f_2, \ldots, f_d$. Then, there must be some node in T which is attached to two edges f'_i and f'_j, for some $1 \leq i < j \leq d$. Label the node as v'. Then, for this node v', we label the third attaching edge as $\overline{e}$. Intuitively, $\overline{e}$ is the new edge added when we transform v from degree d to degree 3. Then, we move e'_1 and its subtree to become re-attached to $\overline{e}$.

Note that the above three cases summarize all possible STT operations on X, and it can be shown that T is transformed to a tree T' which is a degree-3 representation of X'. Moreover, the transformation cost is c. □

Now, we show that $\texttt{STTdist}(X_1, X_2) = \texttt{rSTTdist}(T_1, T_2)$.

Lemma 2 *Let X_1 and X_2 be two weighted degree-d phylogenies, where $d \geq 3$. Let T_1 and T_2 be degree-3 representations of X_1 and X_2, respectively. Then, we have $\texttt{STTdist}(X_1, X_2) = \texttt{rSTTdist}(T_1, T_2)$.*

Proof. To transform X_1 into X_2 using STT operations, we can first transform X_1 to T_1, then transform T_1 to T_2 using restricted-STT operations, and finally transform T_2 back to X_2. In other words, $\texttt{STTdist}(X_1, X_2) \leq \texttt{rSTTdist}(X_1, T_1) + \texttt{rSTTdist}(T_1, T_2) + \texttt{STTdist}(T_2, X_2)$. By Fact 1, we have $\texttt{rSTTdist}(X_1, T_1) = 0$ and $\texttt{STTdist}(T_2, X_2) = 0$. So, we have $\texttt{STTdist}(X_1, X_2) \leq \texttt{rSTTdist}(T_1, T_2)$.

On the other hand, by Lemma 1, to transform T_1 into T_2 using restricted-STT operations, we can simulate the transformation from X_1 to X_2 on T_1 to obtain T' where T' is a degree-3 representation of X_2. Then, we transform T' to T_2. By Fact 1, $\texttt{rSTTdist}(T', T_2) = 0$. So, $\texttt{rSTTdist}(T_1, T_2) \leq \texttt{STTdist}(X_1, X_2)$. The lemma follows. □

By Lemma 2, we have the following theorem.

Theorem 2 *The weighted STT-d problem is equivalent to the weighted rSTT-3 problem.*

Proof. Consider two weighted degree-d trees X_1 and X_2. Let R_1 and R_2 be the degree-3 representations of X_1 and X_2, respectively. By Lemma 2, we have $\texttt{STTdist}(X_1, X_2) = \texttt{rSTTdist}(R_1, R_2)$. Thus, the weighted STT-d problem can be reduced to the weighted rSTT-3 problem.

Given two weighted degree-3 trees T_1 and T_2, T_1 and T_2 are their own degree-3 representations. By Lemma 2, $\texttt{rSTTdist}(T_1, T_2) = \texttt{STTdist}(T_1, T_2)$. Hence, the weighted rSTT-3 problem can be reduced to the weighted STT-d problem. □

Together with the results in DasGupta et al [1], we get the following corollary.

Corollary 1 *The problem of computing the weighted STT-d distance can be approximated within a factor of 2 and the problem is NP-hard when the set of leaf labels is not distinct.*

4. An approximation algorithm for unweighted STT distance

This section gives an approximation algorithm for computing the STT distance between two unweighted degree-d phylogenies. The approximation factor is $2d - 4$, which is independent of the number of leaves, n.

Given two phylogenies, we first define what a *non-leaf-label-shared* edge is, and give a lower bound on the STT distance between the phylogenies based on the number of non-leaf-label-shared edges in the phylogenies.

Let T and T' be two phylogenies with the same set of leaf labels. An edge e in T is said to be *leaf-label-shared* (w.r.t. T'), if for some edge e' in T', e and e' have the same split; otherwise, e is said to be *non-leaf-label-shared*.

Lemma 3 *Let T and T' be two degree-d phylogenies with the same set of leaf labels. Let b and b' denote the number of non-leaf-label-shared edges in T and T', respectively. Then* $\texttt{STTdist}(T, T') \geq \max(b, b')$.

Proof. By viewing an edge as a partition of leaves, a sequence of STT operations with cost d can create at most d new edges and delete at most d edges. To transform T to T', we must either delete or create at least $max(b, b')$ edges, because any non-leaf-label-shared edge of one tree is not contained in another. Thus, the STT distance is at least $max(b, b')$. □

Note that STT operations are reversible in the sense that if a tree T_1 can be transformed to T_2 using a sequence σ of STT operations, we can easily transform T_2 to T_1 by reversing the operations in σ with the same cost. Based on the following lemma and the reversibility of STT operations, we will derive a linear time approximation algorithm for computing the STT distance between two phylogenies.

Lemma 4 (Theorem 4, [9]) *Let T and T' be two unweighted degree-d phylogenies with the same set of leaf labels. If both of them do not contain non-leaf-label-shared edges, T and T' are isomorphic.*

Figure 5 details the approximation algorithm. The basic idea is to transform each phylogeny to one without non-leaf-label-shared edges using STT operations.

1. Identify non-leaf-label-shared edges in T and T'.
2. Transform T to T_s by "contracting" all non-leaf-label-shared edges using a sequence σ_1 of STT operations as follows:
   2.1 Partition the set of non-leaf-label-shared edges into groups such that if two edges are connected in T, they are in the same group.
   2.2 For each group, pick an internal node x which attaches to one of the edges. For each non-leaf-label-shared edge (x, y), let d be the degree of y. By STT operations, we detach $(d-2)$ subtrees from y and re-attach them to x. Then, y becomes degree-2 and disappears. Repeat until all non-leaf-label-shared edges in the group are removed.
3. Repeat Step 2 to transform T' to T'_s using a sequence σ_2 of STT operations.
4. Output the total cost c of all STT operations in σ_1 and σ_2.

Figure 5: Approximation Algorithm for computing STT distance.

The following lemma analyses the approximation factor and the time complexity of the algorithm.

Lemma 5 *Let T and T' be two degree-d phylogenies with the same set of leaf labels. Then,* $\texttt{STTdist}(T, T')$ *can be approximated within a factor of $2d - 4$ in $O(n)$ time.*

Proof. We apply the approximation algorithm presented in Figure 5 to T and T'. Since the leaf labels are distinct in T (T'), Step 1 can be done in $O(n)$ time [3]. All other steps can be completed in $O(n)$ time, so the overall time complexity is $O(n)$.

By Lemma 4, T_s and T'_s are isomorphic. Thus, by using the sequence of STT operations in σ_1 and reversing the operations in σ_2, we can transform T to T' with cost c.

To determine the approximation factor, it remains to bound the value of c. Note that all the STT operations are performed in Step 2, where we remove (contract) each the non-leaf-label-shared-edges. The cost of STT operations for each removal is at most $d-2$. Thus, $c \leq b(d-2) + b'(d-2)$ where b and b' are the number of non-leaf-label-shared-edges in T and T', respectively. By Lemma 3, we have $c \leq (2d-4) \cdot \max(b, b') \leq (2d-4)\texttt{STTdist}(T, T')$, so the result follows. □

5. Unweighted degree-d phylogenies and NP-hardness

This section studies the computational complexity for computing the STT distance between two unweighted degree-d phylogenies. We prove the NP-hardness of a slightly more general problem. We prove that the problem of computing the STT distance between two unweighted trees with leaves labeled by possibly non-distinct labels is NP-hard. Our result also implies the NP-hardness of the weighted version of this problem, which was proven in [1, 8].

We consider the following decision problem. Given two degree-d unrooted trees T and T' and an integer t, the problem is to determine whether $\texttt{STTdist}(T, T') \leq t$. We show that this decision problem is NP-hard by reducing the Exact Cover by 3-Sets (X3C) problem [4] to it. The X3C problem is defined as follows. Given a set $\Gamma = \{s_1, s_2, \ldots, s_{3q}\}$ for some integer q and a collection $\mathcal{C} = C_1, C_2, \ldots, C_n$ of subsets of Γ where $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$ and $\cup_{i=1}^{n} C_i = \Gamma$, determine whether $\mathcal{C}$ contains an exact cover for Γ, that is, a sub-collection $\mathcal{D} = D_1, D_2, \ldots, D_q$ of $\mathcal{C}$ such that $\cup_{j=1}^{q} D_j = \Gamma$. Given an instance of X3C problem, we construct two degree-d trees T and T' where $d = 4n - q$ as follows.

Construction of T: For each $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$, we construct a subtree with three leaves labeled as $s_{i_1}, s_{i_2}, s_{i_3}$, respectively (see Figure 6(a)). In T, each of these subtrees is attached to a long arm where a long arm is made up of three short arms. Each short arm is a path with n^2 leaves labeled as $x_0, x_1, \ldots, x_{n^2-1}$ hanging on the path (see Figure 6(b)). All long arms meet at a single node M (see Figure 6(c)). Note that the set of leaf labels in T is $\{s_1, s_2, \ldots, s_{3q}, x_0, x_1, \ldots, x_{n^2-1}\}$.
Construction of T': We construct $3q$ leaves with labels $s_1, \ldots, s_{3q}$, respectively. Each of these leaves is attached to a short arm as shown in Figure 6(d). The short arms meet at a node N. Besides the short arms, there are $n-q$ long arms attached to N. To make T' have the same multi-set of leaf labels as T, we create additional leaves with appropriate labels and attach these leaves to N directly.

Let $t = 3n^3 + 6n$. We show that $\texttt{STTdist}(T, T') \leq t$ if and only if Γ has an exact cover. If Γ has an exact cover, there exists q long arms in T such that the set of labels s_i of all leaves at the end of these long arms is equal to Γ. To transform T to

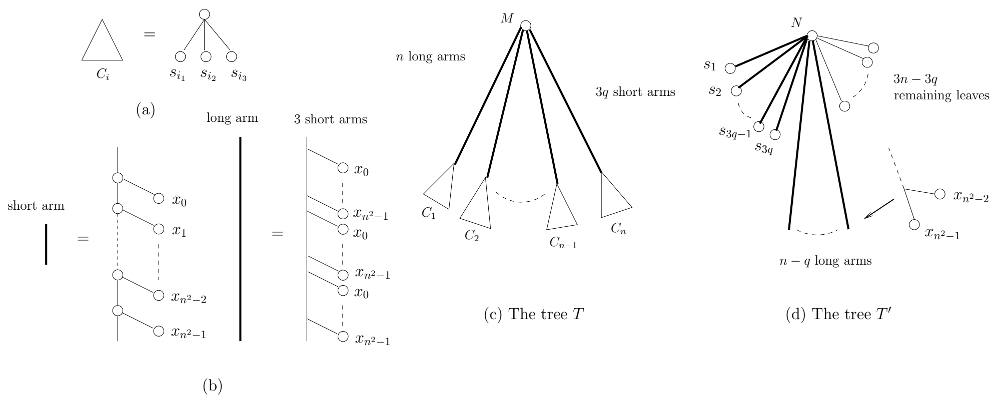

Figure 6: Trees T and T'.

T', basically, each of these long arms is transformed into 3 short arms where every short arm contains a leaf with label s_i at its end. For the other long arms of T, we move up all leaves with labels s_i to M. The whole procedure can be done using STT operations of cost at most t. The detail steps are given below (c.f. [1, 8]). Although T is an unrooted tree, for the ease of description, if a STT operation moves a subtree towards M, we say that it *moves the subtree up* the tree, otherwise, we say that it *moves the subtree down* the tree.

Without loss of generality, let $\{C_1, C_2, \ldots, C_q\}$ be an exact cover of Γ. There are two cases.

Case 1. For the long arm corresponding to C_i where $1 \leq i \leq q$, let $C_i = \{s_p, s_q, s_r\}$. We first move the two leaves with labels s_p and s_q up n^2+1 edges (see Figure 7(a)) using STT operations of cost n^2+2. Note that the subtree R (see the figure) will be a short arm in T'. The next step is to move the leave with label s_q and R up n^2+1 edges (see Figure 7(b)) using STT operations of cost n^2+2. Note that the subtree P will be another short arm in T'. The last step is to move both P and R to M and make them attach to M directly. This requires STT operations of cost n^2+2. The total cost for each long arm is $3n^2+6$. So, the total cost of to transform these long arms into short arms is $3n^2q+6q$.

Case 2. For each of the remaining long arms, move the subtree containing the three leaves labeled by Γ up to M using STT operations of cost $3n^2$, then using two more STT operations to make each leaf labeled by Γ attached to M directly. The total cost of STT operations for these long arms is $(3n^2+2)(n-q)$.

Thus, we can transform T to T' using STT operations with total cost at most $3n^3+6n$. Thus, the correctness of the if-part is established. In the following, we focus on showing the correctness of the only-if part. That is, we show that if $\texttt{STTdist}(T, T') \leq 3n^3+6n$, then there is an exact cover of Γ.

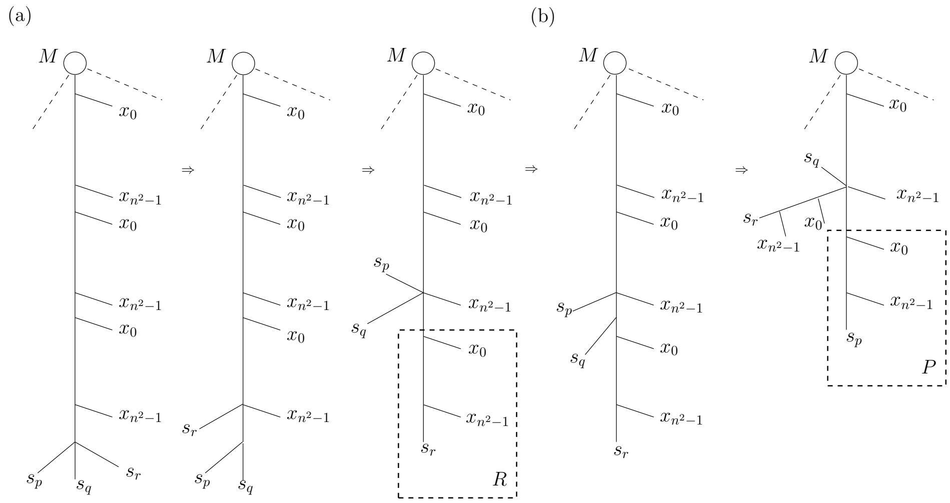

Figure 7: Transforming a long arm in T to three short arms in T'.

Observation 3 *For any STT operation of cost k, we can decompose it into k STT operations where each of them has cost one.*

From this point onwards, we assume that each STT operation has unit cost unless otherwise stated. In other words, each STT operation will move a subtree together with the edge attached to it from one end of an edge to the middle or the other end of the same edge. To prove the only-if part, for a given sequence of STT operations that transform T to T', we identify a set of *effective* STT operations (defined in Section 5.1. We show that each of these operations can be characterized by a unique edge (called an *upward* edge) in T. If the total number of STT operations is small ($\leq 3n^3 + 6n$), then the number of effective operations must be large, that is, there must be a lot of upward edges and this implies the existence of an exact cover for Γ. We first give some preliminary definitions and concepts in Section 5.1. We then give a lower bound on the number of upward edges in Section 5.2. The proof of the only-if part is given in Section 5.3.

5.1. Definitions and concepts

Let F be a given sequence of STT operations of total cost at most $3n^3 + 6n$ that transforms T to T'. By tracing the operations, there is a one-to-one correspondence of the leaves in T to those in T'. We can relabel the leaves of T by giving an extra index to the leaves with same labels. For example, we can use $x_{0,1}, x_{0,2}, \ldots$ to distinguish leaves with label x_0. The leaves in T' can be relabeled according to the new labels of the corresponding leaves in T. After the relabeling, we can regard leaves in T (or T') having distinct labels. Since the index j in $x_{i,j}$ is not important, so we will refer to any particular $x_{i,j}$ simply by x_i in the following. Similarly for leaves with labels s_i's, we relabel them and refer to them using the same approach. In other words, we can now assume that all leaves are relabeled with distinct labels.

Recall that the split of each edge e in T induces a bipartition of leaves, denoted as b_e. Since T and T' have $3n^3+n$ and $3n^3-n+q$ internal edges respectively, let $\mathcal{B} = \{b_1, b_2, \ldots, b_{3n^3+n}\}$ and $\mathcal{B}' = \{b'_1, b'_2, \ldots, b'_{3n^3-n+q}\}$ be the the set of bipartitions induced by the splits of the internal edges in T and T' respectively. Observe that $\mathcal{B} \cap \mathcal{B}' = \emptyset$.

Let T_i be the resulting tree after performing the ith STT operation in F. Let $\mathcal{B}^i$ be the set of bipartitions induced by the splits of the internal edges in T_i. For any i, since the leaves are uniquely relabeled, the bipartitions in $\mathcal{B}^i$ are unique. Let *op* be an STT operation in F that transforms T_i to T_{i+1}. After the operation, we either (1) delete a bipartition $b \in \mathcal{B}^i$ and create a new bipartition $b' \in \mathcal{B}^{i+1}$; or (2) create a new bipartition without deleting any bipartition in $\mathcal{B}^i$; or (3) delete a bipartition in $\mathcal{B}^i$; or (4) do not change the set of bipartitions. See Figure 8 for all these STT operations.

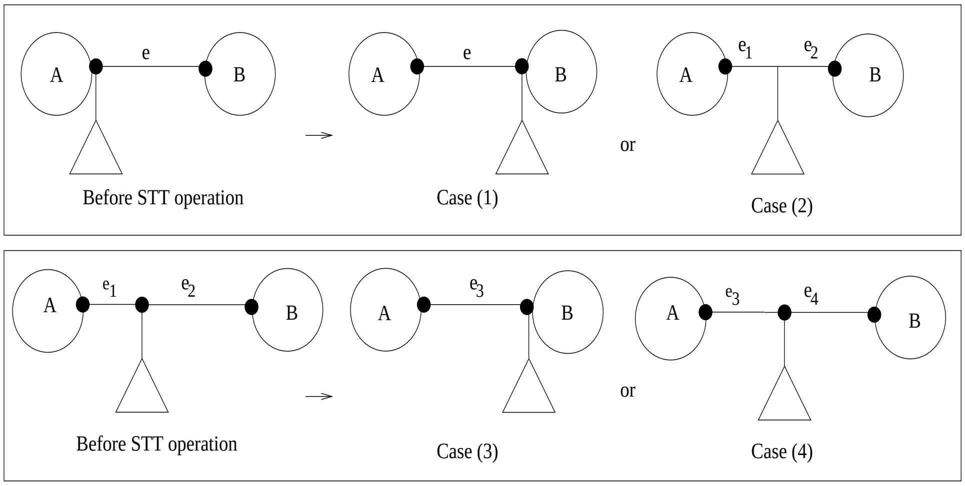

Figure 8: Possible STT operations.

An STT operation *op* is called an *effective operation* if *op* deletes a bipartition $b \in \mathcal{B}^i$ and creates a new bipartition $b' \in \mathcal{B}^{i+1}$ (Case 1) such that $b \in \mathcal{B}$, $b' \in \mathcal{B}'$ and no subsequent STT operations in F delete b'. The unique edge e in T which induces b is called an *effective edge* and we say that e maps $\tilde{e}$ if $\tilde{e}$ induces b' in T'. Finally, an STT operation which is not effective is called an *ineffective operation*.

Note that each effective operation *op* can be characterized by the pair of edges $(e, \tilde{e})$. The following lemma gives the bounds for the numbers of effective and ineffective operations.

Lemma 6 *If* $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$, *then* (i) *number of effective operations* $\geq 3n^3 - 6n$ *and* (ii) *number of ineffective operations* $\leq 12n$.

Proof. Since $\mathcal{B} \cap \mathcal{B}' = \emptyset$, there must be STT operations that delete all bipartitions in $\mathcal{B}$ and create the bipartitions in $\mathcal{B}'$. $|\mathcal{B}| = 3n^3+n$ and $|\mathcal{B}'| = 3n^3-n+q \geq 3n^3-n$. Since the total number of STT operations is at most $3n^3+6n$, the number of effective operations, each of which deletes a bipartition in $\mathcal{B}$ and creates a bipartition in $\mathcal{B}'$, must be at least $(3n^3+n)+(3n^3-n)-(3n^3+6n) = 3n^3-6n$. And the number of ineffective operations must be at most $(3n^3+6n)-(3n^3-6n) = 12n$. □

To help the analysis, we further classify effective edges in T. Let e be an effective edge. Let *op* be the effective operation that deletes the bipartition induced by the split of e. If *op* moves a subtree up the tree, e is called an *upward effective edge*. Otherwise, it is called a *downward effective edge*.

The number of upward effective edges is critical to the proof of the only-if part. In the next section, we will show that the number of upward effective edges will be large if the total number of STT operations is upper bounded by $3n^3+6n$.

5.2. Lower bound on the number of upward effective edges

Consider some large enough n, this section shows that if $\texttt{STTdist}(T,T') \leq 3n^3+6n$, then every long arm in T contains more than $2n^2$ upward effective edges.

We first show that the number of downward effective edges is at most $89n$. We classify the downward effective edges into the following three groups. For each downward effective edge, there is a corresponding STT operation that moves a subtree S and its attached edge e, called the *carrying edge*, from one end of some edge to the other end. The subtree S is said to be carried by the edge e. Let b_e be the bipartition induced by the split of the carrying edge e. We count the number of downward effective edges in each of the following groups. (a) e is an external edge or $b_e \in \mathcal{B}'$, (b) $b_e \in \mathcal{B}$, (c) e is not an external edge and $b_e \notin \mathcal{B} \cup \mathcal{B}'$.

Case (a). In this case, the subtree carried by e must have exactly $0, 1, 3n-1$ or $3n$ leaves labeled by Γ while the bipartition induced by the split of the downward effective edge has three leaves labeled by Γ on one side and $3n-3$ leaves labeled by Γ on the other side. By checking all cases, it is not possible to produce a bipartition in $\mathcal{B}'$ using the corresponding carrying edge. The number of downward effective edges in this group is 0.

Case (b). In this case, since b_e is in $\mathcal{B}$, let e_i be the edge in T such that its split is b_e. Let the corresponding downward effective edge be e_j and let e_j maps $\tilde{e_j}$. By checking all possible cases, we know that both b_{e_i} and b_{e_j} must contain the same set of three leaves labeled by Γ on one side of the bipartition. From the structure of T, we know that e_i and e_j must be in the same long arm A. Since e_j is a downward effective edge, e_i is closer to M (the node where other long arms meet).

Consider A and denote its edges by $e_0, e_1, \ldots, e_{3n^2-1}$ ordered from M. The set of leaves L between e_i and e_j in T contains leaves of labels $x_{i \bmod n^2}$, $x_{(i+1) \bmod n^2}$, …, $x_{(j-1) \bmod n^2}$. It can be easily verified that after the STT operation, one side of $b_{\tilde{e_j}}$ will contain leaves of these labels x_i only. In other words, if L has m leaves of label x_p, it must contain at least m leaves of label x_{p+1}, …, x_{n^2-1}. Otherwise, it is not a bipartition in $\mathcal{B}'$. Therefore, we deduce that if $j \neq n^2, 2n^2$, then $i = j \mod n^2$.

When $i = j \bmod n^2$, we know that L contains the same number of leaves with label $x_0, x_1, \ldots, x_{n^2-1}$. In T', there exists at most $3n$ bipartitions in $\mathcal{B}'$ such that one side of these bipartitions contain the same number of leaves with label $x_0, \ldots, x_{n^2-1}$. Since each effective STT operation will be corresponding to a unique edge in T', so there are at most $3n$ downward effective edges when $j \neq n^2, 2n^2$. And if we include the case when $j = n^2, 2n^2$, then the total number of downward effective edges in this group is at most $5n$.

Case (c). In this case, the bipartition induced by the split of the carrying edge e is not in $\mathcal{B}$ or $\mathcal{B}'$. First, we have the following observation.

Observation 4 *Let e_i and e_j be downward effective edges such that the splits of their corresponding carrying edges are the same. Then e_i and e_j are in the same long arm in T.*

Proof. Since the STT operation on both e_i and e_j are effective, that means once a subtree is 'moved across' any of the edge, no other subtree can move across the edge. Now, let the carrying edge of both downward effective operation is labeled as e, and assume without loss of generality that the effective operation on e_i is performed first. Then, if e_i and e_j are not on the same long arm, once the operation on e_i is performed, e and e_j are 'separated' by e_i forever, so that it is impossible to apply a downward operation on e_j afterwards.

Thus, e_i and e_j must be on the same long arm. This completes the proof. $\square$

The next lemma shows that there are at most 7 downward effective edges in the same long arm in T such that the splits of the corresponding carrying edges are the same.

Lemma 7 *For each carrying edge e, there are at most 7 downward effective edges such that their corresponding carrying edge is e.*

Proof. Fixed a long arm in T and denote its edges by $e_0, e_1, \ldots, e_{3n^2}$ from M. Let e_{i_1}, e_{i_2}, $\ldots$, e_{i_m} $(i_1 < i_2 < \ldots < i_m)$ be m downward effective edges such that their corresponding carrying edge is e.

For $k = 1, 2, \ldots, m-1$, let L_k be the set of leaves between e_{i_k} and $e_{i_{k+1}}$. Thus $L_k = \{x_{i_k \mod n^2}, \ldots, x_{(i_{k+1}-1) \mod n^2}\}$. We claim that $L_k \cup L_{k+1}$ contains at least one x_{n^2-1} leaf for all $k = 1, \ldots, m-2$. The claim can be proved as follows.

By contrary, assume $L_k \cup L_{k+1}$ does not contain x_{n^2-1}. Then, the sets L_k and L_{k+1} should be $\{x_a, \ldots, x_{b-1}\}$ and $\{x_b, \ldots, x_c\}$ respectively where $a = i_k \mod n^2$, $b = i_{k+1} \mod n^2$, $c = (i_{k+2} - 1) \mod n^2$, and $0 \leq a \leq b \leq c < n^2$.

Let e_{i_k}, $e_{i_{k+1}}$, $e_{i_{k+2}}$ map to $\tilde{e}_{i_k}$, $\tilde{e}_{i_{k+1}}$, and $\tilde{e}_{i_{k+2}}$, respectively. Let R be the partition of $b_{e_{i_k}}$ which contains more than three leaves labeled by Γ. Let R' be the set of leaves in R excluding the leaves in the subtree carried by e. Note that R' should have less than or equal to one leaf labeled by Γ. Otherwise, both partitions of $b_{\tilde{e}_{i_k}}$ in T' contains more than one leaf labeled by Γ, which is impossible. Thus, R' contains all leaves below $\tilde{e}_{i_k}$ of some arm A' in T'.

Using the same argument for $\tilde{e}_{i_{k+1}}$ and $\tilde{e}_{i_{k+2}}$, we can show that $R' \cup L_k$ and $R' \cup L_k \cup L_{k+1}$ contains all leaves below $\tilde{e}_{i_{k+1}}$ and $\tilde{e}_{i_{k+2}}$ respectively of A' in T'. Recall that $x_a \in L_k$ and $x_c \in L_{k+1}$. By construction of T', $c < a$. We arrive at contradiction and the claim follows.

Since there are only three x_{n^2-1} leaves in each arm of T, the above claim implies that at most 7 downward effective edges whose corresponding edge is e. $\square$

By Observation 4 and Lemma 7, each carrying edge corresponds to at most 7 downward effective edges in T. Each such carrying edge e requires a distinct ineffective STT operation to delete the corresponding bipartition $b_e \notin \mathcal{B} \cup \mathcal{B}'$, therefore, the number of downward effective edges in this group is at most $7 * 12n = 84n$, since by Lemma 6, there are at most $12n$ ineffective STT operations.

Summing up the number of possible downward effective edges, we have the following lemma.

Lemma 8 *The number of downward effective edges is at most* $89n$*.*

Proof. Recall the classification of downward effective edges in this subsection. The number of such edges in cases (a), (b) and (c) are at most 0, $5n$ and $84n$, respectively. Thus, the total number is at most $89n$. □

Combining Lemmas 6 and 8, we have the following.

Lemma 9 *If* $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$*, then* T *has at least* $3n^3 - 95n$ *upward effective edges.*

Proof. The number of upward effective edges

$$
\begin{aligned}
&= \text{number of effective operations} - \text{number of downward effective edges} \\
&\geq \text{number of effective operations} - 89n \qquad \text{(by Lemma 8)} \\
&\geq 3n^3 - 6n - 89n \\
&= 3n^3 - 95n.
\end{aligned}
$$

Thus, Lemma 9 follows. □

Based on Lemma 9, we have the following corollary.

Corollary 2 *For large enough* n *(*$n \geq 96$*), if* $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$*, then every long arm in* T *contains more than* $2n^2$ *upward effective edges.*

Proof. We prove the corollary by contradiction. Suppose one of the long arms contains $\leq 2n^2$ upward effective edges, then the number of upward effective edges in $T \leq (3n^2 + 1)(n - 1) + 2n^2 = 3n^3 - n^2 + n - 1$. However, since $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$, by Lemma 9, T should has at least $3n^3 - 95n$ upward effective edges. We arrive at a contradiction and the corollary follows. □

5.3. Proof of the only-if part

Now, we prove the only-if part.

Lemma 10 *Suppose that an upward effective edge* e *in a long arm* A *in* T *maps an edge* $\tilde{e}$ *in an arm* A' *in* T'*. Then, the leaves below* $\tilde{e}$ *in* A' *are originally the leaves below* e *in* A*.*

Proof. Consider a particular leaf ℓ which is below $\tilde{e}$ in A'. By contrary, suppose ℓ is originally not a leave below e in A, that is, ℓ originally appears on the bipartition of e with more than $3n - 3$ leaves labeled by Γ. Since e is an upward effective edge, this means that after $\tilde{e}$ is created, ℓ will remain on the bipartition of $\tilde{e}$ with more than $3n - 3$ leaves labeled by Γ. Thus, contradiction occurs. □

Lemma 11 *Upward effective edges from distinct long arms in* T *cannot map edges in the same arm in* T'*.*

Proof. Let e_1 and e_2 be upward effective edges from distinct long arms A_1 and A_2 in T. Suppose on contrary that the edges $\tilde{e_1}$ and $\tilde{e_2}$, which are mapped from e_1 and e_2, are in the same arm, say A', in T'. Consider the unique leaf with label x_{n^2-1} at the bottom of A' in T'. By Lemma 10, this unique leaf should be below

e_1 in A_1 in T. Similarly, we can show that the unique leaf should be below e_2 in A_2 in T. The uniqueness of the leaf implies that e_1 and e_2 are in the same arm. Thus, contradiction occurs and the lemma follows. □

Based on Corollary 2 and Lemmas 10, 11, the short arms in T' can be divided into groups such that each group has three short arms. Each group corresponds to one long arm in T where the leaves of the same group are exactly those leaves of the corresponding long arm. Thus, we have the following theorem.

Theorem 5 *If* $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$*, then* Γ *has an exact cover.*

Proof. Suppose $\texttt{STTdist}(T, T') \leq 3n^3 + 6n$. By Corollary 2, every long arm in T contains more than $2n^2$ upward effective edges. By Lemma 11, there exist at most $n - q$ long arms in T whose upward effective edges can create edges in the $n - q$ long arms in T'.

Let R be the set of remaining (at least q) long arms in T. Since each long arm in R contains more than $2n^2$ upward effective edges and each short arm in T' contains n^2 edges, the upward effective edges on each long arm in R create edges in at least three short arms in T'. As there are $3q$ short arms in T', we conclude that R contains exactly q long arms and each short arm in T' has at least one edge created from some upward effective edge in long arm in R. By Lemma 10, every leaf labeled by Γ at the bottom of the short arms of T' should appear in the long arms in R. It implies that the set of leaves labeled by Γ at the bottom of the q long arms must be equal to those at the bottom of the $3q$ short arms. In other words, there is an exact cover of Γ. This concludes the correctness of the only-if part. □

6. Conclusion

In this paper, we extend the study of the STT distance to general degree-d trees. For the weighted case, we show that computing the STT distance of two degree-d trees is equivalent to computing the restricted STT distance of two degree-3 trees. For the unweighted case, we give an algorithm that approximates the STT distance within a factor of $2d-4$. Also, we show that computing the STT distance between two non-distinctly labeled trees is NP-hard.

For future work, we would like to know whether computing the STT distance between two *distinctly* labeled trees is NP-hard or not.

References

1. B. DasGupta, X. He, T. Jiang, M. Li, and J. Tromp. On the Linear-Cost Subtree-Transfer Distance between Phylogenetic Trees. *Algorithmica*, 25(2):176–195, 1999.
2. B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang. On Distances between Phylogenetic Trees. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 427–436, 1997.
3. W. H. E. Day. Optimal Algorithms for Comparing Trees with Labeled Leaves. *Journal of Classification*, 2:7–28, 1985.
4. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

5. J. Hein. Reconstructing Evolution of Sequences subject to Recombination using Parsimony. *Mathematical Biosciences*, 98:185–200, 1990.
6. J. Hein. A Heuristic Method to Reconstruct the History of Sequences subject to Recombination. *Journal of Molecular Evolution*, 36:396–405, 1993.
7. J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, 71(1–3):153–169, 1996.
8. M. Li, J. Tromp, and L. X. Zhang. On the Nearest Neighbour Interchange Distance between Evolutionary Trees. *Journal of Theoretical Biology*, 182(4):463–467, 1996.
9. D. F. Robinson and L. R. Foulds. Comparison of Phylogenetic Trees. *Mathematical Biosciences*, 53(1–2):131–147, 1981.