Correcting Gene Tree by Removal and Modification: Tractability and Approximability[☆]

Stefano Beretta[a,b,*], Mauro Castelli[c], Riccardo Dondi[d]

[a] *Istituto di Tecnologie Biomediche, Consiglio Nazionale delle Ricerche, Segrate - Italia*
[b] *Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, Milano - Italia*
[c] *NOVA IMS, Universidade Nova de Lisboa, Lisboa - Portugal*
[d] *Dipartimento di Scienze Umane e Sociali, Università degli Studi di Bergamo, Bergamo - Italia*

Abstract

Gene tree correction with respect to a given species tree is a problem that has been recently proposed in order to better understand the evolution of gene families. One of the combinatorial methods proposed to tackle with this problem aims to correct a gene tree by removing the minimum number of leaves/labels (Minimum Leaf Removal and Minimum Label Removal, respectively). The two problems have been shown to be APX-hard, and fixed-parameter tractable, when parameterized by the number of leaves/labels removed. In this paper, we focus on the approximation complexity of these two problems and we show that they are not approximable within factor $b \log m$, where m is the number of leaves of the species tree and $b > 0$ is a constant. Furthermore, we introduce and study two new variants of the problem, where the goal is the correction of a gene tree with the minimum number of leaf/label modifications (Minimum Leaf Modification and Minimum Label Modification, respectively). We show that the two modification problems, differently from the removal versions, are unlikely to be fixed-parameter tractable. More precisely, we prove that the Minimum Leaf Modification problem is $W[1]$-hard, when parameterized by the number of leaf modifications, and that the Minimum Label Modification problem is $W[2]$-hard, when parameterized by the number of label modifications.

Keywords: Computational Biology, Gene Tree Reconciliation, Gene Tree Correction, Approximation Complexity, Parameterized Complexity

1. Introduction

Macro-evolutionary events, like duplications and losses, are crucial evolutionary events for genome evolutions [2, 3]. In particular, due to duplications, many gene copies can be found inside a genome. A *gene family* consists of those gene copies originating from duplications of a single gene.

Given a gene family, a first step to understand its evolutionary history is to construct a phylogeny, called *gene tree*, that represents the evolution associated with different gene families in a given set of species. Usually, gene trees are built based on the similarity of the associated sequences. Then, the gene tree is compared to a species tree, which is a phylogeny that represents the *speciation history* of the genomes of the considered species, hence it is based on a model that considers only speciations as evolutionary events. The comparison of a gene tree and a species tree is known as *reconciliation* [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], and has the goal of inferring the macro-evolutionary events (duplications, losses, and in some cases lateral gene transfers) that occurred during evolution.

[☆] A preliminary and abridged version of this paper has appeared in [1]

[*] Corresponding author
Email addresses: stefano.beretta@itb.cnr.it (Stefano Beretta), mcastelli@novaims.unl.pt (Mauro Castelli), riccardo.dondi@unibg.it (Riccardo Dondi)

When no species tree is known, then the definition of the problem changes: starting from a set of possibly discordant gene trees, it asks to infer a correct species tree, usually based on a parsimonious evolutionary scenario [16, 6, 17].

It has been observed that reconciliation is highly sensitive to errors in the gene trees. Indeed, few errors can produce a completely misleading evolutionary scenario, which usually leads to a greater number of duplications and losses [18, 10]. Hence, in order to avoid such a drawback, gene trees have to be corrected before the reconciliation process.

Some of the approaches presented in literature for gene tree correction, compute a set of possible candidate solutions, obtained from the given gene tree with rearrangement operations on the tree structure (for example nearest neighbour interchange) [19, 7, 20]. The goal of this procedure is the computation of a gene tree with the minimum number of duplications and losses.

Errors in gene trees can be related to a special kind of duplications, called *Non-Apparent Duplications* (NAD) [6]. NAD nodes are considered as potential results of errors in a gene tree, since each NAD node represents a contradiction between the structure of a gene tree and a species tree that is not directly explainable by gene duplications. Motivated by this observation, some recent approaches aim to correct a gene tree by modifying its structure via polytomy refinement [21] or by removing misplaced leaves/labels [22].

The combinatorial approach considered in [22, 23], asks for the minimum number of leaves/labels to be removed so that the computed gene tree does not contain NAD nodes. In [22, 23], the complexity of two combinatorial problems related to the removal of leaves/labels (Minimum Leaf Removal and Minimum Label Removal) has been investigated. In [23] the two problems have been shown to be APX-hard, even when each label has at most two occurrences in the gene tree. When the gene tree contains no duplicated leaves, then the problems are related to the Maximum Agreement Subtree of two trees, hence they are polynomial time solvable [22]. Moreover, if the problems are parameterized by the number of leaves/labels removed, then both Minimum Leaf Removal and Minimum Label Removal are fixed-parameter tractable [23].

In this paper, we further extend the approximability aspects of Minimum Leaf Removal and Minimum Label Removal problems by strengthening the results obtained in [23]. More precisely, we prove that these two problems are not approximable within factor $b \log m$ (while in [23] they were proved to be APX-hard). Moreover, we investigate the complexity of two other variants of the gene tree correction problem, where, instead of removing leaves/labels, we correct the gene tree by modifying its leaves/labels. First, we study the approximation complexity of Minimum Leaf Removal and of Minimum Label Removal. We show in Section 3 and in Section 4 that these two problems are not approximable within factor $b \log m$, for some constant $b > 0$, where m is the number of leaves of the species tree, even when each label has at most two occurrences in the input gene tree. Then, we consider two new variants of the problem, called Minimum Leaf Modification and Minimum Label Modification. The aim of these new variants is to correct the given gene tree by modifying the minimum number of leaves (labels, respectively). We show in Section 5 that Minimum Leaf Modification problem, differently from the removal version, is $W[1]$-hard, when parameterized by the number of leaf modifications. Then, we show in Section 6 that Minimum Label Modification, differently from the removal version, is $W[2]$-hard. We refer the reader to [24] for a formal description of the W hierarchy of the parameterized complexity classes, and we just recall that a problem $W[t]$-hard, for $t \geq 1$, is commonly assumed not to be fixed-parameter tractable. Moreover, the last reduction implies that the Minimum Label Modification problem is also not approximable within factor $b \log m$, for some constant $b > 0$, where m is the number of leaves of the species tree.

The inapproximability results we have obtained in this paper implies that designing efficient heuristics with a provable approximation ratio (for example constant), with respect to an optimal solution, is unlikely. On the other hand, the W-hardness of the new introduced problems implies that designing exact algorithms, having the complexity exponentially depending only on the number of leaf/label modifications, is not a promising research direction.

2. Preliminaries

In this section, we introduce some preliminary definitions that will be useful in the rest of the paper.

Consider a set $\Lambda = \{1, 2, \ldots, m\}$ of integers, each one representing a different species. Consider a tree R, then we denote by $L(R)$ the set of its leaves, by $\Lambda(R)$ the set of labels associated with $L(R)$. Given an internal node x of R, x_l (x_r, respectively) denotes the left child (the right child, respectively) of x. $R[x]$ denotes the subtree of R rooted at node x, and $\Lambda(R[x])$ denotes the set of labels associated with leaves of $R[x]$. When there is no ambiguity on the tree, we consider $\mathcal{C}(x) = \Lambda(R[x])$ (we call $\mathcal{C}(x)$ the *cluster* of x). Any node on the path from the root of R to a node x is called an *ancestor* of x; the *parent* y of x is the ancestor of x such that (y, x) is an arc of R.

In this paper, we consider two kinds of rooted binary trees leaf-labeled by the elements of Λ: *species trees* and *gene trees*. For a *species tree* T there exists a bijection from $L(T)$ to Λ (hence each element of Λ labels exactly one leaf of T). For a *gene tree* G there exists a function from $L(G)$ to Λ (hence each element of Λ may label more than one leaf of G). In the rest of the paper, we denote by m the size of $L(T)$ and by n the size of $L(G)$.

Given a tree R, a *leaf removal* of leaf l consists of: (1) removing l from R, and (2) contracting the resulting node having degree two (that is the parent of l). A tree R' obtained from a tree R through a sequence of leaf removals, is said to be *included* in R. Given a set $X \subseteq \Lambda(R)$, we denote by $R|X$ the *homomorphic restriction* of subtree R to X, that is the subtree of R obtained by a sequence of leaf removals, one for every leaf with a label in $\Lambda(R) \setminus X$. Moreover, a *label removal* of label $\lambda \in \Lambda(R)$ consists of: (1) removing all the leaves of R associated with λ, and (2) starting from the leaves, contracting the resulting nodes of the tree having degree at most two.

We compare a gene tree G and a species tree T both leaf-labeled by Λ by means of the *LCA mapping* (Least Common Ancestor mapping), denoted as $\text{lca}_{G,T}$. More precisely, $\text{lca}_{G,T}$ maps every node x of G to a node of T. Formally, for every node x of G, $\text{lca}_{G,T}(x) = y$, where y is the node of T such that (1) $\mathcal{C}(y) \supseteq \mathcal{C}(x)$, and (2) $\mathcal{C}(y_l) \not\supseteq \mathcal{C}(x)$, $\mathcal{C}(y_r) \not\supseteq \mathcal{C}(x)$. A node x of G is a *duplication node* (or a duplication occurs in x), when x and at least one of its children are mapped by $\text{lca}_{G,T}$ to the same node y of the species tree T. A node of G, which is not a duplication node, is a *speciation* node.

Consider a duplication node x. Then if $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) \neq \emptyset$, x is called an *Apparent Duplication node* (*AD node*). It can be easily shown that if x is an AD node, then x is a duplication node for any species tree T. A duplication node x which is not an AD node, that is when $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) = \emptyset$, is called a *Non-Apparent Duplication node* (*NAD node*). As observed in [22, 6], NAD nodes are related to errors in the gene tree. In fact, each NAD node generates a contradiction with the species tree which does not correspond to the presence of duplicated gene copies. A gene tree G is said to be *consistent* with a species tree T if and only if each node of G is either a speciation or an AD node.

Therefore, the following combinatorial problems, Minimum Leaf Removal Problem and Minimum Label Removal, have been introduced in [22, 23] for error-correction in gene trees.

Problem 1. Minimum Leaf Removal Problem[MinLeafRem]
Input: A gene tree G and a species tree T, both leaf-labeled by Λ.
Output: A gene tree G^* consistent with T such that G^* is obtained from G by a minimum number of leaf removals.

Problem 2. Minimum Label Removal Problem[MinLabRem]
Input: A gene tree G and a species tree T, both leaf-labeled by Λ.
Output: A gene tree G^* consistent with T such that G^* is obtained from G by a minimum number of label removals.

Moreover, we introduce two new combinatorial problems, where we modify, instead of removing, leaves/labels of the gene tree so that the resulting tree is consistent with the given species tree. Given a leaf x of G labeled by $\lambda_x \in \Lambda$, a *leaf modification* consists of replacing λ_x with a label in $\Lambda \setminus \{\lambda_x\}$. A *label modification* of a label $\lambda \in \Lambda$ consists of replacing λ with a label in $\Lambda \setminus \{\lambda\}$, that is, each occurrence of label λ in the leaves of the tree G is replaced with a label in $\Lambda \setminus \{\lambda\}$.

Problem 3. Minimum Leaf Modification Problem[MinLeafMod]
Input: A gene tree G and a species tree T, both leaf-labeled by Λ.

Output: A gene tree G^* consistent with T such that G^* is obtained from G by a minimum number of leaf modifications.

Problem 4. Minimum Label Modification Problem [MinLabelMod]
Input: A gene tree G and a species tree T, both leaf-labeled by Λ.
Output: A gene tree G^* consistent with T such that G^* is obtained from G by a minimum number of label modifications.

3. Inapproximability of MinLeafRem

In this section, we consider the approximation complexity of the MinLeafRem problem. We show that the problem is not approximable within factor $c \log m$, for some constant $c > 0$, even when each label has at most two occurrences in the gene tree (we denote this restriction of MinLeafRem as MinLeafRem(2)). Inspired by the reduction presented in [23], we give a gap-preserving reduction from the Minimum Set Cover (MinSC) problem. A gap-preserving reduction for two minimization problems is a reduction from a problem which is known to be inapproximable (hence it is NP-hard to decide whether an instance admits an optimal solution of value at most h or at least ch, where c is the *gap*) to a second problem such that the gap is preserved (that is, for the second problem it is NP-hard to decide whether an instance admits an optimal solution of at most value h' or at least $c'h'$, where c' is the *gap*). We refer the reader to [25] for details on gap-preserving reduction. We recall that MinSC, given a collection $\mathcal{F} = \{S_1, \ldots, S_p\}$ of sets over a finite set $U = \{u_1, \ldots u_q\}$, asks for a minimum subcollection $\mathcal{F}'$ of $\mathcal{F}$ such that each $u_x \in U$ belongs to at least one set of $\mathcal{F}'$. Notice that MinSC is known to be not approximable in polynomial time within factor $b \log q$, for some constant $b > 0$ [26].

Let $(\mathcal{F}, U)$ be an instance of MinSC. In the following, we define an instance of MinLeafRem(2) associated with $(\mathcal{F}, U)$, consisting of a gene tree G and a species tree T, both leaf-labeled by a set Λ.

First, we define the set Λ of labels. For each element $u_i \in U$, let $d(u_i) = |\{S_j : u_i \in S_j, 1 \leq j \leq p\}|$. Moreover, set $k = p^2q^2$, and $t = pk + 2pq + 1$. The set Λ is defined as:

$$\Lambda = \left(\bigcup_{j=1}^{p} A_j \cup B_j \right) \cup \left(\bigcup_{i=1}^{q} U_i \right) \cup Z \cup \{\alpha\}$$

where the sets A_j, B_j, with $1 \leq j \leq p$, U_i, $1 \leq i \leq q$, and Z are defined as follows:

- $A_j = \{a_{j,l} : 1 \leq l \leq k\}$, with $1 \leq j \leq p$;
- $B_j = \{b_{j,l} : u_l \in S_j\} \cup \{b'_{j,l} : 1 \leq l \leq q - |S_j|\}$, with $1 \leq j \leq p$;
- $U_i = \{u_{i,l} : 1 \leq l \leq t\} \cup \{u'_{i,l} : 0 \leq l \leq p - d(u_i)\}$, with $1 \leq i \leq q$;
- $Z = \{z_i : 1 \leq i \leq t\}$.

Let R be a tree, which is either the gene tree G, the species tree T, or a tree included in G with a leaf labeled by α. The *spine* of R is the unique path that connects the root of R to the unique leaf of R labeled by α.

The gene tree G is shown in Fig. 1. It consists of the following subtrees connected to the spine of G (starting from the farthest from the root):

1. a subtree $G(S_j)$, for each set S_j in $\mathcal{F}$, where $\Lambda(G(S_j)) = A_j \cup B_j$;
2. t leaves, each one labeled by a distinct z_i, with $1 \leq i \leq t$;
3. a collection of t subtrees $G_1(u_i)$, …, $G_t(u_i)$, for each $u_i \in U$. Subtree $G_1(u_i)$ is leaf labeled by the set $\{u_{i,1}\} \cup \{u'_{i,l} : 0 \leq l \leq p - d(u_i)\} \cup \{b_{j,i} : u_i \in S_j\}$ and subtree $G_l(u_i)$, with $2 \leq l \leq t$, is leaf labeled by the set $\{u_{i,l-1}, u_{i,l}\}$.

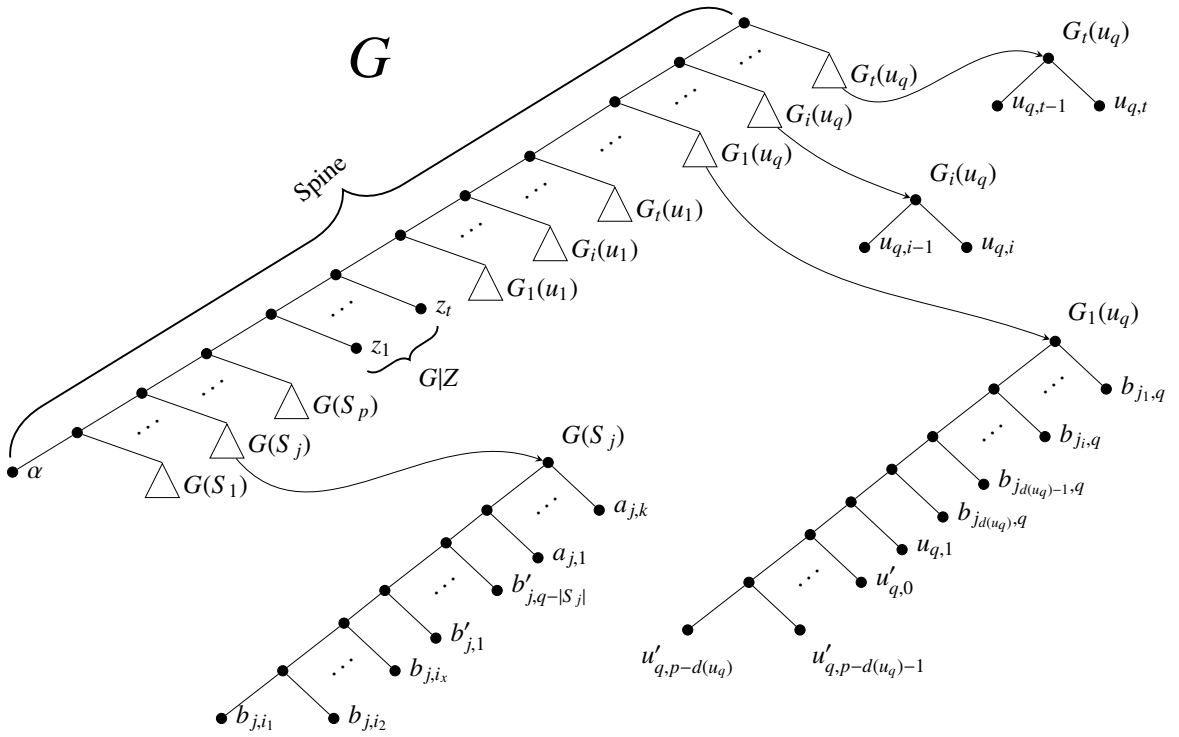

Figure 1: The gene tree G and the subtrees $G(S_j)$, $G_1(u_q)$, $G_i(u_q)$, and $G_t(u_q)$. Notice that in $G_1(u_q)$ the leaves $b_{j_{d(u_q)},q}, b_{j_{d(u_q)-1},q}, \ldots, b_{j_i,q}, \ldots, b_{j_1,q}$ refer to the sets $S_{j_{d(u_q)}}, S_{j_{d(u_q)-1}}, \ldots, S_{j_i}, \ldots, S_{j_1}$, respectively, containing u_q (with $j_{d(u_q)} > j_{d(u_q)-1} > \cdots > j_i > \cdots > j_1$). Moreover, in $G(S_j)$ the leaves $b_{j,i_1}, b_{j,i_2}, \ldots, b_{j,i_x}$ refer to $u_{i_1}, u_{i_2}, \ldots, u_{i_x} \in S_j$.

Similarly, T is shown in Fig. 2 and it consists of the following subtrees connected to the spine of T (starting from the farthest from the root):

1. a subtree $T(S_j)$, for each set $S_j \in \mathcal{F}$, where $\Lambda(T(S_j)) = A_j \cup B_j$;
2. t leaves, each one associated with a distinct label in U_i;
3. t leaves, each one labeled by a distinct z_i, with $1 \leq i \leq t$.

It is easy to see that T is a species tree uniquely leaf-labeled by Λ. The gene tree G is leaf-labeled by Λ, and each label in Λ is associated with at most two leaves of G. Indeed, the sets of labels associated with more than one leaf are $\{b_{j,i} : u_i \in S_j\}$ ($b_{j,i}$ labels one leaf of the subtree $G(S_j)$ and one leaf of the subtree $G(u_i)$), and $\{u_{i,l} : 1 \leq l \leq t-1\}$ ($u_{i,l}$ labels one leaf of the subtree $G_l(u_i)$ and one leaf of the subtree $G_{l+1}(u_i)$).

Before giving the details of the proof, we present an outline of the reduction. First, we prove some local properties of the subtrees $G(S_j)$, with $S_j \in \mathcal{F}$: in Remark 1 and in Lemma 1, we show that a solution of MinLeafRem(2) over instance (G, T) can be computed by removing leaves from $G(S_j)$, in (essentially) two possible ways: the set of leaves labeled by A_j or the set of leaves labeled by B_j. Then, exploiting some properties of the subtrees $G_l(u_i)$, with $u_i \in U$ and $1 \leq l \leq t$, and by Lemma 2 and Lemma 4, we are able to relate the former case (the removal of leaves labeled by A_j) to a set S_j in a set cover (see Lemma 5), and the latter case (the removal of leaves labeled by B_j) to a set S_j not in a set cover (see Lemma 6). First, we introduce two preliminary properties of G and T.

Remark 1. *Let S_j be a set of $\mathcal{F}$, and let $G(S_j)$ ($T(S_j)$, respectively) be the subtree of G (of T, respectively) associated with S_j. Then (1) the subtree of $G(S_j)$ obtained by removing the leaves with labels in A_j is consistent with $T(S_j)$; (2) the subtree of $G(S_j)$ obtained by removing the leaves with labels in B_j is consistent with $T(S_j)$.*

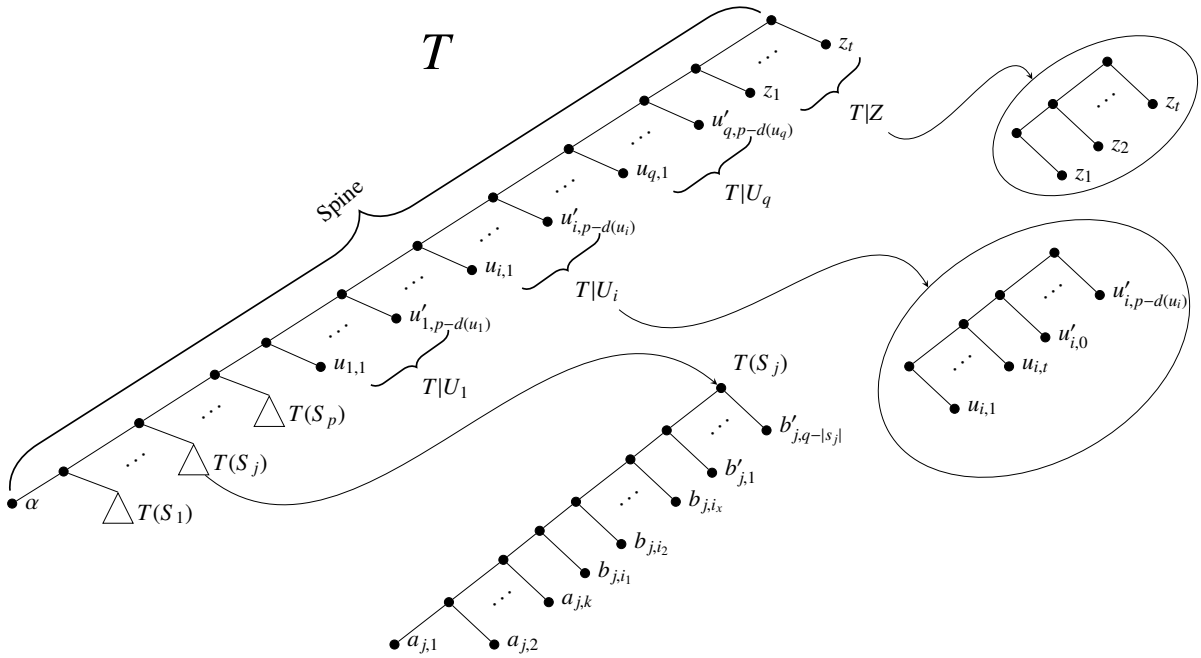

Figure 2: The species tree T and its subtrees $T|Z$, $T|U_i$, and $T(S_j)$. Notice that in in $T(S_j)$ the leaves $b_{j,i_1}, b_{j,i_2}, \ldots, b_{j,i_x}$ refer to $u_{i_1}, u_{i_2}, \ldots, u_{i_x} \in S_j$.

Proof. The proof follows from the observation that the trees $G(S_j)|B_j$ and $T(S_j)|B_j$ are isomorphic, and that the trees $G(S_j)|A_j$, $T(S_j)|A_j$ are isomorphic.

□

Next, we introduce a property of the subtrees $G(S_i)$ of G, with $S_i \in \mathcal{F}$.

Lemma 1. *Let S_j be a set of $\mathcal{F}$, and consider the corresponding subtrees $G(S_j)$ of G and $T(S_j)$ of T. Then: (1) a solution of MinLeafRem(2) over instance (G,T) is obtained by removing at least q leaves from $G(S_j)$; (2) a solution of MinLeafRem(2) over instance (G,T) that contains a leaf of $G(S_j)$ with a label in B_j is obtained by removing at least k leaves from $G(S_j)$.*

Proof. (1) Assume that G^* is a solution of MinLeafRem(2) over instance (G,T) that it is obtained by removing less than q leaves from $G(S_j)$. It follows that G^* contains a subtree $G^*(S_j)$ (included in $G(S_j)$) that contains a leaf with a label in B_j, and at least $k-q+1$ leaves of $G(S_j)$ with label in A_j (where $k-q+1 \geq q \geq 2$). Then, if $G^*(S_j)$ contains a leaf labeled by B_j and two leaves labeled by A_j, by construction it contains a NAD node. Indeed, among the leaves of $G^*(S_j)$ with a label in B_j, let l_x be the leaf of $G^*(S_j)$ which is the closest to the root of $G^*(S_j)$. Denote with w_x the parent of l_x in $G^*(S_j)$. By construction, each node x of $G^*(S_j)$ which is a parent of a leaf of $G^*(S_j)$ with a label in A_j, is mapped by $\text{lca}_{G^*,T}$ in the same node of T where w_x is mapped. Hence, every node x would represent a NAD node. Since $G^*(S_j)$ must contain at least $k-q+1$ leaves with a label in A_j, it follows that a solution of MinLeafRem(2) over instance (G,T) removes at least q leaves from $G(S_j)$ (the leaves having labels in B_j).

(2) If $G^*(S_j)$ contains more than one leaf with a single label in B_j, then it must contain no leaf with a label in A_j, otherwise by construction $G^*(S_j)$ would have a NAD node. Hence, in this case at least k leaves are removed from $G(S_j)$. Now, if $G^*(S_j)$ contains exactly one leaf with a single label in B_j, then it contains at most one leaf with a label in A_j, hence in this case more than k leaves are removed from $G(S_j)$.

□

Now, we show that we can assume that a solution of MinLeafRem(2) over instance (G,T) contains all the leaves of G with a label in Z.

Lemma 2. *Given a solution G^* of MinLeafRem(2) over instance (G, T) that is obtained by removing less than t leaves from G and in which a leaf with a label in Z is removed, then we can compute in polynomial time a solution of MinLeafRem(2) over instance (G, T) that is obtained by removing less leaves than G^* and contains all the leaves with labels in Z.*

Proof. Let G^* be a solution of MinLeafRem(2) over instance (G, T) obtained from G by removing less than t leaves. Notice that, since $|Z| = t$, at least one leaf with a label in the set Z must be in G^*. Assume that G^* is obtain by removing a leaf with label z_l, with $1 \leq l \leq t$, from G. It is easy to see that the insertion of this leaf into G^* (so that the order of the leaves labeled by Z is the same as in G) does not affect other nodes of G^*, that is the insertion of the leaf with label z_l does not create any NAD node.

□

Remark 2. *Given an instance (G, T) of MinLeafRem(2), we can always compute in polynomial time a solution having cost less than t.*

Proof. Consider the following subtree G^* included in G and consistent with T:

- for each subtree $G(S_J)$ remove all the leaves having labels $a_{j,1}, \ldots, a_{j,k}$;
- for each subtree $G_1(u_i)$ remove all the leaves except for those having labels $u_{i,1}$ and $b_{j_1,i}$.

The solution G^* is obtained by removing $kp + pq$ leaves (k leaves from each subtree $G(S_j)$, plus p leaves from each subtree $G_1(u_i)$). Since $t = pk + 2pq + 1$ it follows that G^* is obtained by removing less than t leaves.

□

Hence, by Remark 2 we can always compute a solution with less than t leaf removals. In what follows, by Lemma 2 we assume that all the leaves with a label in Z belong to G^*. Now, for each $u_i \in U$, we introduce some properties of the subtree $G_1(u_i)$ (Lemma 3), and of the subtrees $G_l(u_i)$, with $1 \leq l \leq t$ (Lemma 4). This latter lemma implies that a solution contains all the leaves labeled by $u_{i,l}$, with $1 \leq l \leq t$, for each $u_i \in U$.

Lemma 3. *Given $u_i \in U$, let $G_1(u_i)$ be the associated subtree of G. Each solution of MinLeafRem(2) over instance (G, T) is obtained by removing at least p leaves from $G_1(u_i)$.*

Proof. Let G^* be a solution of MinLeafRem(2) over instance (G, T), and let $G_1^*(u_i)$ be the subtree of G included in $G_1(u_i)$. Assume that $G_1^*(u_i)$ contains an internal node, that is, it contains at least two leaves. Consider the internal node x of $G_1^*(u_i)$ which is the farthest from the root of $G_1^*(u_i)$. Let y be the node of T where x is mapped to by $\text{lca}_{G^*,T}$. Notice that y is a node on the spine of T. By construction, since the order of the leaves in $G_1^*(u_i)$ is opposite with respect to T, then $\text{lca}_{G^*,T}$ maps all the internal nodes of $G_1^*(u_i)$ to y. Since $G_1(u_i)$ (and also $G_1^*(u_i)$) is uniquely leaf labeled, then any internal node $G_1^*(u_i)$ other than x would be a NAD node. Hence, $G_1^*(u_i)$ must contain at most one internal node and thus $G_1^*(u_i)$ contains at most two leaves. Finally, since $G_1(u_i)$ contains $p + 2$ leaves, at least p leaves are removed from $G_1(u_i)$.

□

Lemma 4. *Let u_i be an element of U and let $G_1(u_i)$, $G_2(u_i)$, …, $G_t(u_i)$ be the associated subtrees of G, with $1 \leq i \leq q$. Then, the subtree $G_1^*(u_i)$ of G^* contains the leaf labeled by $u_{i,1}$.*

Proof. Let G^* be a tree included in G and consistent with T. From Lemma 2, it follows that each leaf with a label in Z belongs to G^*, hence each node x on the spine of G^* above Z is mapped to the root of T and must be an AD node, so it must hold $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) \neq \emptyset$.

Assume that G^* is obtained by removing from $G_1(u_i)$ the leaf labeled by $u_{i,1}$, and it contains a subtree $G_2^*(u_i)$ of $G_2(u_i)$, such that a leaf of $G_2(u_i)$ is not removed. Let y be the node on the spine of G^* connected to the root of $G_2^*(u_i)$. Since $G_1^*(u_i)$ does not contain u_i, then it holds $\mathcal{C}(y_l) \cap \mathcal{C}(y_r) = \emptyset$, and all the leaves of $G_2(u_i)$ must be removed. The same argument holds for each of the subtrees $G_l(u_i)$, with $3 \leq l \leq t$. Hence, if G^* is obtained by removing from $G_1(u_i)$ the leaf labeled by $u_{i,1}$, then each leaf of $G_l(u_i)$, with $2 \leq l \leq t$ and

at least one leaf of $G(u_1)$ must be removed by G^*, leading to an overall number of $2t-1$ leaves removed to obtain G^* (which contradicts the assumption that each solution is obtained with less than t leaf removals). □

Now, we are ready to show the two main technical results of the reduction.

Lemma 5. *Let $(\mathcal{F}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLeafRem(2). Then, starting from a set cover $\mathcal{F}'$ of U, we can compute in polynomial time a feasible solution of MinLeafRem(2) over instance (G, T) that it is obtained by removing exactly $k|\mathcal{F}'| + q(|\mathcal{F}| - |\mathcal{F}'|) + pq$ leaves from G.*

Proof. Let $\mathcal{F}'$ be a set cover of $(\mathcal{F}, U)$, then we define a feasible solution G^* of MinLeafRem(2) over instance (G, T) by removing some leaves of G as follows:

- for each S_i in $\mathcal{F}'$, remove from the subtree $G(S_i)$ the set of leaves labeled by A_i (hence this subtree $G^*(S_i)$ of G^* has leafset labeled by B_i and k leaves are removed from $G(S_i)$);
- for each S_i not in $\mathcal{F}'$, remove from the subtree $G(S_i)$ the set of leaves labeled by B_i (hence this subtree $G^*(S_i)$ of G^* has leafset labeled by A_i and q leaves are removed from $G(S_i)$);
- for each $u_i \in U$, remove from $G_1(u_i)$ all the leaves, except for the leaf labeled by $u_{i,1}$ and a leaf labeled by $b_{j,i}$, where $u_i \in S_j$ and $S_j \in \mathcal{F}'$ (hence this subtree $G_1^*(u_i)$ of G^* has leafset labeled by $u_{i,1}$ and $b_{i,j}$ and p leaves are removed from $G_1(u_i)$). Notice that, since there could exist several leaves $b_{i,j}$ for which the previous conditions are satisfied, we arbitrary choose one of them.

Next, we show that the gene tree G^* included in G is consistent with T.

By Remark 1, the subtree $G^*(S_i)$, with $1 \leq i \leq p$, is consistent with T. Furthermore, by construction, since each subtree $G_l^*(u_i)$, with $1 \leq l \leq t$, consists of two leaves, it follows that it is consistent with T. Hence, the only nodes left are those on the spine of G^*.

In the following we show that each node on the spine of G^* is either a speciation node or an AD node. Indeed, by construction each node x on the spine of G^* such that $\mathcal{C}(x) \not\supseteq Z$ is a speciation node. Each node x on the spine of G^*, and such that $\mathcal{C}(x) \supseteq Z$ is a duplication node. First, consider the node x that connects a subtree $G_1^*(u_i)$ to the spine of G^*. Since element $u_i \in U$ is covered by a set of $\mathcal{F}'$, it follows that $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) = b_{j,i}$, for some set $S_j \in \mathcal{F}'$, hence x is an AD node. Now, consider a node x that connects a subtree $G_l^*(u_i)$, with $2 \leq l \leq t$, to the spine of G^*. Since no leaf labeled by $u_{i,l}$, with $1 \leq j \leq t$, is removed from the trees $G_1^*(u_i)$, ..., $G_t^*(u_i)$ it follows that x is an AD node.

The feasible solution G^* is obtained by removing k leaves from each subtree $G(S_i)$ associated with a set S_i in $\mathcal{F}'$, q leaves from each subtree $G(S_i)$ associated with a set S_i not in $\mathcal{F}'$, and p leaves from each subtree $G_1(u_i)$, with $u_i \in U$. It follows that G^* is obtained by removing $k|\mathcal{F}'| + q(|\mathcal{F}| - |\mathcal{F}'|) + pq$ leaves from G. □

Lemma 6. *Let $(\mathcal{F}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLeafRem(2). Then, for every h such that $1 \leq h \leq p$, starting from a solution of MinLeafRem(2) over instance (G, T) that is obtained by removing at most $kh + q(|\mathcal{F}| - h) + pq$ leaves, we can compute in polynomial time a solution of MinSC over instance $(\mathcal{F}, U)$ that consists of at most h sets.*

Proof. Let G^* be a solution of MinLeafRem(2) over instance (G, T), such that G^* is obtained by removing at most $kh + q(|\mathcal{F}| - h) + pq$ leaves. First, by Remark 2, G^* is obtained with less than t removals. Then, by Lemma 2 we can assume that all the leaves with labels in Z belong to G^*.

Furthermore, we prove the following claim in order to relate the solution G^* of MinLeafRem(2) to a cover of $(\mathcal{F}, U)$.

Claim 1. *Each $G_1^*(u_i)$ contains exactly two leaves labeled by $u_{i,1}$ and $b_{j,i}$, for some S_j such that $u_i \in S_j$.*

Proof. By Lemma 4 each $G_1^*(u_i)$ must contain the leaf labeled by $u_{i,1}$. This implies that $G_1^*(u_i)$ must also contain a leaf labeled by $b_{j,i}$ that labels a leaf of a subtree $G^*(S_j)$, otherwise the node on the spine of G^* connected to the root of $G_1^*(u_i)$ would be a NAD node.

□

From Claim 1 and Lemma 3 it follows that exactly p leaves are removed from each $G_1(u_i)$, with $1 \leq i \leq t$. Notice that by Lemma 4, $G_1^*(u_i)$ contains the leaf labeled by $u_{i,1}$. Assume that the subtrees $G_2^*(u_i), \ldots, G_t^*(u_i)$ are computed without any leaf removal. Then, by construction each of these subtrees is consist with T. Moreover, each node on the spine of G^* connecting the subtrees $G_2^*(u_i), \ldots, G_t^*(u_i)$ is an AD node. Hence, we can assume that all the leaves with a label $u_{i,w}$, with $1 \leq i \leq q$ and $1 \leq w \leq t$, belong to G^*.

Finally, consider a subtree $G^*(S_j)$, with $1 \leq j \leq p$. By Lemma 1, we can assume that either $G^*(S_j)$ has leafset B_j or it has leafset A_j. As a consequence we can define a cover $\mathcal{F}'$ of U as follows:

$$\mathcal{F}' = \{S_j : \Lambda(G^*(S_j)) = B_i\}.$$

Since at most $kh + q(|\mathcal{F}| - h) + pq$ leaves are removed from G, it follows that G^* contains at most h subtrees $G^*(S_j)$, with $\Lambda(G^*(S_j)) = B_i$, hence $|\mathcal{F}'| \leq h$. Notice that $\mathcal{F}'$ covers each element of U. Indeed, by Claim 1 $G_1^*(u_i)$ must contain a leaf labeled by $b_{j,i}$. Moreover, since the node on the spine of G^* connecting the subtree $G_1^*(u_i)$ must be an AD node, $b_{j,i}$ labels also a leaf of a subtree $G^*(S_j)$, with $u_i \in S_j$.

□

The inapproximability of MinLeafRem(2) follows from Lemma 5 and Lemma 6.

Theorem 1. *MinLeafRem(2) is not approximable within factor $c \log m$, for some constant $c > 0$, where $m = \Lambda$.*

Proof. Given an instance $I = (\mathcal{F}, U)$ of MinSC, let $J = (G, T)$ be the corresponding instance of MinLeafRem(2). We denote by $OPT_{MinSC}(I)$ ($OPT_{MinLeafRem}(J)$, respectively) the value of an optimal solution of MinSC (that is the number of leaf removals in an optimal solution of MinLeafRem(2), respectively) over the instance I (over the instance J corresponding to I, respectively).

The MinSC problem is known to be inapproximable within factor $b \ln q$, for some constant $b > 0$, where $q = |U|$. This implies that, given an instance of MinSC, it is NP-hard to decide whether the instance admits an optimal solution of value at most $f(I)$, for some function $f(I) \to \mathbb{N}$, or an optimal solution having value at least $f(I) b \ln q$.

Now, let $f : I \to \mathbb{N}$ be a function, we have proved in Lemma 5 that it holds

$$OPT_{MinSC}(I) \leq f(I) \Rightarrow OPT_{MinLeafRem}(J) \leq f(I)k + q(p - f(I)) + pq$$

and, by Lemma 6,

$$OPT_{MinSC}(I) > b \ln q f(I) \Rightarrow OPT_{MinLeafRem}(J) > b \ln q f(I)k + q(p - f(I) b \ln q) + pq,$$

for some constant $b > 0$. Since $k = p^2 q^2$, it follows that $k \geq pq$, and $k \geq q(p - f(I))$. Then, since $p - f(I) b \ln q$ (p is the number of sets and $f(I) b \ln q$ is the size of a set cover),

$$b \ln q f(I)k + q(p - f(I) b \ln q) + pq \geq b \ln q f(I)k = \frac{b}{3} \ln q f(I)k + \frac{b}{3} \ln q f(I)k + \frac{b}{3} \ln q f(I)k$$

that is

$$b \ln q f(I)k + q(p - f(I) b \ln q) + pq \geq \frac{b}{3} \ln q f(I)k + \frac{b}{3} \ln q f(I) q(p - f(I) b \ln q) + \frac{b}{3} \ln q f(I) pq$$

Since $f(I) \geq 1$ and we assume $OPT_{MinSC}(I) > b \ln q f(I)$, it holds:

$$OPT_{MinLeafRem}(J) > \frac{b}{3} \ln q \left(k f(I) + q(p - f(I)) + pq\right)$$

that is

$$OPT_{MinLeafRem}(J) > \frac{\ln q}{d}\left(kf(I) + q(p - f(I)) + pq\right)$$

for some constant $d > 0$. This implies that it is NP-hard to decide if an instance J of MinLeafRem(2) admits an optimal solution with at most $kf(I) + q(p - f(I)) + pq$ leaf removals or if it admits an optimal solution with at least $\frac{\ln q}{d}\left(kf(I) + q(p - f(I)) + pq\right)$ leaf removals. Hence MinLeafRem(2) cannot be approximated within factor $\frac{\ln q}{d}$. Now, notice that $|\Lambda| = m = 2t + 2pq \leq 2p^2q^3 + 5pq + 1$ and that MinSC is known to be inapproximable within factor $b \ln q$, for some constant $b > 0$, when q and p are polynomially related [27]. This implies that $m \leq q^\alpha$, for some constant $\alpha > 0$, and that MinLeafRem(2) cannot be approximated within factor $c \log m$, for some constant $c > 0$.

□

4. Inapproximability of MinLabelRem

In this section, we consider the approximation complexity of the MinLabelRem problem, even when each label has at most two occurrences in the gene tree (we denote this restriction of MinLabelRem as MinLabelRem(2)). By slightly modifying the reduction of Section 3, we show that that MinLabelfRem(2) is not approximable within factor $c \log m$, for some constant $c > 0$, via a gap-preserving reduction from the Minimum Set Cover (MinSC) problem.

Given an instance $(\mathcal{F}, U)$ of MinSC, we construct an instance of MinLabelRem(2) associated with $(\mathcal{F}, U)$, consisting of a gene tree G and a species tree T, both leaf-labeled by a set Λ. Notice that T and Λ are identical to the previous reduction, while G must be modified. More precisely, we will modify subtree $G_1(u_i)$, for each $u_i \in U$. Indeed, notice that, for the construction of the previous section, each feasible solution of MinLeafRem(2) (and similarly of MinLabelRem(2)) contains at most two leaves of $G_1(u_i)$ and one of these leaves is labeled by u_i. Then, notice that if element u_i is covered by two sets, say S_{j_1} and S_{j_2}, then all the leaves labeled by $b_{j_1,i}$ or by $b_{j_2,i}$ must be removed, hence, it is not possible to directly define a relation with a set cover.

In the instance of MinLabelRem(2), we define $G_1(u_i)$ as a subtree whose leaves are (uniquely) labeled by the set $\{u_{i,1}\} \cup \{b_{j,i} : u_i \in S_j\}$, such that the leaves labeled by $b_{j_1,i}, \ldots, b_{j_x,i}$, associated with sets $S_{j_1}, \ldots, S_{j_x}$ containing u_i where $x = deg(u_i)$, are in the same order as in T (see Fig. 3).

Notice that, by construction of G and T, Remark 1 and Lemma 2 hold also in this case.

Next, we prove some properties of the subtrees $G_1(u_i)$, $G_2(u_i)$, …, $G_t(u_i)$.

Lemma 7. *Given $u_i \in U$, let $G_1(u_i)$ be the associated subtree of G. Each solution of MinLabelRem(2) over instance (G, T) either is obtained by removing label $u_{i,1}$ or it contains at least one label $b_{j,i}$.*

Proof. Assume that a solution G^* of MinLabelRem(2) over instance (G, T) contains a subtree $G_1^*(u_i)$ of $G_1(u_i)$ containing a leaf labeled by $u_{i,1}$. Moreover, assume that $G_1^*(u_i)$ contains exactly the leaf labeled by $u_{i,1}$. Then, notice that by construction, the node on the spine connected to $G_1^*(u_i)$ in G^* is a NAD node, since each label associated with a leaf in $G|Z$ is not removed. Assume that $G_1^*(u_i)$ contains more that one leaf. Then, a label $b_{j,i}$ is not removed, and two leaves with label $b_{j,i}$ belong to G^*, one in subtree $G_1^*(u_i)$ and one in subtree $G^*(S_j)$, thus implying that the node on the spine connected to $G_1^*(u_i)$ in G^* is an AD node.
□

Lemma 8. *Let u_i be an element of U and let $G_1(u_i)$, $G_2(u_i)$, …, $G_t(u_i)$ be the associated subtrees of G. If a solution G^* of MinLabelRem(2) over instance (G, T) is obtained by removing the label $u_{i,1}$ of $G_1(u_i)$, then G^* is obtained by removing at least t labels.*

Proof. From Lemma 2, it follows that each leaf with a label in Z belongs to G^*. Let G^* be a tree included in G and consistent with T. Each node x on the spine of G^* over Z must be an AD node, hence it must hold $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) \neq \emptyset$.

Assume that G^* is obtained by removing from $G_1(u_i)$ the label $u_{i,1}$, and contains a subtree $G_2^*(u_i)$ of $G_2(u_i)$, that is $u_{i,2}$ labels a leaf of G^*. Let y be the node on the spine of G^* connected to the root of $G_2^*(u_i)$.

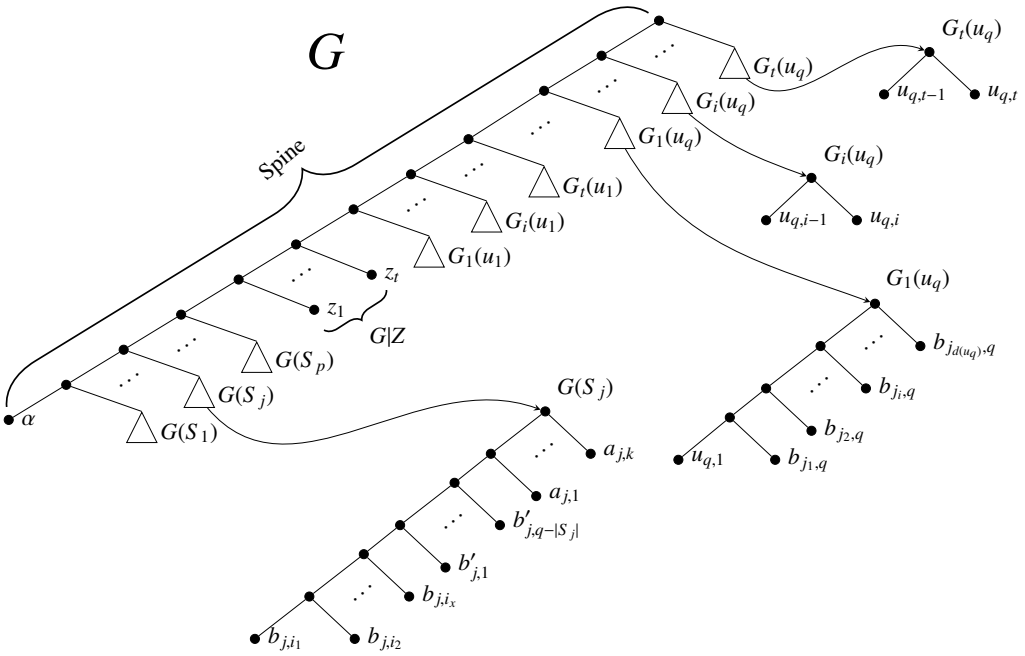

Figure 3: The gene tree G and the subtrees $G(S_j)$, $G_1(u_q)$, $G_i(u_q)$, and $G_t(u_q)$. Notice that in $G_1(u_q)$ the leaves $b_{j_1,q}, b_{j_2,q}, \ldots, b_{j_i,q}, \ldots, b_{j_{d(u_q)},q}$ refer to the sets $S_{j_1}, S_{j_2}, \ldots, S_{j_i}, \ldots, S_{j_{d(u_q)}}$, respectively, containing u_q (with $j_1 < j_2 < \cdots < j_i < \cdots < j_{d(u_q)}$). Moreover, in $G(S_j)$ the leaves $b_{j,i_1}, b_{j,i_2}, \ldots, b_{j,i_x}$ refer to $u_{i_1}, u_{i_2}, \ldots, u_{i_x} \in S_j$.

Then, it holds $\mathcal{C}(y_l) \cap \mathcal{C}(y_r) = \emptyset$, implying that y would be a NAD node. Then, also $u_{i,2}$ would be removed. The same argument holds for each of the subtrees $G_l(u_i)$, with $3 \leq l \leq t$. Hence, if G^* is obtained by removing label $u_{i,1}$, then each label $u_{i,l}$, with $2 \leq l \leq t$, does not belong to G^*, leading to an overall number of t labels which have been removed to obtain G^*.

□

Now, we show the main results of the reduction.

Lemma 9. *Let $(\mathcal{F}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLabelRem(2). Then, starting from a set cover $\mathcal{F}'$ of U, we can compute in polynomial time a solution of MinLabelRem(2) over instance (G, T) that it is obtained by removing at most $k|\mathcal{F}'| + q(|\mathcal{F}| - |\mathcal{F}'|)$ labels from G.*

Proof. Let $\mathcal{F}'$ be a set cover of $(\mathcal{F}, U)$, then we define a solution G^* of MinLbelRem(2) over instance (G, T) by removing some labels of G as follows:

- for each S_i in $\mathcal{F}'$, remove (from the subtree $G(S_i)$) the labels A_i (hence subtree $G^*(S_i)$ of G^* has leafset labeled by B_i and k labels are removed from G);
- for each S_i not in $\mathcal{F}'$, remove from G the set of labels B_i (hence $G^*(S_i)$ of G^* has leafset labeled by A_i and q labels are removed from $G(S_i)$).

Next, we show that the gene tree G^* included in G is consistent with T.

By Remark 1, the subtree $G^*(S_i)$, with $1 \leq i \leq p$, is consistent with T. Furthermore, by construction, each subtree $G^*_l(u_i)$, with $1 \leq l \leq t$ is consistent with T. Hence, the only nodes left to verify are those on the spine of G^*.

By construction, each node on the spine of G^* is either a speciation node or an AD node. Indeed, each node x on the spine of G^* such that $\mathcal{C}(x) \not\supseteq Z$ is a speciation node. Each node x on the spine of G^*, and such that $\mathcal{C}(x) \supseteq Z$ is a duplication node. First, consider the node x that connects a subtree $G^*_1(u_i)$ to the

spine of G^*. Since element $u_i \in U$ is covered by a set of $\mathcal{F}'$, it follows that $b_{j,i} \in \mathcal{C}(x_l) \cap \mathcal{C}(x_r)$, for some set $S_j \in \mathcal{F}'$, hence x is an AD node. Now, consider a node x that connects a subtree $G^*_l(u_i)$, with $2 \leq l \leq t$, to the spine of G^*. Since no label $u_{i,l}$, with $1 \leq j \leq t$, is removed from G, it follows that x is an AD node.

Solution G^* is obtained by removing k labels for each set S_i in $\mathcal{F}'$, q labels for each set S_i not in $\mathcal{F}'$. It follows that G^* is obtained by removing $k|\mathcal{F}'| + q(|\mathcal{F}| - |\mathcal{F}'|)$ labels from G.

□

Lemma 10. *Let $(\mathcal{F}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLabelRem(2). Then, for every h such that $1 \leq h \leq p$, starting from a solution of MinLabelRem(2) over instance (G, T) that is obtained by removing at most $kh + q(|\mathcal{F}| - h)$ labels, we can compute in polynomial time a solution of MinSC over instance $(\mathcal{F}, U)$ that consists of at most h sets.*

Proof. Let G^* be a solution of MinLabelRem(2) over instance (G, T), such that G^* is obtained by removing at most $kh + q(|\mathcal{F}| - h)$ leaves. By Lemma 2, we can assume that all the leaves with labels in Z belong to G^*.

By Lemma 8, we can assume that no label $u_{i,1}$ is removed and hence, that G^* contains all the labels $u_{i,w}$, with $1 \leq i \leq q$ and $1 \leq w \leq t$. Moreover, this fact implies that at least one label $b_{j,i}$ is not removed. Indeed, if no label $b_{j,i}$ belongs to G^* the node on the spine that connects subtree $G^*_1(u_i)$ would be a NAD node.

Finally, consider a subtree $G^*(S_j)$, with $1 \leq j \leq p$. By Lemma 1, we can assume that either $G^*(S_j)$ has leafset B_j or it has leafset A_j.

As a consequence, we can define a cover $\mathcal{F}'$ of U as follows:

$$\mathcal{F}' = \{S_j : \Lambda(G^*(S_j)) = B_i\}.$$

Since G^* is obtained by removing at most $kh + q(|\mathcal{F}| - h)$ labels, it follows that G^* contains at most h subtrees $G^*(S_j)$, with $\Lambda(G^*(S_j)) = B_i$, hence $|\mathcal{F}'| = h$. Notice that $\mathcal{F}'$ covers each element of U, since by Lemma 8 a label $b_{j,i}$ is not removed, hence there exist two leaves labeled by $b_{j,i}$, one in subtree $G^*(S_j)$ and one in subtree $G^*_1(u_i)$.

□

The inapproximability of MinLeafRem(2) follows from Lemma 9 and Lemma 10.

Theorem 2. *MinLabelRem(2) is not approximable within factor $c \log m$, for some constant $c > 0$.*

Proof. The proof is similar to that of Theorem 1. Given an instance $I = (\mathcal{F}, U)$ of MinSC, let $J = (G, T)$ be the corresponding instance of MinLabelRem(2). We denote by $OPT_{MinSC}(I)$ ($(OPT_{MinLabelRem}(J))$, respectively) the value of an optimal solution of MinSC (MinLabelRem(2), respectively) over the instance I (over the instance J corresponding to I, respectively). From Lemma 5 it holds

$$OPT_{MinSC}(I) \leq f(I) \Rightarrow OPT_{MinLabelRem}(J) \leq f(I)k + q(p - f(I))$$

and, by Lemma 6,

$$OPT_{MinSC}(I) > b \ln q f(I) \Rightarrow OPT_{MinLabelRem}(J) > b \ln q f(I)k + q(p - f(I) b \ln q),$$

for some constant $b > 0$. By using bounds similar to those of the proof of Theorem 1, we can show that if $OPT_{MinSC}(I) > b \ln q f(I)$ then

$$OPT_{MinLabelRem} > \frac{\ln q}{d} (kf(I) + q(p - f(I)))$$

for some constant $d > 0$, and this implies that MinLabelRem(2) cannot be approximated within factor $\frac{\ln q}{d}$. Now, notice that $|\Lambda| = m = 2t + 2pq \leq 2p^2q^3 + 5pq + 1$ and that MinSC is known to be inapproximable within factor $b \ln q$, for some constant $b > 0$, when q and p are polynomially related [27]. This implies that $m \leq q^\alpha$, for some constant $\alpha > 0$, and that MinLabelRem(2) cannot be approximated within factor $c \log m$, for some constant $c > 0$.

□

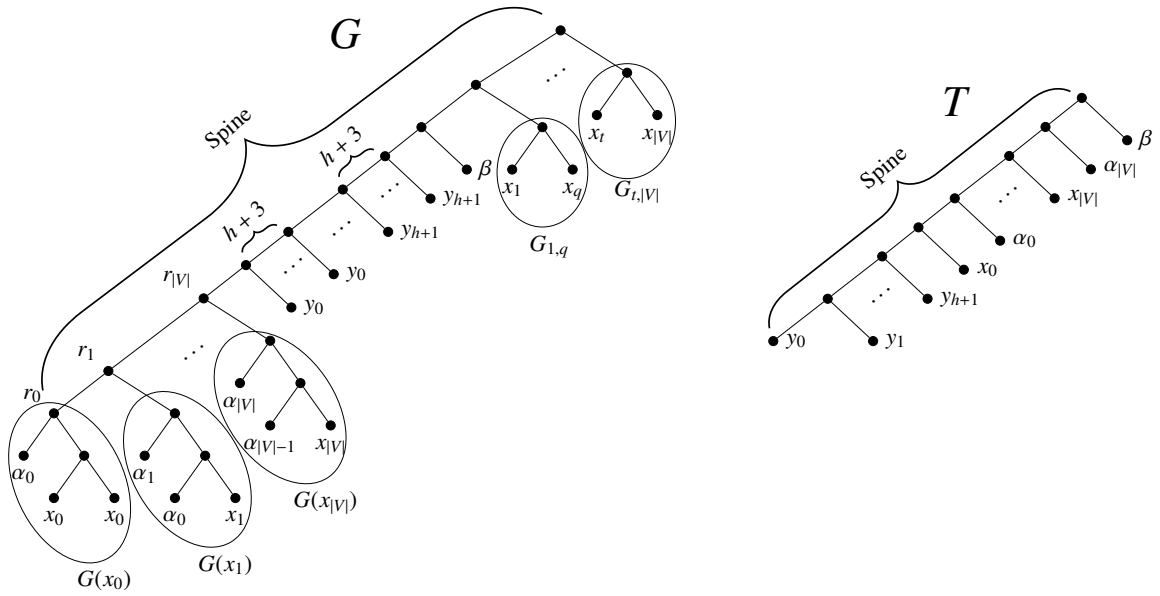

Figure 4: The gene tree G and the species tree T. Notice that the subtrees $G_{1,q}$, $G_{t,|V|}$ encode the edges $\{v_1, v_q\}$, $\{v_t, v_{|V|}\}$ of $\mathcal{G}$, respectively. These subtrees are connected to the spine of G following the lexicographic order of the corresponding edges.

5. W[1]-hardness of MinLeafMod

In this section, we investigate the parameterized complexity of MinLeafMod and we show that the problem is W[1]-hard when parameterized by the number of modified leaves, by giving a parameterized reduction from the Maximum Independent Set (MaxIS) problem. We recall that MaxIS, given a graph $\mathcal{G} = (V, E)$, asks for a subset $V' \subseteq V$ of maximum cardinality such that for each $u, v \in V'$ it holds $\{u, v\} \notin E$. Notice that the parameterized version of MaxIS asks whether there exists an independent set of $\mathcal{G}$ of size at least h. Hence, in what follows h will denote the size of an independent set of $\mathcal{G}$. We recall that MaxIS is known to be W[1]-hard [28].

Remark 3. *Given an instance $\mathcal{G} = (V, E)$ of MaxIS, we assume that the size of an independent set V' is at most $|V| - 3$.*

Proof. Let $\mathcal{G} = (V, E)$ be an instance of MaxIS, and let $h > |V| - 2$ be the size of independent set $|V'|$. We can compute in polynomial time (by removing every possible set consisting of at most 2 vertices) whether there exists an independent set of size h. □

Consider an instance $\mathcal{G}$ of MaxIS. Then, we will show how to construct (in polynomial time) a corresponding instance (G, T) of MinLeafMod. First, we introduce the leafset Λ that labels the leaves of the two trees:

$$\Lambda = \{x_i, \alpha_i : 0 \leq i \leq |V|\} \cup \{y_i : 0 \leq i \leq h + 1\} \cup \{\beta\}.$$

Now, we describe the two trees (see Fig. 4). Similarly to the previous reduction, the *spine* of G is the unique path that connects the root of G to the internal node of G denoted as r_0, while the *spine* of T, is the unique path that connects the root of T to the unique leaf of T labeled by y_0.

The species tree T is a "caterpillar" over leafset Λ. More precisely, G is built by connecting the following subtrees to the spine of G (starting from the farthest from the root):

- a subtree $G(x_i)$, with $0 \leq i \leq |V|$; $G(x_0)$ is a "caterpillar" over three leaves labeled by α_0, x_0, and x_0, respectively; $G(x_i)$, with $1 \leq i \leq |V|$, is a "caterpillar" over three leaves labeled by x_i, α_{i-1} and α_i, respectively. The nodes of the spine connected to $G(x_i)$, with $0 \leq i \leq |V|$ are denoted as r_i;

- $h+3$ leaves each one labeled by y_i, for each i, with $0 \leq i \leq h+1$;
- a leaf labeled by β;
- for each edge $\{v_i, v_j\} \in E$, a subtree $G_{i,j}$ having two leaves labeled by x_i and x_j, respectively.

First, we state a property of the instance (G, T).

Remark 4. *Consider the instance (G, T) of MinLeafMod associated with an instance of MaxIS. Then, each node that connects the farthest leaf from the root labeled by y_i, with $0 \leq i \leq h+1$, to the spine of the gene tree G is a NAD node.*

We call a leaf modification *useless* if it does not change the label of a leaf into a label y_i, with $0 \leq i \leq h+1$. Next, we show that if there exists a solution of MinLeafMod with at most $h+2$ leaf modifications, then there exists a solution with at most $h+2$ leaf modification obtained without useless modifications.

Lemma 11. *Consider a solution G^* in which at most $h+2$ leaves are modified. Then: (1) none of the leaves labeled by y_i, with $0 \leq i \leq h+1$, is modified and (2) G^* is obtained modifying the labels of $h+2$ leaves of $G[r_{|V|}]$ and each of these leaves is assigned a distinct label in $\{y_0, \ldots, y_{h+1}\}$.*

Proof. (1) First notice that, since G contains $h+3$ leaves each one labeled by y_i, with $0 \leq i \leq h+1$, then there exists at least one leaf of G labeled by y_i which is not modified. Now, consider a solution G^* in which some of the leaves labeled by y_i are modified and let w be the leaf farthest from the root of G^* having a label y_i and that does not belong to $G[r_{|V|}]$. Let x be the the parent of w. Notice that x cannot be a speciation, since by construction this will imply the modification of all the leaves labeled by α_j, with $0 \leq j \leq |V|$ and, by Remark 3, $|V| > h+2$. Hence, x must be an AD node. This implies that there exists a node labeled by y_i in $G[r_{|V|}]$. As a consequence, we can assume that no leaf with label in $\{y_i : 0 \leq i \leq h+1\}$ is modified, otherwise we can compute a solution G' in which less leaves are changed, by not modifying any leaf with a label in $\{y_i : 0 \leq i \leq h+1\}$. Hence, $h+2$ leaves must be modified in $G[r_{|V|}]$, it follows that no leaf of $\{y_i : 0 \leq i \leq h+1\}$ is modified in G^*.

(2) We have shown that each leaf with a label in $\{y_i : 0 \leq i \leq h+1\}$ is not modified. By Remark 4, each node that connects the first leaf labeled by y_i, with $0 \leq i \leq h+1$, is a NAD node. It follows that in an consistent tree G^* this node must be AD node. Then, in $G[r_{|V|}]$ there are exactly $h+2$ leaves that are modified, and each of these leaves is assigned a distinct labels of $\{y_0, \ldots, y_{h+1}\}$.

□

Lemma 12. *Consider a solution in which at most $h+2$ leaves are modified. Then none of the leaves labeled by α_i is modified.*

Proof. By Lemma 11, we can assume that exactly $h+2$ leaves are modified, changing each of their labels to a distinct label of $\{y_0, \ldots, y_{h+1}\}$. Assume that a leaf labeled α_i is modified in $G^*(x_i)$ or in $G^*(x_{i+1})$. In both cases, since the only modifications possible are to distinct labels in $\{y_0, \ldots, y_{h+1}\}$, then the node r_{i+1} would be a NAD node.

□

Now, we can present the main results of this section.

Lemma 13. *Given an independent set of $\mathcal{G}$ size h, we can compute in polynomial time a solution of MinLeafMod over instance (G, T) in which exactly $h+2$ leaves are modified.*

Proof. Consider an independent set I of $\mathcal{G}$ of size at least h. Choose the first h vertices, $v_{i_1}, \ldots v_{i_h} \in I$ and compute a solution G^* as follows: modify the node labeled by x_{i_j} of $G(x_{i_j})$ by assigning the label y_{i_j}. Moreover, modify the nodes labeled by x_0 of $G(x_0)$ by assigning labels y_0 and y_{h+1}.

By construction, G^* is consistent with T. Indeed, by construction each subtree $G(x_i)$, with $0 \leq i \leq h+1$, is consistent with T and the nodes of G^* corresponding to r_i, with $0 \leq i \leq h+1$, are all AD nodes due to the leaves labeled by α_i. By construction, all the nodes with children y_i, with $0 \leq i \leq h+1$, are AD node.

Moreover, notice that, since the set $\{v_{i_1}, \dots v_{i_h}\}$ is an independent set of $\mathcal{G}$, for each subtree $G_{i,j}$ at least one of the labels in $\{x_i, x_j\}$ belongs to $G^*_{i,j}$ and to $G^*[r_{|V|}]$, implying that these nodes are all AD nodes.

□

Lemma 14. *Given a solution of MinLeafMod over instance (G, T) in which exactly $h+2$ leaves are modified, we can compute in polynomial time an independent set of $\mathcal{G}$ consisting of at least h vertices.*

Proof. Consider a solution G^* of MinLeafMod in which exactly $h+2$ leaves are modified. By Lemma 11, it follows that the solution must modify exactly $h+2$ leaves of $G[r_{|V|}]$. Then, for each tree $G_{i,j}$ having leaves labeled by x_i, x_j, at least one of the leaves in $G[r_{|V|}]$ having those labels, is not modified. If this is not the case, the node on the spine of G^* connected to the root of $G_{i,j}$ is a NAD node. Moreover, by Lemma 11 and by Lemma 12, it follows that the modified leaves of $G^*[r_{|V|}]$ are associated with labels in $\{x_0, \dots, x_{|V|}\}$. Hence, define an independent set of $\mathcal{G}$ as follows: $V' = \{v_i \in V : x_i$ is a label associated with a modified leaf of $G^*[r_{|V|}]\}$.
Then, since for each $G_{i,j}$, with leaves labeled by x_i, x_j, at least one of the leaves in $G[r_{|V|}]$ is labeled by x_i, x_j, it follows that for each $v_i, v_j \in V'$, it holds that $\{v_i, v_j\} \notin E$. Then, it follows that V' is an independent set of $\mathcal{G}$ of size h.

□

As a consequence of Lemma 13, of Lemma 14, and of the $W[1]$-hardness of MaxIS [28], we have the following result.

Theorem 3. *MinLeafMod is W[1]-hard when parameterized by the number of leaf modifications.*

6. W[2]-hardness and Inapproximability of MinLabelMod

In this section, we investigate the parameterized and approximation complexity of the MinLabelMod problem and we show that it is W[2]-hard and not approximable within factor $c \log n$, for some constant $c > 0$, by giving a (parameterized and approximation preserving) reduction from the Minimum Set Cover (MinSC) problem (for a definition of MinSC see Section 3).

In the following, given an instance $(\mathcal{C} = \{S_1, \dots, S_p\}, U = \{u_1, \dots, u_q\})$ of MinSC, we show how to construct in polynomial time an instance (G, T) of MinLabelMod. First, we introduce the leafset Λ that labels the gene tree G and the species tree T:

$$\Lambda = \{c_i, \alpha_i : S_i \in \mathcal{C}\} \cup \{u_{i,1}, u_{i,2} : u_i \in U\}.$$

Now, we describe the two trees G and T (see Fig. 5). The species tree T is obtained by inserting in the spine (starting from the root) a set of subtrees, each one having leafset labeled by $\{u_{i,1}, u_{i,2}\}$, with $1 \leq i \leq q$, then a set of subtrees each one having leafset labeled by $\{c_i, \alpha_i\}$, with $1 \leq i \leq p$.

The gene tree G is obtained by inserting in the spine (starting from the root) the following subtrees:

- a subtree $G(u_i)$, one for each $u_i \in U$, where $G(u_i)$ contains a left subtree which is a "caterpillar" with leaf uniquely labeled by $c_{i,1}, \dots, c_{i,d(u_i)}$ (being $S_{i,1}, \dots, S_{i,d(u_i)}$ the sets of $\mathcal{C}$ that contain u_i), a right subtree which contains a subtree with leaves uniquely labeled by $\{\alpha_1, \dots, \alpha_p, u_{i,1}, u_{i,2}\}$, with $1 \leq i \leq q$.
- a set of p leaves labeled by α_i, with $1 \leq i \leq p$.

Next, we show that the instance (G, T) has the following property: (1) no label $u_{i,x}$, with $1 \leq i \leq q$ and $x \in \{1, 2\}$, is modified (see Lemma 15), (2) no label α_i, with $1 \leq i \leq p$, is modified, and (3) each modified label c_i, with $1 \leq i \leq p$, is modified into a label α_i (see Lemma 16).

First, we illustrate a property of the instance (G, T).

Remark 5. *The NAD nodes of G are exactly the roots of the subtrees $G(u_i)$, with $1 \leq i \leq q$.*

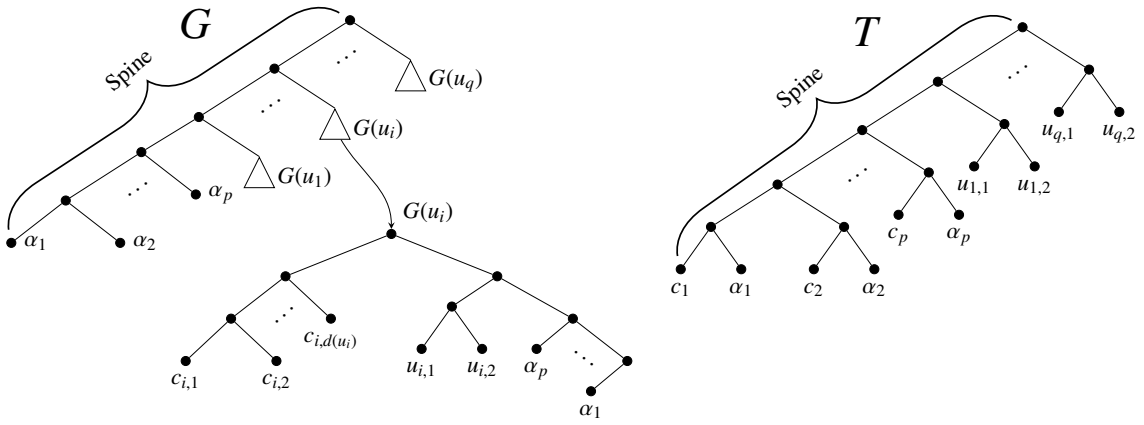

Figure 5: The gene tree G with its subtree $G(u_i)$ and the species tree T.

As a consequence of Remark 5, we have that for each subtree $G(u_i)$, there exists at least one label in $\Lambda(G(u_i))$ that must be modified in each feasible solution of MinLabelMod.

Now, we prove that modifying a label $u_{i,x}$ is essentially useless.

Lemma 15. *Let G' be a solution of MinLabelMod obtained with h label modifications such that label $u_{i,x}$, with $1 \leq i \leq q$ and $x \in \{1,2\}$, is modified. Then, starting from G' we can compute in polynomial time a solution G^* with less than h modifications, such that in G^* no label $u_{i,x}$, with $1 \leq i \leq q$ and $x \in \{1,2\}$, is modified.*

Proof. First notice that label $u_{i,x}$ is associated with a single leaf of G, namely of $G(u_i)$. Hence, we have that an eventual modification affects only subtree $G(u_i)$ and, eventually, nodes on the spine of G (that are already AD nodes due to labels α_i). Now, if labels $u_{i,x}$ and c_j (which is associated with a leaf of $G'(u_i)$) are modified in G', then solution G^* is obtained from G' by modifying c_j into α_j and by not changing label $u_{i,x}$. The resulting subtree $G^*(u_i)$ does not contain NAD nodes.

Now, if a label $u_{i,x}$ is modified in G', and no label c_j associated with a leaf of $G(u_i)$ is modified in G', then solution G^* is obtained from G' as follows: modify c_j into α_j and do not change label $u_{i,x}$. Again, the resulting subtree $G^*(u_i)$ does not contain NAD nodes.

□

Now, we show that, if we modify a label c_i, then this label must be modified into label α_i.

Lemma 16. *Let G' be a solution of MinLabelMod over instance (G,T) obtained with h label modifications such that label c_i, with $1 \leq i \leq p$, is modified. Then starting from G' we can compute in polynomial time a solution G^* with at most h label modifications such that G^* is obtained by modifying label c_i into α_i, with $1 \leq i \leq p$.*

Proof. First notice that, by Lemma 15, we can conclude that no label $u_{i,x}$, with $1 \leq i \leq q$ and $x \in \{1,2\}$, is modified. Hence, by construction (see Remark 5), we can conclude that in each subtree $G(u_j)$, with $1 \leq j \leq q$, either a label c_i or a label α_i associated with a leaf of $G(u_j)$ must be modified. Notice that, by construction, in each subtree $G(u_j)$ containing a leaf labeled by c_i, with $1 \leq i \leq p$, there exists a leaf labeled by α_i. Consider a label c_i that has has been modified in a solution G'. We compute G^* by modifying c_i into α_i (and hence in G^* label α_i is not modified). Then the root of each subtree $G^*(u_j)$ corresponding to $G(u_j)$, with $1 \leq j \leq q$, is an AD node. Since no other subtree of G (not containing a leaf labeled by c_i) is affected by the modification of c_i and α_i and since G' is obtained with h modifications, we can conclude that G^* is a solution obtained with at most h modifications.

□

Now, we are ready to prove the two main results of this section.

Lemma 17. *Let $(\mathcal{C}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLabelMod. Then, starting from a cover $\mathcal{C}'$ of U we can compute in polynomial time a solution of MinLabelMod over instance (G, T) in which $|\mathcal{C}'|$ labels are modified.*

Proof. The solution G^* is constructed in polynomial time, starting from a cover $\mathcal{C}'$, as follows: for each C_i in $\mathcal{C}'$, G^* is obtained by modifying label c_i into α_i.

Obviously, G^* is obtained by modifying $|\mathcal{C}'|$ labels. Next, we show that G^* is consistent with T. Consider a subtree $G^*(u_i)$. Each of these subtrees is obtained by modifying some labels, since $\mathcal{C}'$ is a cover of U_i. By construction, the internal nodes in the left subtree of $G^*(u_i)$, where there are leaves with modified labels, are all speciation nodes. Moreover, no label associated with a leaf of the right subtree of $G(u_i)$ has been modified in $G^*(u_i)$, hence by construction of G and by Remark 5, $G^*(u_i)$ contains only speciation nodes. Finally, the root of each $G^*(u_i)$ is an AD node, since there exists a label α_j that labels a leaf of the left and the right subtrees of $G^*(u_i)$, being $\mathcal{C}'$ a cover of U. Since the nodes on the spine of G^* (by construction) are either speciation nodes or AD nodes, it follows that G^* is a feasible solution of MinLabelMod over instance (G, T) in which at most h labels are modified, thus concluding the proof.

□

Lemma 18. *Let $(\mathcal{C}, U)$ be an instance of MinSC and let (G, T) be the corresponding instance of MinLabelMod. Then, given a solution of MinLabelMod obtained by modifying h labels, we can compute in polynomial time a cover $\mathcal{C}'$ of U consisting of h sets.*

Proof. Consider a solution G^* of MinLabelMod over instance (G, T). By Lemma 15 we can conclude that no label $u_{i,x}$, with $1 \leq i \leq q$ and $x \in \{1, 2\}$, is modified. It follows that only labels c_i and α_i, with $1 \leq i \leq p$, can be modified. Assume that the latter condition holds. Then, we can compute in polynomial time a solution in which at least the same number of leaves of G^* are modified, by substituting, for each α_i modified in G^*, the label c_i with α_i, leaving α_i unchanged. Hence, we can assume that in G^* only labels c_i, with $1 \leq i \leq p$, are modified and by Lemma 16, we can assume that in G^* the modified label c_i is substituted with α_i. Now, we can define a set cover a follows:

$$\mathcal{C}' = \{S_i : \text{label } c_i \text{ is modified into } \alpha_i\}.$$

$\mathcal{C}'$ is indeed a set cover. Assume that u_i is not covered by a set of $\mathcal{C}'$, then no leaf of $G^*(u_i)$ has a modified label, hence, by Remark 5 the root of $G^*(u_i)$ is a NAD node.

□

The inapproximability of MinLabelMod follows from Lemma 17 and Lemma 18 and from the inapproximability [26]. The W[2]-hardness of MinLabelMod follows from Lemma 17 and Lemma 18 and from the W[2]-hardness of MinSC [29].

Theorem 4. *MinLabelMod is not approximable within factor $c \log |\Lambda|$, for some constant $c > 0$, and is W[2]-hard when parameterized by the number of label modifications.*

Proof. The proof follows from Lemma 17 and Lemma 18 and from the fact that the described reduction is a parameterized reduction and an approximation preserving reduction. Hence, since MinSC is W[2]-hard [29], also MinLabelMod is W[2]-hard. Moreover, notice that MinSC is not approximable within factor $c \log q$ [26], for some constant $c > 0$, and the same property holds for MinLabelMod. Since $|\Lambda| = 2p + 2q$, and MinSC is not approximable within factor $c \log q$, for some constant $c > 0$, even when p and q are related by a polynomial [27], it follows that MinLabelMod is not approximable within factor $c \log |\Lambda|$, for each constant $c > 0$.

□

7. Conclusion

In this paper, we studied the approximation and parameterized complexity of some combinatorial problems related to gene tree correction. For MinLeafRem, MinLabelRem and MinLabelMod we showed that the problems are not approximable within factor $b \log m$, where m is the number of leaves of the species tree and $b > 0$ is a constant. Moreover, we showed that the modification problems, differently from the removal versions, are unlikely to be fixed-parameter tractable. More precisely, we showed that MinLeafMod is $W[1]$-hard, when parameterized by the number of leaf modifications, and MinLabMod is $W[2]$-hard, when parameterized by the number of label modifications.

There are some interesting future directions related to the approximation complexity of these problems. The first natural question is whether MinLeafMod admits a constant factor approximation or not. Another natural question is whether it is possible to have a polylog factor approximation algorithm for the problems we considered.

Acknowledgements

We would like to thanks the anonymous referees for their valuable comments and suggestions.

S.B. is supported by the Italian Ministry of Education and Research (MIUR) through the "Flagship InterOmics" (code PB05), "HIRMA" (code RBAP11YS7K), and the European "MIMOmics" (code 305280) projects.

R.D. is partially supported by the Italian Ministry of Education and Research (MIUR) through PRIN 2010-2011 grant "Automi e Linguaggi Formali: Aspetti Matematici e Applicativi" (code H41J12000190001).

References

[1] S. Beretta, R. Dondi, Gene Tree Correction by Leaf Removal and Modification: Tractability and Approximability, in: A. Beckmann, E. Csuhaj-Varjú, K. Meer (Eds.), CiE, vol. 8493 of *Lecture Notes in Computer Science*, Springer, 42–52, 2014.

[2] S. Ohno, Evolution by gene duplication, Springer, Berlin, 1970.

[3] E. Eichler, D. Sankoff, Structural dynamics of eukaryotic chromosome evolution, Science 301 (2003) 793–797.

[4] P. Bonizzoni, G. Della Vedova, R. Dondi, Reconciling a gene tree to a species tree under the duplication cost model., Theoretical Computer Science 347 (2005) 36–53.

[5] W. Chang, O. Eulenstein, Reconciling gene trees with apparent polytomies, in: D. Chen, D. T. Lee (Eds.), COCOON 2006, vol. 4112 of *LNCS*, Heidelberg, 235–244, 2006.

[6] C. Chauve, N. El-Mabrouk, New perspectives on gene family evolution: losses in reconciliation and a link with supertrees, in: S. Batzoglou (Ed.), RECOMB 2009, vol. 5541 of *LNCS*, Springer, Heidelberg, 46–58, 2009.

[7] D. Durand, B. Haldórsson, B. Vernot, A hybrid micro-macroevolutionary approach to gene tree reconstruction, Journal of Computational Biology 13 (2006) 320–335.

[8] R. Page, GeneTree: comparing gene and species phylogenies using reconciled trees., Bioinformatics 14 (1998) 819–820.

[9] R. Page, J. Cotton, Vertebrate phylogenomics: reconciled trees and gene duplications, in: Pacific Symposium on Biocomputing, 536–547, 2002.

[10] M. Sanderson, M. McMahon, Inferring angiosperm phylogeny from EST data with widespread gene duplication., BMC Evolutionary Biology 7 (2007) S3.

[11] B. Vernot, M. Stolzer, A. Goldman, D. Durand, Reconciliation with non-binary species trees, Journal of Computational Biology 15 (2008) 981–1006.

[12] J.-P. Doyon, C. Scornavacca, K. Gorbunov, G. Szöllősi, V. Ranwez, V. Berry, An Efficient Algorithm for Gene/Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers, in: E. Tannier (Ed.), Comparative Genomics, vol. 6398 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 93–108, 2010.

[13] M. Hallett, J. Lagergren, A. Tofigh, Simultaneous identification of duplications and lateral transfers, in: RECOMB, ACM, 2004.

[14] A. Tofigh, M. Hallett, J. Lagergren, Simultaneous identification of duplications and lateral gene transfers, IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (2011) 517–535.

[15] M. S. Bansal, E. J. Alm, M. Kellis, Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss, Bioinformatics 28 (12) (2012) 283–291.

[16] G. Blin, P. Bonizzoni, R. Dondi, R. Rizzi, F. Sikora, Complexity insights of the Minimum Duplication problem, Theor. Comput. Sci. 530 (2014) 66–79.

[17] B. Ma, M. Li, L. Zhang, From gene trees to species trees, SIAM J. on Comput. 30 (2000) 729–752.

[18] M. Hahn, Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution, Genome Biology 8 (R141).

[19] K. Chen, D. Durand, M. Farach-Colton, Notung: Dating Gene Duplications using Gene Family Trees, Journal of Computational Biology 7 (2000) 429–447.
[20] P. Górecki, O. Eulenstein, A linear time algorithm for error-corrected reconciliation of unrooted gene trees, in: J. Chen, J. Wang, A. Zelikovsky (Eds.), ISBRA 2012, vol. 6674 of *LNCS*, Springer, Heidelberg, 148- 159, 2011.
[21] M. Lafond, C. Chauve, R. Dondi, N. El-Mabrouk, Polytomy Refinement for the Correction of Dubious Duplications in Gene Trees, Bioinformatics 30 (17) (2014) to appear.
[22] K. M. Swenson, A. Doroftei, N. El-Mabrouk, Gene tree correction for reconciliation and species tree inference, Algorithms for Molecular Biology 7 (2012) 31.
[23] R. Dondi, N. El-Mabrouk, K. M. Swenson, Gene Tree Correction for Reconciliation and Species Tree Inference: Complexity and Algorithms, Journal of Discrete Algorithms 25 (2014) 51–65.
[24] R. G. Downey, M. R. Fellows, Parameterized Complexity, Monographs in Computer Science, Springer, 1999.
[25] V. V. Vazirani, Approximation algorithms, Springer, 2001.
[26] R. Raz, S. Safra, A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP, in: F. T. Leighton, P. W. Shor (Eds.), STOC, ACM, 475–484, 1997.
[27] J. Nelson, A Note on Set Cover Inapproximability Independent of Universe Size, Electronic Colloquium on Computational Complexity (ECCC) 14 (105).
[28] R. G. Downey, M. R. Fellows, Fixed-Parameter Tractability and Completeness II: On Completeness for W[1], Theor. Comput. Sci. 141 (1&2) (1995) 109–131.
[29] A. Paz, S. Moran, Non Deterministic Polynomial Optimization Problems and their Approximations, Theor. Comput. Sci. 15 (1981) 251–277.