DIMACS Technical Report 2007-21
December 2007

Approximation Algorithms for Constrained Generalized Tree Alignment Problem

by

Srikrishnan Divakaran
Dept. of Computer Science
Hofstra University
Hempstead, New York 11649

ABSTRACT

In generalized tree alignment problem, we are given a set S of k biologically related sequences and we are interested in a minimum cost evolutionary tree for S. In many instances of this problem partial topology of the phylogenetic tree for S is known. In such instances, we would like to make use of this knowledge to restrict the tree topologies that we consider and construct a biologically relevant minimum cost evolutionary tree. So, in this paper we propose the following natural generalization of the generalized tree alignment problem, a problem known to be MAX-SNP Hard, stated as follows:

> *Constrained Generalized Tree Alignment Problem*: Given a set S of k related sequences and a phylogenetic forest comprising of node-disjoint phylogenetic trees that specify the topological constraints that an evolutionary tree of S needs to satisfy, construct a minimum cost evolutionary tree for S.

In this paper, we present constant approximation algorithms for the constrained generalized tree alignment problem. For the generalized tree alignment problem, a special case of this problem, our algorithms provide a guaranteed error bound of $2 - 2/k$ and do not exclude any tree topology a priori.

Key words: Analysis of algorithms; Approximation algorithms; Bioinformatics; Computational Biology; Generalized Tree Alignment;

1 Introduction

Let S be a set of k biological related sequences. The ancestral relationship among these sequences are described by a phylogenetic tree. The tree is rooted and has k leaves each labeled with a unique sequence. The internal nodes of the tree correspond to the hypothetical ancestral sequence and are unlabeled. We construct an alignment by reconstructing each ancestral sequence and optimally align each pair of sequences induced by the edges of the tree. This fully labeled tree is called the evolutionary tree. In this tree, if each edge (u, v) has a cost defined to be the edit distance between the sequence labels of nodes u and v and the cost of the tree to be the sum of the cost of the edges of the tree, then *the tree alignment* is the problem of constructing a minimum cost evolutionary tree for a given phylogenetic tree. The *generalized tree alignment*, a generalization of the tree alignment, is the problem where we need to construct the minimum cost evolutionary tree among all phylogenetic trees of S. That is, we need to determine both the phylogenetic tree as well the minimum cost evolutionary tree. This problem is one of the central problems in computational biology that has been studied widely in the context of inferring evolutionary history of related species from their associated biological sequences. This problem can also be viewed as a minimum cost steiner tree problem in sequence space, where the sequence space consists of biological sequences represented by a point and only sequences that are at edit distance of one are adjacent to it.

In many instances of the generalized tree alignment problem, a partial phylogenetic tree topology for S is known either based on clustering information or based on known biological relationship between some of the sequences in S. In such instances, if there are no constraints on the evolutionary tree topology then there is no incentive for algorithms to exploit the knowledge of partial phylogenetic tree topology to construct biologically relevant evolutionary trees. However, if constraints are placed on the evolutionary tree topology, then there is incentive for algorithms to make use of the partial phylogenetic tree topology to restrict the number of tree topologies it considers and construct biologically relevant minimum cost evolutionary trees for S. So, in this paper we propose the following natural generalization of the generalized tree alignment problem:

> *Constrained Generalized Tree Alignment Problem*: Given a set of k related sequences and a phylogenetic forest comprising of node dis-joint phylogenetic trees that specifies the topological constraints that any evolutionary tree needs to satisfy, construct a minimum cost evolutionary tree for S.

Notice that the generalized tree alignment problem is a special case of constrained tree alignment problem when the phylogenetic forest has k trees each consisting of a single node labeled by one of the k sequences in S.

Previous Results: The Tree alignment problem was shown to be NP-hard by Jiang et al [8]. From the perspective of approximation algorithms, Jiang et al [8] presented a 2-

approximation algorithm for tree alignment and extended it into a polynomial-time approximation scheme. Wang and Gusfield [13] also presented a improved version of 2-approximation algorithm and the polynomial time approximation scheme. From the perspective of heuristics, Sankoff [11], Kruskal and Sankoff [10] and Altschul and Lipman [1], proposed iterative methods for tree alignment, Hein [6,7] introduced an approach for tree alignment based on the concept of sequence graph. For an excellent overview of algorithms for tree alignment and related problems, we refer the reader to Gusfield [5], and Wang and Jiang [15].

The Generalized tree alignment problem was shown to be MAX SNP-hard by Jiang et al [8] and has been studied by Kruskal and Sankoff [10], Hein [6,7], Wang and Jiang [14,15], and Schwikowski and Vingron [12]. From the perspective of approximation algorithms, Gusfield [4] presented a 2 approximation algorithm based on minimum spanning tree heuristic, Jiang et al [8] suggested a steiner tree heuristic based on the steiner tree approximation algorithms of Du et al [3], Zelikovsky [18] and Berman and Ramaiyer [2] that lead to better approximation ratios. However, Schwikowski and Vingron [12] point out in their paper that these heuristics exclude many topologies apriori and as a result do not provide biologically reasonable results. In addition the computational complexity of these heuristics limit their usefulness. Later, Schwikowksi and Vingron [12] proposed a deferred path heuristic that uses sequence graphs to merge a path heuristic for the construction of steiner tree with clustering method as usually applied to only distance data. Their heuristic provides a guaranteed error bound of $(2-2/k)$ and their experimental results indicate that they achieve results that are biologically meaningful and are of good quality when measured in terms of tree length, their underlying scoring function.

Our Results: In this paper, we present polynomial time constant approximation algorithms for the constrained generalized tree alignment problem. First, we present an $O(k^3 + k^2n^2)$ time 4-approximation algorithm. Then, for $0 < \epsilon \leq 1$, we present a $2(1+\epsilon)$ approximation algorithm with a run time proportional to the number of spanning trees of the edit graph of S with cost less that $\frac{2}{1+\epsilon}$ times the cost of an optimal lifted tree of S. In this paper, our focus is on the theoretical results. In view of this and the fact that even restricted instances of the constrained generalized tree alignment problem are known to be intractable, we consider an elementary model where where the cost function is defined in terms of mutational distance. From a practical perspective our algorithms extend to nucledotide substitution and other models where the cost function satisfies triangle inequality. The details about the implementation, the experimental set-up and experimental results of our algorithms and heuristics based on these algorithms are forthcoming in a separate paper.

Paper Outline: The rest of this paper is organized as follows. In Section 2, we first describe the constrained generalized tree alignment problem and other related tree alignment problems. Then, we summarize some of the known complexity and algorithmic results in tree alignment that are relevant to this paper. In Section 3, we present properties of phylogenetic and lifted trees that satisfy some topological constraints. In Section 4 we make

use of the properties presented in Section 3 to design constant approximation algorithms for the constrained generalized tree alignment problem.

2 Problem Description

In this section we first define the tree alignment, generalized tree alignment and constrained generalized tree alignment problems. Then, we summarize known complexity and algorithmic results on the tree alignment and generalized tree alignment problems that we use in this paper.

2.1 Problem Definition

Let $S = \{s_1, ..., s_k\}$ be a set of k related biological sequences. The ancestral relationship between these sequences can be described by a phlogenetic tree. A phylogenetic tree for S is a rooted tree with k leaves where each leaf is labeled by a unique sequence in S and the internal nodes of S are ancestral sequences that are *unlabeled* (i.e. need to be determined). A *loaded tree* is a phylogenetic tree with all its internal nodes fully labeled. A *lifted tree* is a loaded tree where each internal node is labeled by one of the k sequences in S. A lifted tree is called a *uniformly lifted tree* if, for each level of the tree, either every internal node at the level receives its sequence label from its left child or every internal node at the level receives its sequence label from its right child. We typically refer to a fully labeled tree as evolutionary tree and a tree with only the leaves labeled as phylogenetic tree.

For any given rooted tree T, let $r(T)$ be the root of T, $I(T)$ be the set of internal nodes of T and $L(T)$ the set of leaves of T. For each node $v \in T$, let $l(v)$ be the sequence label associated with v, $T(v)$ denote the sub-tree of T rooted at v and $S(v)$ denote the set of sequence labels of all descendant leaves of v. For an edge $(u, v) \in T$, let $l(u, v)$ denote the label associated with (u, v).

Note: In this paper we will be considering multi-graphs where between a pair of vertices there may be multiple edges each with different edge costs. In such a situation, associating labels with edges helps to uniquely identify the multi-edges and their associated costs.

For a set of sequences S, let $\Psi_P(S)$, $\Psi_E(S)$ and $\Psi_L(S)$ denote the respective sets consisting of all phylogenetic, evolutionary and lifted trees of S. Note that $\Psi_L(S) \subset \psi_E(S)$. Also, notice that there can be many evolutionary trees that have the same underlying phlogenetic tree. We will consider all evolutionary trees that have the same underlying phylogenetic tree to belong to the same *phylogenetic class*.

For S, we define a phlogenetic forest $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ to be a set of node disjoint trees such that for $i \in [1..l]$, each $T(u_i)$ is a sub-tree of some phylogenetic tree of S and $\bigcup_1^l L(T(u_i)) = S$. For a phylogenetic forest $\mathcal{F}(\mathcal{S})$, we say $F_L(S) = \{T_L(u_1), ..., T_L(u_l)\}$

is a lifted forest of $\mathcal{F}(\mathcal{S})$ if for $i \in [1..l]$, $T_L(u_i)$ is a lifted tree of $T(u_i)$. Similarly, we say $F_E(S) = \{T_E(u_1), ..., T_E(u_l)\}$ is an evolutionary forest for $\mathcal{F}(\mathcal{S})$ if for $i \in [1..l]$, $T_E(u_i)$ is an evolutionary tree of $T(u_i)$. Notice that for a given phylogenetic forest there are several lifted forests and evolutionary forests possible depending on how sequence labels are assigned to each intermediate node in $\mathcal{F}(\mathcal{S})$.

We say that a phylogenetic tree $T \in \Psi_P(S)$ *satisfies* $\mathcal{F}(\mathcal{S})$ if every tree in $\mathcal{F}(\mathcal{S})$ is a sub-tree of T. We say that an evolutionary tree $T \in \Psi_E(S)$ *satisfies* $\mathcal{F}(\mathcal{S})$ if the phylogenetic tree corresponding to T satisfies $\mathcal{F}(\mathcal{S})$. For a given phylogenetic forest $\mathcal{F}(\mathcal{S})$, let $\Psi_P^F(S)$, $\Psi_L^F(S)$ and $\Psi_E^F(S)$ be the respective phylogenetic, lifted and evolutionary trees that satisfy $\mathcal{F}(\mathcal{S})$. Note that if $F = \{s_1, ..., s_k\}$, then $\Psi_P^F(S) = \Psi_P(S)$, $\Psi_E^F(S) = \Psi_E(S)$ and $\Psi_L^F(S) = \Psi_L(S)$. Otherwise, $\Psi_P^F(S) \subseteq \Psi_P(S)$, $\Psi_E^F(S) \subseteq \Psi_E(S)$ and $\Psi_L^F(S) \subseteq \Psi_L(S)$.

For any given evolutionary tree T, let $Cost(T)$ be defined as the sum of the cost of the edges of T, where the cost of an edge $(u, v) \in T$ is the *edit distance* $dist(u, v)$ between the sequence labels of nodes u and v. The edit distance is is defined to be the minimum number of substitutions/insertions required to change u to v or vice versa and can be easily determined using dynamic programming in time proportional to the product of the lengths of the input sequences.

We will denote by $OPT_E(S)(OPT_L(S))$ the evolutionary tree(lifted tree) in $\Psi_E(S)(\Psi_L(S)$ with minimum cost. For $T \in \Psi_P(S)$, we will denote by $OPT_E(T)(OPT_L(T)$ the evolutionary tree(lifted tree) in the phylogenetic class of T with the minimum cost. For a phlogenetic forest $\mathcal{F}(\mathcal{S})$, we will denote by $OPT_E^F(S)(OPT_L^F(S)$ the evolutionary tree (lifted tree) in $\Psi_E^F(S)(\Psi_L^F(S))$ with the minimum cost.

Now, we define the tree alignment, the generalized tree alignment and constrained generalized tree alignment problems as follows:

Tree Alignment Problem: Given a set S of k related biological sequences and a phylogenetic tree $T \in \Psi_P(S)$, determine $OPT_E(T)$.

Generalized Tree Alignment Problem: Given a set S of k related biological sequences, determine $OPT_E(S)$.

Constrained Generalized Tree Alignment Problem: Given a phylogenetic forest $\mathcal{F}(\mathcal{S})$, determine $OPT_E^F(S)$.

2.2 Summary of Complexity and Algorithmic Results on Tree Alignment and Generalized Tree Alignment

In this section, we summarize the existing results on the Tree Alignment and Generalized Tree Alignment Problems that we use.

Tree Alignment

Theorem 1 *The Tree Alignment Problem is NP-Hard even when the given phylogenetic tree is a binary tree.*

This result was established by Jiang et al [8]. For the proof, we refer the reader to the paper of Jiang et al [8].

Theorem 2 *For any given phylogenetic tree $T \in \Psi_P(S)$, there exists a lifted tree for T with a cost at most twice the cost of the optimal evolutionary tree for T and can be obtained using dynamic programming in $O(k^3 + k^2n^2)$ time.*

Theorem 3 *For any $t > 0$, the Tree Alignment Problem has a PTAS with an approximation ratio $1+3/t$ and runs in time $O(k^{2^{t-1}+2}M(d, t-1, n))$ where $M(d, t-1, n)$ is the time needed to optimally align a tree of depth $t-1$.*

The above two results were first established by Jiang et al [8] and later an improved 2-approximation algorithm and PTAS was presented by Wang and Gusfield [13]. In this paper, we make use of the following results of Jiang et al [8] : (1) the dynamic programming algorithm for obtaining an optimal lifted tree for any given phylogenetic tree, and (2) PTAS for the tree alignment problem. For details about these algorithms and their analysis we refer the reader to the paper of Jiang et al [8].

Generalized Tree Alignment

Theorem 4 *Generalized Tree Alignment Problem is MAX SNP-hard.*

This result was established by Jiang et al [8]. We refer the reader to the paper of Jiang et al [8] for the proof.

Theorem 5 *Let $G(S)$ be the edit graph of S. The minimum spanning tree of $G(S)$ has a cost within $2 - 2/k$ of any evolutionary tree for S.*

This theorem was established by Gusfield [4]. We refer the reader to the paper of Gusfield [4] for the algorithm and its analysis.

3 Phylogenetic and Lifted Trees that satisfy $\mathcal{F}(\mathcal{S})$

In section 3.1, we first define an edit graph of $\mathcal{F}(\mathcal{S})$ and then characterize spanning, phylogenetic and lifted trees that satisfy $\mathcal{F}(\mathcal{S})$ in terms of spanning trees of the edit graph of $\mathcal{F}(\mathcal{S})$. In Section 3.2, we introduce ordered spanning trees and establish their relationship with phylogenetic trees that satisfy $\mathcal{F}(\mathcal{S})$. In Section 3.3, we present properties of optimal lifted trees of phylogenetic and spanning trees that satisfy $\mathcal{F}(\mathcal{S})$ and use them to establish lower and upper bounds on the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

3.1 Characterization of Phylogenetic and Lifted Trees that satisfy $\mathcal{F}(\mathcal{S})$

We first present a proposition that without loss of generality will help restrict our attention to binary phylogenetic and evolutionary trees that satisfy $\mathcal{F}(\mathcal{S})$. Then, we define the edit graph $G(F(S))$ of $\mathcal{F}(\mathcal{S})$ and characterize spanning, phylogenetic and lifted trees that satisfy $\mathcal{F}(\mathcal{S})$ in terms of the spanning trees of $G(F(S))$.

Proposition 1: Given any evolutionary tree T of S with arbitrary maximum degree there exists an evolutionary tree T' of S with maximum degree two such that $Cost(T') \leq Cost(T)$.
Proof Let T be an evolutionary tree of S with c vertices of degree greater than two. For some $d > 2$, le $v \in T$ be any vertex with d children $u_1, ..., u_d$ with sequence labels $l(u_1), ..., l(u_d)$ respectively. Now, we will replace vertex v by $d-1$ vertices of degree two and obtain an evolutionary tree T' such that $Cost(T') = Cost(T)$ and T' has $c-1$ vertices of degree greater than two. We create T' from T as follows: Let $T_1 = T(u_1)$. Create $d-1$ trees $T_2, ..., T_{d-1}$ such that for $i \in [2..d]$, the tree T_i is rooted at a node v_i with a sequence label $l(v_i) = l(v)$ and the left and right sub-trees of T_i are respectively T_{i-1} and $T(u_i)$. Now set $T' = T_{d-1}$. It is easy to observe that $Cost(T') = Cost(T)$ and T' has $c-1$ nodes of degree greater than two. Now, if we repeat this construction c times we will get an evolutionary tree with the same cost as T and the maximum degree of every vertex will be less than or equal to two. □

Edit Graph of $\mathcal{F}(\mathcal{S})$

Let $S = \{s_1, ..., s_k\}$ be a set of k related biological sequences and $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ be a phylogenetic forest for S. Let $\mathbf{F_L}$ denote the set of all lifted forests of $\mathcal{F}(\mathcal{S})$.

DEFINITION 3.1 *We define* the edit graph *of $\mathcal{F}(\mathcal{S})$ to be a graph $G(F(S)) = (V, E)$, where $V = \{u_1, u_2, ..., u_l\}$, and for each pair of sequences $s_i, s_j \in S$ there is an edge $e = (r(s_i), r(s_j))$ in E with label $l(e) = (s_i, s_j)$ and cost $cost(e) = dist(s_i, s_j)$, where $r(s_i)$ and $r(s_j)$ are the respective roots of the trees in $\mathcal{F}(\mathcal{S})$ to which s_i and s_j belong.*

Note: $G(F(S))$ is a multi-graph and between a pair of vertices there can be multiple edges each with a different cost. We label the edges in $G(F(S))$ to associate each edge with the

sequence pair in S that determines its cost.

Spanning Trees that satisfy $\mathcal{F}(\mathcal{S})$

DEFINITION 3.2 *A spanning tree T that satisfies $\mathcal{F}(\mathcal{S})$ is defined as a spanning tree on the vertices of $\mathcal{F}(\mathcal{S})$ such that each tree in $\mathcal{F}(\mathcal{S})$ is a sub-tree of T.*

A spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be constructed from $\mathcal{F}(\mathcal{S})$ by inserting a minimal set of edges from $G(F(S))$. Notice that each such minimal set is a spanning tree of $G(F(S))$. More formally, any spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be expressed as $\mathcal{F}(\mathcal{S}) \bigcup T$, for some spanning tree T of $G(F(S))$, and each edge $e = (u_i, u_j) \in T$ can be viewed as merging the phylogenetic trees $T(u_i)$ and $T(u_j)$.

Phylogenetic Trees that satisfy $\mathcal{F}(\mathcal{S})$

We can view a phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$ to be a steiner tree consisting of the nodes in $\mathcal{F}(\mathcal{S})$ and up to $l-1$ additional steiner points, whereas a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be viewed as a steiner tree consisting of the nodes in $\mathcal{F}(\mathcal{S})$ with no additional steiner points. Now, we present a property that states the relationship between spanning trees that satisfy $\mathcal{F}(\mathcal{S})$ and phylogenetic trees that satisfy $\mathcal{F}(\mathcal{S})$.

Property 1: Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ be a phylogenetic forest. A spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ and is rooted at a node in $\{u_1, ..., u_l\}$ is a phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$.

From the earlier steiner tree characterization and Property 1, we can see that the set of all spanning trees that satisfy $\mathcal{F}(\mathcal{S})$ and rooted at a node in $\{u_1, ..., u_l\}$ are a sub-set of the set of all phylogenetic trees that satisfy $\mathcal{F}(\mathcal{S})$.

Lifted Trees that satisfy $\mathcal{F}(\mathcal{S})$

A lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ can be constructed from a phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$ by labeling each internal node by one of the k sequences in S. We can also construct lifted trees that satisfy $\mathcal{F}(\mathcal{S})$ from any spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ and is rooted at a node in $\{u_1, ..., u_l\}$ by labeling each internal node by one of the k sequences in S. Notice that the set of all lifted trees that are constructed from spanning trees that satisfy $\mathcal{F}(\mathcal{S})$ are a sub-set of all lifted trees that satisfy $\mathcal{F}(\mathcal{S})$.

Any lifted tree $\bar{T}$ of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be expressed as $F_L \bigcup T$, where $F_L \in \mathbf{F_L(S)}$ is a lifted forest of $\mathcal{F}(\mathcal{S})$ and T is a spanning tree of $G(F(S))$. Let $e = (u_i, u_j)$ be an arbitrary edge that is present in $\bar{T}$ as well as $G(F(S))$. Let $l(e) = (s_{i'}, s_{j'})$ be the label associated with edge e in $G(F(S))$. Let $s_{\bar{i}}$ and $s_{\bar{j}}$ be the sequence labels of u_i and u_j in $\bar{T}$ respectively. Notice that the cost of the edge e in $G(F(S))$ may not be the same as its cost in $\bar{T}$. In $G(F(S))$, $Cost(e) = dist(s_{i'}, s_{j'})$, whereas in $\bar{T}$, $Cost(e) = dist(s_{\bar{i}}, s_{\bar{j}})$. If

$s_{i'} = s_{\bar{i}}$ and $s_j = s_{\bar{j}}$ then $Cost(e)$ in $G(F(S))$ is the same as $Cost(e)$ in $\bar{T}$. Now, we classify lifted trees of spanning trees that satisfy $\mathcal{F}(\mathcal{S})$ into *proper trees* and *improper trees* based on the relationship between the cost of the edges of T in $\bar{T}$ and the cost of the edges of T in $G(F(S))$.

DEFINITION 3.3 *Let $\bar{T} = F_L \bigcup T$ be a lifted tree of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$. We define $\bar{T}$ to be* proper *if the cost of the edges of T in $G(F(S))$ is the same as the cost of the edges of T in $\bar{T}$, otherwise* improper.

Now, we present a property about proper trees that we later use in establishing a lower bound on the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

Property 2: Let $\bar{T} = F_L \bigcup T$ be a lifted tree of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$. If $\bar{T}$ is a proper tree then $Cost(\bar{T}) = Cost(F_L) + Cost(T)$.

3.2 Ordered Trees and their relationship with phylogenetic trees that satisfy $\mathcal{F}(\mathcal{S})$

In this section, we will first introduce a new type of spanning tree that we refer to as *ordered tree*. Then, we present some properties that we primarily use to show that we can construct an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ by only considering lifted trees of spanning trees that satisfy $\mathcal{F}(\mathcal{S})$. These properties also imply that if we associate an ordered tree with each spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ and then consider only the lifted trees of these ordered trees, then we can construct an optimal lifted trees that satisfy $\mathcal{F}(\mathcal{S})$ without excluding any tree topologies apriori.

Ordered Trees

DEFINITION 3.4 *For a spanning tree T and a node $v \in T$, let an ordering $\sigma = \{\sigma(u) : u \in T(v)\}$, where $\sigma(u)$ specifies the rank among the children of u in $T(v)$. Let $\sigma(\mathbf{T})$ denote the set of all possible orderings on T. For a spanning tree T, a node $v \in T$ and an ordering $\sigma \in \sigma(\mathbf{T})$, we define $T^{\sigma}(v)$ to be* the ordered tree *of T rooted at v where the rank among the children of each node in $T(v)$ is specified by σ.*

For a spanning tree T , the number of ordered trees depends on the number of choices for the root and the number of orderings in $\sigma(\mathbf{T})$. The number of choices for the root equals k, the number of nodes in T. The number of orderings in $\sigma(\mathbf{T})$ depends on the topology of T. The worst case is when T has a star topology. In this situation the number of orderings is $(k-1)!$. However, spanning trees that correspond to biologically relevant evolutionary trees usually have good depth. In these cases the number of orderings is usually $O(2^k)$.

Relationship between Ordered Trees and Phylogenetic Trees that satisfy $\mathcal{F}(\mathcal{S})$

First, we present a function $construct - phylogenetic - tree$ that given any ordered tree

of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ transforms it into a phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$. Then, we show that $construct-phylogenetic-tree$ can construct any phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$ by transforming some ordered tree that satisfies $\mathcal{F}(\mathcal{S})$ and is rooted at a node in $\{u_1, ..., u_l\}$. This would imply that by constructing lifted trees by only considering optimal lifted trees of ordered trees we will not be excluding any tree topologies apriori.

Basic Idea: First construct a tree T_0 consisting of a single leaf node v_0. Then, recursively construct the phylogenetic tree for each of the sub-trees of $T(v)$. Finally, construct the phylogenetic tree $T_P^\sigma(v)$ by successively merging two at a time the trees T_0 and the phylogenetic trees of the sub-trees of $T(v)$. The sequence in which these phylogenetic trees are merged is specified by σ^v.

Function Construct-Phylogenetic-Tree$(T^\sigma(v))$
Inputs:(1) $T(v)$: spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ and rooted at $v \in \{u_1, ..., u_l\}$;
(2) σ : An ordering on the children of each node in $T(v)$;
Output: (1) $T_P^\sigma(v)$: A phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$;
begin
Let n_v be the number of children of v;
Let T_0 be a tree consisting of leaf node v_0;
for $(i = 1; i \leq n_v; i = i + 1)$
begin
Let $u = \sigma^v(i)$ be the i^{th} child of v ;
Let $T_P^\sigma(u) = Construct - Phylogenetic - Tree(T, u, \sigma)$ be the phylogenetic tree for $T^\sigma(u)$ constructed recursively;
Let T_i be the tree with root v_i, left sub- tree T_{i-1} and right sub-tree $T_P^\sigma(u)$;
end
return T_{n_v}
end

We now show that $construct-phylogenetic-tree$ can construct any phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$ by transforming some ordered tree of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ and is rooted at a node in $\{u_1, ..., u_l\}$.

Lemma 6 *Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ be a phylogenetic forest. For any phylogenetic tree T' that satisfies $\mathcal{F}(\mathcal{S})$ there exists an ordered tree $\bar{T}^\sigma(v)$ such that $T' = Construct - phylogenetic - tree(\bar{T}^\sigma(v))$, where $\bar{T}$ is a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$, $v \in \{u_1, ..., u_l\}$ and $\sigma \in \sigma(\bar{\mathbf{T}})$.*

Proof Let T' be any phylogenetic tree that satisfies $\mathcal{F}(\mathcal{S})$ and T'_L be some arbitrary lifted tree of T'. Now, we will construct a spanning tree $\bar{T}$ that satisfies $\mathcal{F}(\mathcal{S})$ such that $T'_L = Construct - phylogenetic - tree(\bar{T}^\sigma(v)$ for some $v \in \{u_1, ..., u_l\}$ and and some ordering $\sigma \in \sigma(\bar{\mathbf{T}})$. We can view the lifted tree T'_L as a tournament. That is at all levels the internal nodes receive its sequence label from one of its children. Let v be a node at level i and let a and b be the respective labels of its left and right child. If node v receives its sequence label

from its left child then we say that a defeated b at level i otherwise we say b defeated a at level i. Now, for each node $v \in \{u_1, ..., u_l\}$, we define $defeated(v) = \{w : w \in \{u_1, ..., u_l\}$ and $l(v)$ defeated $l(w)\}$, where $l(v)$ and $l(w)$ are the sequence labels of nodes v and w respectively. We order the nodes in $defeated(v)$ in the increasing order of the level at which they were defeated by v. Now, we define $\bar{T}$ to be $\mathcal{F}(\mathcal{S}) \bigcup \{(u_i, u_j) : u_j \in defeated(u_i)\}$, the ordering $\sigma(u_i)$ for $i \in [1..l]$ among the children of node u_i is specified by their rank in $defeated(u_i)$, and the root of $\bar{T}$ to be the node $u_r \in \{u_1, ..., u_l\}$ with an undefeated sequence label. From the construction of $\bar{T}$, we can see that $\bar{T}$ satisfies $\mathcal{F}(\mathcal{S})$ and the phylogenetic tree $Construct - phylogenetic - tree(\bar{T}^\sigma(u_r)$ has the same topology as T'_L. □

3.3 Optimal lifted trees that satisfy $\mathcal{F}(\mathcal{S})$

In this section, we first present a lemma that establishes the relation between the cost of an optimal lifted tree of an ordered tree $T^\sigma(v)$ and the cost of an optimal lifted of the phylogenetic tree $T^\sigma_P(v) = construct - phylogenetic - tree(T^\sigma(v))$. Then, we present lemmas that establish the lower and upper bounds on the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ be a phylogenetic forest consisting of binary phylogenetic trees. Let v, w be any two nodes in $\{u_1, ..., u_l\}$. Let $T(v)$ be any spanning tree that satisfies $\mathcal{F}(\mathcal{S})$, and $T^\sigma(w)$ be any ordered tree of T where $\sigma \in \sigma(\mathbf{T})$. Let $T^\sigma_P(w) = Construct - Phylogenetic - Tree(T^\sigma(w))$ be the phylogenetic tree obtained by transforming the ordered tree $T^\sigma(w)$.

Now, we show that the cost of an optimal lifted tree of $T(v)$ is the same as the cost of the optimal lifted tree of $T^\sigma_P(w)$.

Lemma 7 $Cost(OPT_L(T(v))) = Cost(OPT_L(T^\sigma_P(w))$

Proof From definition we know that the edges in $T(v)$ are the same as the edges in $T^\sigma(w)$. So any lifted tree for $T(v)$ is also a lifted tree for $T^\sigma(w)$. Now, using a function similar to $construct - phylogenetic - tree$, given any lifted tree for $T^\sigma(w)$ we can transform it into a lifted tree for T^σ_P with the same cost. This implies that $Cost(OPT_L(T(v)) \geq Cost(OPT_L(T^\sigma_P(w))$. Conversely, given any lifted tree for $T^\sigma_P(w)$, if we compress all the edges of zero cost, we will get a lifted tree for $T^\sigma(w)$ with the same cost. Hence the result. □

Now, we present three lemmas that establish lower and upper bounds on the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$, $F^{OPT}_L = \{OPT_L(T(u_1)), ..., OPT_L(T(u_l))\}$ be the lifted forest corresponding to $\mathcal{F}(\mathcal{S})$, where for $i \in [1..l]$, $OPT_L(T(u_i))$ is the optimal lifted tree of $T(u_i)$ constructed using the dynamic programming algorithm of Wang and Gusfield [13]. For $0 < \epsilon \leq 1$, let $F^\epsilon_E = \{T^\epsilon_E(u_1), ..., T^\epsilon_E(u_l)\}$ be an evolutionary forest corresponding to $\mathcal{F}(\mathcal{S})$, where for $i \in [1..l]$, $T^\epsilon_E(u_i)$ is the $(1 + \epsilon)$ approximate evolutionary tree corresponding

to $T(u_i)$ constructed using the PTAS of Wang and Gusfield [13]. Let $G(F(S))$ be the edit graph of $F(S)$ and MST be the minimum cost spanning tree of $G(F(S))$.

Lemma 8 *For any spanning tree T of $G(F(S))$ with cost $Cost(T)$, there exists a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ with cost at most $3Cost(F_L^{OPT}) + 2Cost(T)$.*

Proof Let T be some arbitrary spanning tree of $G(F(S))$ with cost $Cost(T)$, $\bar{T}_P = \mathcal{F}(\mathcal{S}) \bigcup T$ and $\bar{T}_L = F_L^{OPT} \bigcup T$. Notice that $\bar{T}_L$ is a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$. If $\bar{T}_L$ is proper then we know from property 2 that $Cost(\bar{T}_L) = Cost(F_L^{OPT}) + Cost(T)$ and we are done. So without loss of generality we assume that $\bar{T}_L$ is not proper. Now, we will establish this lemma as follows: First, we will transform T into a spanning tree T' with the degree of each node at most two and cost at most $2Cost(T)$. Then, we will transform T' into a spanning tree T_R such that the degree of each node is at most two and $F_L^{OPT} \bigcup T_R$ is a proper tree with cost at most $2Cost(F_L^{OPT}) + Cost(T') = 2Cost(F_L) + 2Cost(T)$.

Now, we will construct T'. Let $v \in \{u_1, ..., u_l\}$ be any arbitrary node in T and $u_{\pi(1)}, ..., u_{\pi(l)}$ be an ordering of the nodes of T based on a DFS of T starting at v. Let $T' = \{e = (u_{\pi(i)}, u_{\pi(i+1)}) : e \in G(F(S))$ and $Cost(e)$ is the lowest among all edges in $G(F(S))$ between $u_{\pi(i)}$ and $u_{\pi(i+1)}$. We want to remind the reader that $G(F(S))$ is a multi-graph on $\{u_1, ..., u_l\}$ that has edges between every pair of vertices and the edge costs satisfy triangle inequality. From the construction of T' and repeated application of triangle inequality, we can easily see that T' is a spanning path with cost at most $2Cost(T)$.

Now, we construct T_R from T' by replacing each edge $e' = (u_i, u_j) \in T'$ by an edge $e_r = (u_i, u_j) \in G(F(S))$ such that $Cost(e_r) = Cost(l(u_i), l(u_j))$, where $l(u_i)$ and $l(u_j)$ are the respective labels of u_i and u_j in F_L^{OPT}. From the construction of T_R, we can see that T_R is a spanning path and $F_L^{OPT} \bigcup T_R$ is a lifted tree that is proper. Now, we will show that the cost of this proper tree is at most $3Cost(F_L^{OPT}) + Cost(T')$. From property 2, we know $Cost(F_L^{OPT} \bigcup T_R) = Cost(F_L^{OPT}) + Cost(T_R)$. Now, to prove this lemma, we need to show that $Cost(T_R \leq 2Cost(F_L^{OPT} + Cost(T')$. Now, we will relate $Cost(T_R)$ in terms of $Cost(F_L^{OPT})$ and $Cost(T')$. Let $e' = (u_i, u_j)$ be some arbitrary edge in T and $e'' = (u_i, u_j)$ be its corresponding edge in T_R. Let $l(e') = (s^{i'}, s^{j'})$ and $l(e'') = (s^{i''}, s^{j''}$ be the edge labels of the edges e' and e'' respectively. Let $P(u_i, s^{i'})$ and $(P(u_j, s^{j'}))$ be the respective paths in F_L^{OPT} from node u_i to node $s^{i'}$, and node (u_j) to $s^{j'}$ respectively. Using triangle inequality, we get $Cost(e'') \leq Cost(P(u_i, s^{i'}) + Cost((s^{i'}, s^{j'}) + Cost(P(s^{j'}, u_j)$. Summing over all edges in T_R, and using the fact that the maximum degree of any node in T_R is two, we get $Cost(T_R) \leq 2Cost(F_L^{OPT}) + Cost(T')$. Hence the result. □

Lemma 9 *Let T be any spanning tree of $G(F(S))$ with cost $Cost(T)$. Then there exists an evolutionary tree that satisfies $\mathcal{F}(\mathcal{S})$ with cost at most $2(Cost(F_L^{OPT}) + Cost(T))$.*

The proof of this lemma is essentially the same as the previous lemma except that we use the PTAS of Wang and Gusfield [13] to construct an evolutionary forest F_E^{ϵ} with a stronger guarantee on its cost than the optimal lifted forest F_L^{OPT}.

Lemma 10 $Cost(OPT_L^F(S)) \geq Cost(F_L^{OPT}) + Cost(MST)$

Proof From lemmas 6 and 7, we know that to construct an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ it is sufficient to consider only lifted trees of spanning trees that satisfy $\mathcal{F}(\mathcal{S})$. We know that every spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be expressed as $\mathcal{F}(\mathcal{S}) \bigcup \mathcal{T}$ for some spanning tree T of $G(F(S))$. This implies that every lifted tree of a spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be expressed as $F_L \bigcup T$, where F_L is a lifted forest corresponding to $\mathcal{F}(\mathcal{S})$ and T is a spanning tree of $\mathcal{F}(\mathcal{S})$. We know from definition that F_L^{OPT} is the optimal lifted forest for $\mathcal{F}(\mathcal{S})$ and MST is the minimum cost spanning tree of $G(F(S))$. Therefore, any lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ will have a cost at least $Cost(F_L^{OPT}) + Cost(MST)$. □

Notice that any spanning tree that satisfies $\mathcal{F}(\mathcal{S})$ can be expressed as $\mathcal{F}(\mathcal{S}) \bigcup \mathcal{T}$, where T is a spanning tree of the edit graph $G(F(S))$. From this observation and Lemmas 6, 7, 8 and 9, we can see that an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ can be constructed by enumerating spanning trees of $G(F(S))$ with cost $\leq Cost(F_L(S)) + 2Cost(MST)$, and then for each such spanning tree T construct the optimal lifted tree for $\mathcal{F}(\mathcal{S}) \bigcup \mathcal{T}$.

4 Algorithms for constructing lifted trees that satisfy $\mathcal{F}(\mathcal{S})$

In this section we use the properties presented in Section 3 to design algorithms for constructing evolutionary trees that satisfy $\mathcal{F}(\mathcal{S})$. In Section 4.1, we present a 4-approximation algorithm. In Section 4.2, for $0 \leq \epsilon \leq 1$, we present a $2(1+\epsilon)$ approximation algorithm when the number of spanning trees of the edit graph $G(F(S))$ with cost less than $\frac{2}{1+\epsilon}(Cost(MST) + Cost(F_L^{OPT}))$ is bounded.

4.1 An Approximation Algorithm for constructing a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$

We present an algorithm A that given any phylogenetic forest $\mathcal{F}(\mathcal{S})$ constructs a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ with cost at most three times the optimal lifted tree in $O(k^2n^2 + k^3)$ time. Then, we modify this algorithm to construct a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ and is within twice the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

Basic Idea: Find the minimum spanning tree MST of the edit graph $G(F(S))$. If $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$ then return MST. Otherwise, first construct a lifted forest F_L^{OPT} by constructing the optimal lifted tree for each phylogenetic tree in $\mathcal{F}(\mathcal{S})$. Second, construct a spanning tree T' of $G(F(S))$ by joining any two vertices i and j in T such that $\pi(j) - \pi(i) = 1$, where π is any DFS order of the nodes of MST. Third, construct the tree T_R from T' by replacing each edge $(u_i, u_j) \in T'$ by the edge $(r(u_i), r(u_j))$, where $r(u_i)$ and $r(u_j)$ are the roots of the the respective trees in $\mathcal{F}(\mathcal{S})$ that contain the nodes u_i and u_j. Finally, construct T_L by concatenating F_L^{OPT} and T_R.

Algorithm A
Input: $\mathcal{F}(\mathcal{S})$: A Phylogenetic Forest $F(S) = \{T(u_1), ..., T(u_l)\}$;
Output: A lifted tree that satisfies $\mathcal{F}(\mathcal{S})$;
begin
Let $G(F(S))$ be the edit graph of $\mathcal{F}(\mathcal{S})$ and MST be a minimum cost spanning tree of $G(F(S))$;
If $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$;
Let $T_L = MST$;
else
Let $F_L^{OPT} = \{OPT_L(T(u_1)), ..., OPT_L(T(u_l))\}$, where for $i \in [1..l]$,
$OPT_L(T(u_i))$ is the optimal lifted tree of $T(u_i)$ constructed using the dynamic programming algorithm of Gusfield and Wang [13];
Let $u_{\pi(1)}, ..., u_{\pi(l)}$ be any DFS order of the vertices in MST;
Let $T' = \{e = (u_{\pi(i)}, u_{\pi(i+1)}) : e \in G(F(S))$ and $Cost(e)$ is the lowest among all the edges in $G(F(S))$ between $u_{\pi(i)}$ and $u_{\pi(i+1)}$;
Let T_R be the tree obtained by replacing each edge $(u_i, u_j) \in T'$ by an edge $(u_i, u_j) \in G(F(S))$ such that $Cost(u_i, u_j) = Cost(l(u_i), l(u_j))$, where $l(u_i)$ and $l(u_j)$ are the respective labels of u_i and u_j in F_L^{OPT};
Let $T_L = F_L^{OPT} \bigcup T_R$;
return T_L;
end

We now show that algorithm A constructs a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ and is within three times the cost of any lifted tree that satisfies $\mathcal{F}(\mathcal{S})$. From Theorem 2 this would imply that A is a 6-approximation algorithm for the constrained generalized tree alignment problem. For the special case when $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$ this problem reduces to the generalized tree alignment problem. In this case algorithm A returns the minimum cost spanning tree. From Theorem 5, this would imply that algorithm A is a 2-approximation algorithm for the generalized tree alignment problem.

Theorem 11 *Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ and let T_L be the lifted tree constructed by algorithm A then $Cost(T_L) \leq 6Cost(OPT_E^F(S))$. If $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$ then $Cost(T_L) \leq 2Cost(OPT_E^F(S))$.*

Proof If $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$ then this problem reduces to the generalized tree alignment problem. In this case this theorem follows from Theorem 5. Otherwise, this theorem follows from Theorem 2 and Lemmas 8 and 10. □

We now modify the algorithm A to make use the PTAS of Wang and Gusfield [13] to construct an evolutionary tree that satisfies $\mathcal{F}(\mathcal{S})$ and is within twice the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$. From Theorem 2 this would imply that A is a 4-approximation algorithm for the constrained generalized tree alignment problem.

We modify the algorithm A by replacing statement (1) by statement (2)

(1) $F_L^{OPT} = \{OPT_L(T(u_1)), ..., OPT_L(T(u_l))\}$, where for $i \in [1..l]$, $OPT_L(T(u_i))$ is the optimal lifted tree for $T(u_i)$ constructed using the dynamic programming algorithm of Wang and Gusfield [13].

(2) $F_L = \{T_E^{\epsilon}(u_1), ..., T_E^{\epsilon}(u_l)\}$, where for $i \in [1..l]$, $T_E^{\epsilon}(u_i)$ is the evolutionary tree for $T(u_i)$ constructed using the PTAS of Wang and Gusfield [13].

Theorem 12 *Let $\mathcal{F}(\mathcal{S})$=$\{T(u_1), ..., T(u_l)\}$ and let T_L be the lifted tree constructed by algorithm A then $Cost(T_L) \leq 4Cost(OPT_E^F(S))$. If $\mathcal{F}(\mathcal{S}) = \{s_1, ..., s_k\}$ then $Cost(T_L) \leq 2Cost(OPT_E^F(S))$.*

Proof If $\mathcal{F}(\mathcal{S}) =\{s_1, ..., s_k\}$ then this problem reduces to the generalized tree alignment problem. In this case this theorem follows from Theorem 5. Otherwise, this theorem follows from Theorem 2 and Lemmas 9 and 10. □

4.2 An Algorithm for Constructing an Optimal Lifted Tree that satisfies $\mathcal{F}(\mathcal{S})$

We now present algorithm B that given any phylogenetic forest $\mathcal{F}(\mathcal{S})$ and a real number $0 \leq \epsilon \leq 1$, constructs a lifted tree that is within $(1 + \epsilon)$ the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$.

Basic Idea: First, construct the lifted forest F_L^{OPT} by constructing the optimal lifted tree for each phylogenetic tree in $\mathcal{F}(\mathcal{S})$. Second, starting with the minimum spanning tree enumerate the spanning trees of the edit graph $G(F(S))$ in non-decreasing order of cost. For each spanning tree T of $G(F(S))$ we construct the optimal lifted tree for the phylogenetic tree $\mathcal{F}(\mathcal{S}) \bigcup T$. Finally, from among all the constructed lifted trees return the lifted tree with the lowest cost.

Algorithm B(Phylogenetic Forest $\mathcal{F}(\mathcal{S})$)
Inputs: (1) $\mathcal{F}(\mathcal{S})$: A Phylogenetic Forest $F(S) = \{T(u_1), ..., T(u_l)\}$;
(2) ϵ : The desired quality of approximation;
Output: An lifted tree that satisfies $\mathcal{F}(\mathcal{S})$;
begin
Let $G(F(S))$ be the edit graph of $\mathcal{F}(\mathcal{S})$ and MST be a minimum cost spanning tree of $G(F(S))$;
Let $F_L^{OPT} = \{OPT_L(T(u_1)), ..., OPT_L(T(u_l))\}$ be the lifted forest, where for $i \in [1..l]$ $OPT_L(T(u_i))$ is the optimal lifted tree for $T(u_i)$ constructed using the dynamic programming algorithm of Gusfield and Wang [13];
Let $c = \frac{2}{1+\epsilon}(Cost(F_L^{OPT}) + Cost(MST))$, $min = c + 1$;
Let $ST(F(S))$ be the spanning trees of $G(F(S))$ with cost less than c arranged in non-decreasing order of cost and v be any vertex in $\{u_1, ..., u_l\}$;
For $T \in ST(F(S))$

```
    begin
        Let T_P = F(S) ⋃ T;
        Let T_L^OPT(v) be the optimal lifted tree for T_P(v) constructed using the dynamic
        programming algorithm of Wang and Gusfield [13];
        If (Cost(T_L(v)) < min)
          Let T_min = T_L(v) and min = Cost(T_L^OPT(v));
    end
    return T_min;
end
```

Now, we show that algorithm B constructs a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ and has a cost that is within $(1+\epsilon)$ times the cost of any lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ in time proportional to the number of spanning trees of $G(F(S))$ with cost less than $\frac{2}{1+\epsilon}(Cost(MST) + Cost(F_L^{OPT}))$.

Theorem 13 *Let $\mathcal{F}(\mathcal{S}) = \{T(u_1), ..., T(u_l)\}$ and $0 \leq \epsilon \leq 1$ and $N(x)$ be the number of spanning trees of $G(F(S))$ with cost less than or equal to x. Let T_{min} be the lifted tree constructed by algorithm B and $c = \frac{1}{1+\epsilon}(2Cost(MST) + 3Cost(F_L^{OPT}))$, then B constructs T_{min} in $O(k^3 + k^2n^2)N(c)$ time such that $Cost(T_{min}) \leq (1+\epsilon)OPT_L^F(S)$.*

Proof From Lemmas 6 and 7, we know that to determine an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ it is sufficient to consider lifted trees of spanning trees that satisfy $\mathcal{F}(\mathcal{S})$. Now, depending on the cost of the optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ there are two situations possible. If $c > OPT_L^F(S)$, then from Lemmas 8 and 10, we know that the algorithm B will construct an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$. Otherwise, the algorithm B will construct a lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ with cost at most $3Cost(F_L^{OPT})) + 2Cost(MST)$. In this case, the cost of an optimal lifted tree that satisfies $\mathcal{F}(\mathcal{S})$ is at least $\frac{1}{1+\epsilon}(2Cost(MST)+3Cost(F_L^{OPT}))$. Hence the result. □

Remark 1: In Algorithm B if we used the PTAS of Wang and Gusfield [13] instead of their dynamic programming algorithm the worst case performance guarantee would still be the same.

5 Conclusions and Future Work

In this paper, we present polynomial time constant approximation algorithms for the constrained generalized tree alignment problem. Our focus in this paper is on the theoretical results. In view of this and the fact that even restricted instances of the constrained generalized tree alignment problem are known to be intractable, we consider an elementary model where where the cost function is defined in terms of mutational distance. From a practical perspective our algorithms extend to nucledotide substitution and other models where the cost function satisfies triangle inequality. The details about the implementation, the experimental set-up and experimental results of our algorithms and heuristics based on these algorithms are forthcoming in a separate paper.

References

[1] S. Altschul and D.Lipman, Trees, stars, and multiple sequence alignment, SIAM Journal on Applied Math **49**(1989) 197-209.

[2] P. Berman and V. Ramaiyer, Improved Approximations for the Steiner Tree Problem, Journal of Algorithms bf 17:3(1994) 381-408.

[3] D.Z. Du, Y. Zhang and Q. Zeng, On better heuristic for eucledian steiner minimum trees, Proc. of the 32nd Symposium on the Foundations of Computer Science (1991) 431-439.

[4] D. Gusfield, Efficient methods for multiple sequence alignment with guaranteed error bounds, Bull. Math. Biol. **55(1)**(1993) 141-154.

[5] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press. (1997).

[6] J. Hein, A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given, Molecular Biology and Evolution. **6**(1989) 649-668.

[7] J. Hein, Unified to Alignment and Phylogenies, Methods in Enzymology. **183**(1990) 626-645.

[8] T. Jiang, E.L. Lawler, and L. Wang, Aligning sequences via an evolutionary tree: complexity and approximation, In Symposium on Theory of Computing.(1994) 760-769.

[9] S. Kapoor and H. Ramesh, Algorithms for Enumerating All Spanning Trees of Undirected and Weighted Graphs, SIAM Journal of Computing.(1995)

[10] J.B. Kruskal and D. Sankoff, An anthology of Algorithms and Concepts for Sequence Comparison in: Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison., Addison Wesley(1983).

[11] D. Sankoff, Minimal mutation trees of sequences, SIAM Journal on Applied Math, **28**(1975) 35-42.

[12] B. Schwikowski and M. Vingron, The deferred path heuristic for the generalized tree alignment problem, In Proceedings of the first annual International Conference on Computational Molecular Biology (RECOMB'97), ACM Press. (1997) 257-266.

[13] L. Wang and D. Gusfield, Improved Approximation Algorithms for Tree Alignment, Journal of Algorithms. **25**(1997) 255-273.

[14] L. Wang and T. Jiang, On the Complexity of Multiple Sequence Alignment, Journal of Computational Biology. **1**(1994) 337-348.

[15] L. Wang and T. Jiang, Algorithmic Methods for Multiple Sequence Alignment, Current Topics in Computational Molecular Biology, MIT Press. (2002) 72-110.

[16] L. Wang, T. Jiang, and D. Gusfield, A more efficient approximation scheme for tree alignment, SIAM J. Comput. **30**(2001) 283-299.

[17] L. Wang, T. Jiang, and E.L. Lawler, Approximation algorithms for tree alignment with a given phylogeny, Algorithmica. **16**(1996) 302-315.

[18] A. Zelikovsky. Better Approximation Bounds for the Network and Eucledian Steiner tree problems. Technical Report 96-06, Department of Computer Science, U. of Virginia (1996).