

Minimum Leaf Removal for Reconciliation: Complexity and Algorithms

Riccardo Dondi¹ and Nadia El-Mabrouk²

¹ Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali,
Università degli Studi di Bergamo, Bergamo - Italy

² Département d'Informatique et Recherche Opérationnelle, Université de Montréal,
Montréal - Canada

riccardo.dondi@unibg.it, mabrouk@iro.umontreal.ca

Abstract. Reconciliation is a well-known method for studying the evolution of a gene family through speciation, duplication, and loss. Unfortunately, the inferred history strongly depends on the considered gene tree for the gene family, as a few misplaced leaves can lead to a completely different history, possibly with significantly more duplications and losses. It is therefore essential to develop methods that are able to preprocess and correct gene trees prior to reconciliation. In this paper, we consider a combinatorial problem, known as the **Minimum Leaf Removal** problem, that has been proposed to remove errors from a gene tree by deleting some of its leaves. We prove that the problem is APX-hard, even in the restricted case of a gene family with at most two copies per genome. On the positive side, we present fixed-parameter algorithms where the parameters are the size of the solution (minimum number of leaf removals) and the number of genomes containing multiple gene copies.

1 Introduction

The evolution of genomes is determined by a combination of micro-evolutionary events affecting their sequences, and macro-evolutionary events, involving rearrangement and content-modifying operations, affecting their overall gene content and organization. Among content-modifying operations, duplication is a fundamental process in the evolution of species, and a major source of gene innovation [24,14]. The consequence of duplications is that genes are not present in one, but in many copies, in the genome. In parallel to duplications, gene losses appear generally to maintain a minimum number of functional gene copies [5,10,11,20]. Using a local similarity search tool such as BLAST [2], genes can be clustered by sequence homology into *gene families*. From a conceptual evolutionary point of view, homologous gene copies originate from the same ancestral gene.

Understanding the evolution of gene families through duplication and loss is fundamental for many reasons. In particular, it allows distinguishing between two classes of gene homologs [21]: *orthologs* which are copies in different species that arose by speciation at their most recent point of origin, and *paralogs* which are gene copies in the same genome or in two different genomes that arose

from a duplication at their most recent point of origin. While orthologs are, in essence, instances of the ‘same gene’ in different species, paralogs represent different copies of the ancestor that are likely to have independently evolved and diverged in their function. Consequently, identifying the “true” orthology relationship between genes is fundamental for functional annotation of genes, as well as phylogenetic inference and comparative genomics purposes.

Based on a micro-evolutionary model for sequences, a gene tree T that best explains the data can be constructed for a given gene family, by using a classical phylogenetic method. When a species tree S reflecting the speciation history of the genomes is known, then the macro-evolutionary events that gave rise to the data can be inferred by using a method known as *Reconciliation*. It consists in “embedding” T into S , and interpreting the disagreement between the two trees as a footprint of the evolution of the gene family through duplication and loss. This concept was pioneered by Goodman [15] and then widely accepted, utilized, and improved [3,6,7,8,10,13,26,28,29,30]. When no preliminary knowledge on the species tree is given, a natural problem, known as the *species tree inference problem*, is to infer, from a set of gene trees, a species tree leading to a parsimonious evolution scenario [4,8,22].

A major problem in the application of gene tree reconciliation is its high sensitivity to error-prone gene trees. Indeed, a few misplaced leaves can lead to a completely different history, possibly with significantly more duplications and losses [19,29]. Typically bootstrapping values are used as a measure of confidence in each edge of a phylogeny. How should the weak edges of a gene tree be handled? This problem has been addressed in [9,13,16] by exploring the space of gene trees obtained from the original one by performing rearrangements (such as NNIs) around weakly-supported edges and select the tree giving rise to the minimum duplications and losses. A different strategy that has been recently adopted for preprocessing a gene tree T prior to reconciliation or species tree inference, is to “remove” misplaced leaves (gene copies). Criteria for identifying such leaves were given in [8]. The duplication nodes of T with respect to a species tree S can be subdivided into apparent and non-apparent duplication (NAD) nodes, where the latter class has been flagged as potentially resulting from the misplacement of leaves in the gene tree. The reason is that each one of the NAD nodes reflects a phylogenetic contradiction with the species tree that is not due to the presence of duplicated gene copies. In [12], algorithmic results have been presented for the problem of removing, from a given gene tree, the minimum number of leaves leading to a tree without any NAD node (the **Minimum Leaf Removal Problem**). An exact polynomial-time algorithm has been described for two special classes of gene trees, and a polynomial-time heuristic with no guarantee of optimality, has been presented for the general case.

In this paper, we study the theoretical complexity of the **Minimum Leaf Removal Problem**. More precisely, we show in Section 3 that the problem is APX-hard, by reduction from the Minimum Vertex Cover problem on Cubic graph [1]. We then turn our attention in Section 4 to finding tractable versions of the problem under some biological meaningful parameterizations. The goal is to identify

parameters that are small in practice, and to constraint the exponential explosion only to these parameters. We identify two fixed-parameter tractable versions of the problem and present exact polynomial-time algorithms constrained by: (1) the size of the solution (minimum number of leaf removal) and (2) the number of genomes containing multiple gene copies (paralogs). We begin in the next section by introducing the concepts and notations used in the rest of the paper. Due to space limitations some of the proofs are omitted.

2 Preliminary Definitions

2.1 Trees

Let $\Gamma = \{1, 2, \dots, \gamma\}$ be a set of integers representing γ different species (genomes). We consider two kinds of rooted binary trees leaf-labelled by the elements of Γ : a *species tree* S is a tree where each element of Γ labels at most one leaf, while a *gene tree* T is a tree where each element of Γ may label more than one leaf (Figure 1 (a) and (b)). A gene tree represents a gene family, where each leaf labelled x represents a gene copy located on genome x .

Given a tree U , we denote by $L(U)$ the set of its leaves and by $V(U)$ the set of its nodes. Given an internal node x of U , we denote by x_l and x_r respectively, the left and right child of x , by $U(x)$ the subtree of U rooted at x , and by $\Gamma(U(x))$ the set of leaf-labels of $U(x)$. If there is no ambiguity on the tree being considered, we denote $\mathcal{C}(x) = \Gamma(U(x))$; $\mathcal{C}(x)$ is called the *cluster* of x . An *ancestor* of a node x of U is any node on the path from the root of U to x .

Given a tree U , a *leaf removal* consists in removing a given leaf l of U , and suppressing the resulting degree two node (that is the parent of l). If a tree U' is obtained from a tree U through a sequence of leaf removals, then U' is *included* in U . On the other hand a *subtree insertion* in U consists in creating a new node x on a branch (a, b) (joining node a to node b , b being the child of a), making b the left child of x , setting the parent of x to a , and grafting the subtree being inserted as the second child of x (create an edge from x to the root of the subtree). An *extension* of U is a tree obtained from U through a sequence of subtree insertions.

2.2 Reconciliation

Usually, the gene tree T obtained for a given gene family is different from the species tree S . Roughly speaking, a *reconciliation* between T and S is an extension $R(T, S)$ of T that is “consistent” with S , i.e. reflects the same phylogeny. A rigorous definition can be found in [8,12]. A history of duplications and losses can immediately be inferred from such a reconciliation. Different algorithms have been developed for recovering a reconciliation minimizing a duplication and/or loss cost [6,13,17,18,22,25,27,8], most of them based on a method called *LCA mapping*.

The LCA mapping between a gene tree T and a species tree S , denoted by $\text{lca}_{T,S}$, maps every node x of T to the Lowest Common Ancestor (LCA) of $\mathcal{C}(x)$

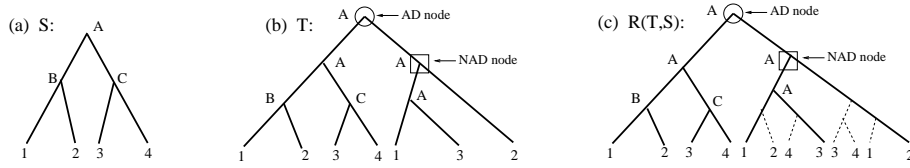


Fig. 1. (a) A species tree S for $\Gamma = \{1, 2, 3, 4\}$. The three internal nodes of S are named A , B and C ; (b) A gene tree T . A leaf label g indicates a gene copy in genome g . Internal nodes are labelled according to the LCA mapping between T and S . Flagged nodes are duplication nodes of T with respect to S ; (c) A reconciliation $R(T, S)$ of T and S . Dotted lines represent subtree insertions. This reconciliation reflects a history of the gene family with two gene duplications preceding the first speciation event, and 4 losses.

in S . Formally, $\text{lca}_{T,S}(x) = y$, where y is the node of S that has the minimum cluster such that $\mathcal{C}(x) \subseteq \mathcal{C}(y)$. A *duplication* occurs in a node x of T (or x is a duplication), if x and at least one of its children are mapped by $\text{lca}_{T,S}$ in the same node y of the species tree S . If x is not a duplication node, then x is a *speciation* (Figure 1).

2.3 Duplication Nodes and MD-trees

The notations of this section are those used in [8,12]. Let x be a node of a gene tree T verifying $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) \neq \emptyset$. Then, for any species tree S , x is guaranteed to be a duplication node. Such a node x is called an *Apparent Duplication node* (*AD node* for short). Given a species tree S , a duplication node x which is not an AD node is called a *Non-Apparent Duplication node* (*NAD node* for short). A gene tree T is *MD-consistent* (MD holds for “Minimum Duplication”) with a species tree S if and only if each node of T is either a speciation or an AD node.

As explained in [12], NAD nodes point to disagreement between a gene tree T and a species tree S that are not due to the presence of repeated leaf labels, i.e. duplicated gene copies (see Figure 1.(b)). It has therefore been suggested, and supported by simulations in [8], that NAD nodes may point at gene copies that are erroneously placed in the gene tree. It has to be noticed that a misplaced gene in a gene tree T does not necessarily lead to a NAD node. In other words, NAD nodes can only point to a subset of misplaced leaves. However, in the context of reconciliation, the damage caused by a misplaced leaf leading to a NAD node is to significantly increase the real duplication and/or loss cost of the tree. Following these observations, the *Minimum Leaf Removal Problem*, given below, has been considered in [12] for error-correction in gene trees.

Problem 1 *Minimum Leaf Removal Problem*[MinLeafRem]

Input: A gene tree T and a species tree S , both leaf-labelled by Γ .

Output: A tree T^* MD-consistent with S such that T^* is obtained from T by a minimum number of leaf removals.

3 Hardness of Minimum Leaf Removal

In this section we consider the computational (and approximation) complexity of the MinLeafRem problem. We show that MinLeafRem is APX-hard, even in the restricted case that each label is associated with at most two leaves of T . We denote this restriction of the problem by MinLeafRem(2).

We prove that MinLeafRem(2) is APX-hard, by giving an L -reduction from the MINIMUM VERTEX COVER PROBLEM on Cubic graphs (MVCC is known to be APX-hard [1]).

Problem 2 *Minimum Vertex Cover Problem on Cubic graphs*[MVCC]

Input: A cubic graph $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ is the set of vertices and E the set of edges of G (in a cubic graph, each vertex has degree 3).

Output: A minimum cardinality set $V' \subseteq V$, such that for each edge $e_{i,j} = \{v_i, v_j\} \in E$, at least one of v_i, v_j belongs to V' .

Let $G = (V, E)$ be an instance of MVCC. We define an instance of MinLeafRem associated with G , consisting of a gene tree T and a species tree S , both leaf-labelled by Γ , defined as follows, where $t = 4|V| + |E| + 1$:

$$\Gamma = \{v_{i,l} : v_i \in V, 1 \leq l \leq 4\} \cup \{v_i^j : v_i \in V, \{v_i, v_j\} \in E\} \cup \{e_{i,j} : \{v_i, v_j\} \in E\} \cup \{z_i : 1 \leq i \leq t\} \cup \{\alpha\}.$$

We denote $Z = \{z_i : 1 \leq i \leq t\}$. Let U be a tree, which is either the gene tree T , the species tree S , or a tree included in T with a leaf labelled by α . We define the *spine* of U as the path from the root of U to the unique leaf of U labelled by α .

Next, we define an ordering on the edges E of G . Consider the edges $\{v_i, v_j\}$, with $i < j$, and $\{v_h, v_k\}$, with $h < k$, then $\{v_i, v_j\} < \{v_h, v_k\}$, iff $i \leq h$, and $j < k$ if $i = h$. Denote with $\{v_p, v_q\}$ the last edge in such ordering of E .

The gene tree T is defined as in Fig. 2. It contains the following kinds of subtrees: (1) a subtree T_{v_i} , for each vertex $v_i \in V$; (2) a subtree $T_{e_{i,j}}$ and a leaf $e_{i,j}$, for each edge $e_{i,j} = \{v_i, v_j\} \in E$; (3) a tree T_Z , which is a caterpillar tree of size t with leaves uniquely leaf-labelled by the set Z . Notice that the order in which the subtrees $T_{e_{i,j}}$ and the leaf $e_{i,j}$ appear in T , depends on the order of the corresponding edges of E .

The species tree S is defined in Fig. 3. It contains the three following kinds of subtrees: (1) a subtree S_{v_i} , for each vertex $v_i \in V$; (2) a single leaf labelled by $e_{i,j}$, for each edge $e_{i,j} = \{v_i, v_j\} \in E$; (3) a tree S_Z , which is a caterpillar tree of size t uniquely leaf-labelled by the set Z .

It is easy to see that S is a species tree uniquely leaf-labelled by Γ , and that T is a gene tree where each label in Γ is associated with at most two leaves of T . The following properties of T are directly deduced from the construction of T .

Remark 1 *The root of T_Z and all its ancestors are mapped (by the LCA mapping) to the root r of S . Consequently, all T_Z ancestors are duplication nodes.*

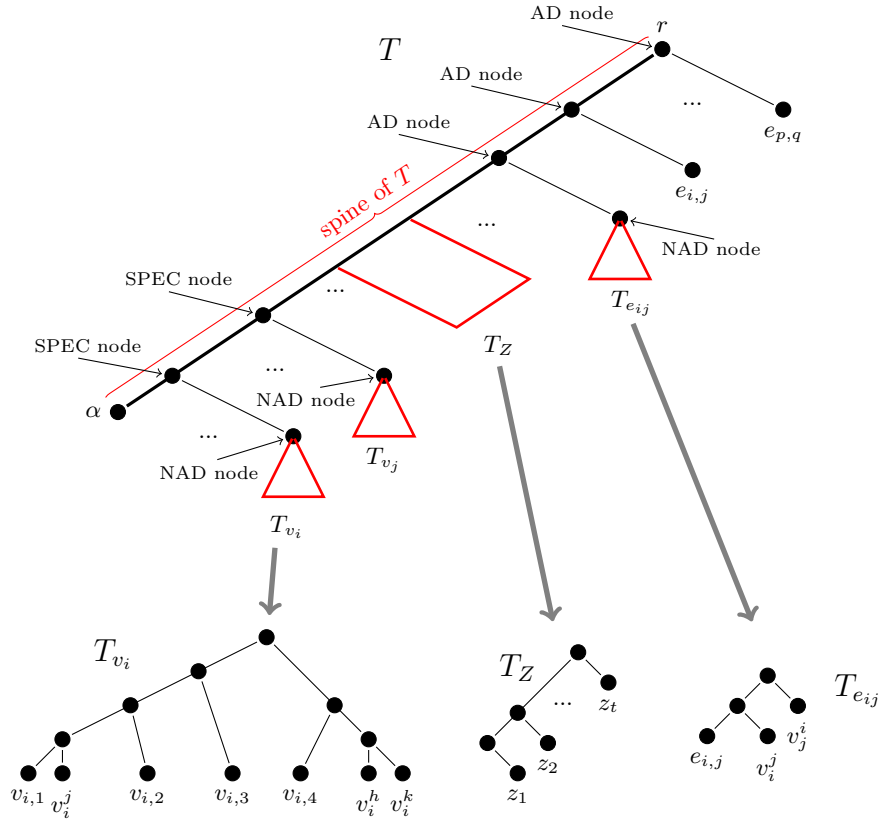


Fig. 2. The gene tree T , and the subtrees T_{v_i} , T_Z and $T_{e_{ij}}$ of T . Notice that $i < j$, hence T_{v_j} is closer to the root than T_{v_i} . Notice that a SPEC node is a speciation node.

Moreover, we deduce from the non-empty intersection of the left and right leaf sets that all these nodes are AD nodes.

Remark 2 For each $e_{i,j} \in E$, the root of $T_{e_{i,j}}$ is a NAD node. Indeed, it is mapped to the same node of S than its left child, and it does not contain any duplicated leaf-label.

Moreover, as each subtree T_{v_i} contains NAD nodes, any solution of Min-LeafRem over instance (T, S) is obtained by removing appropriate leaves from each T_{v_i} . The following results give more details on the required removals.

Remark 3 Let v_i be a vertex of G . Then: (1) the subtree of T_{v_i} obtained by removing the leaves with labels v_i^j, v_i^h, v_i^k is MD-consistent with S_{v_i} ; (2) the subtree of T_{v_i} obtained by removing the leaves with labels $v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}$ is MD-consistent with S_{v_i} .

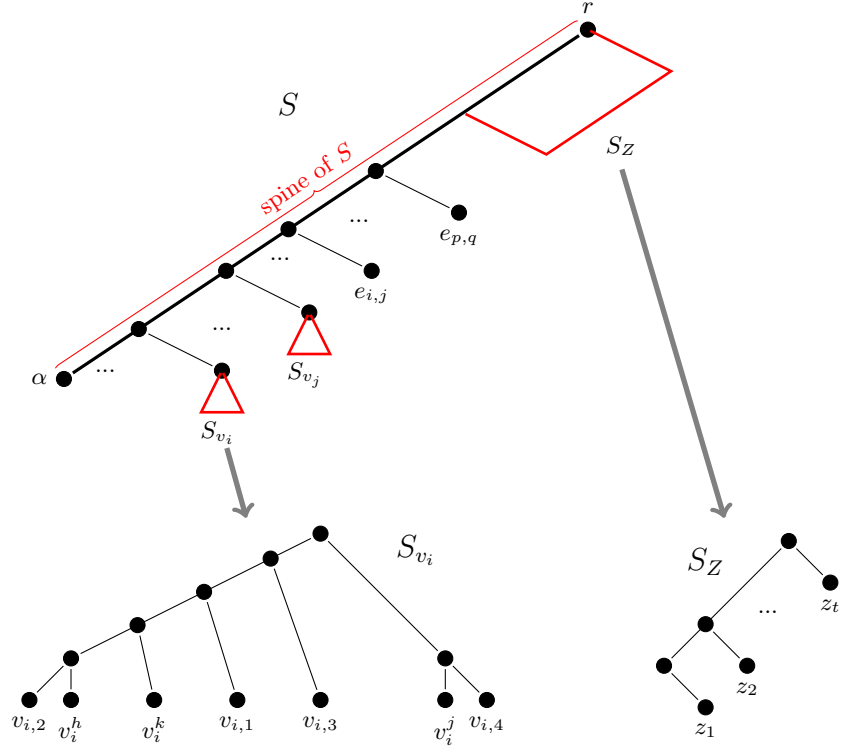


Fig. 3. The species tree S , and the subtrees S_{v_i} , S_Z of S . Notice that $i < j$, hence S_{v_j} is closer to the root than S_{v_i} .

Lemma 1 Let v_i be a vertex of G . Then: (1) in a solution of $\text{MinLeafRem}(2)$ over instance (T, S) at least three leaves are removed from T_{v_i} ; (2) a solution of $\text{MinLeafRem}(2)$ over instance (T, S) that contains a leaf of T_{v_i} with a label in $\{v_i^j, v_i^h, v_i^k\}$, contains at most three leaves of T_{v_i} .

It follows from Remark 3 and Lemma 1 that a solution of $\text{MinLeafRem}(2)$ over instance (T, S) is obtained by removing leaves from each T_{v_i} in essentially two possible ways: either remove the four leaves $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, or remove the three leaves $\{v_i^j, v_i^h, v_i^k\}$. We will relate the former case to the vertex v_i being included in a vertex cover V' of G , and the latter case to the vertex v_i not included in V' (Lemma 4 and Lemma 5). We first give two preliminary lemmas.

Lemma 2 Each solution of $\text{MinLeafRem}(2)$ over instance (T, S) is obtained by removing at least one leaf from $T_{e_{i,j}}$, for each $e_{i,j} \in E$.

Proof. Direct corollary of Remark 2.

The following lemma will be used to show that the caterpillar tree T_Z is kept in a solution of $\text{MinLeafRem}(2)$.

Lemma 3 *There is no optimal solution of $\text{MinLeafRem}(2)$ over instance (T, S) that is obtained by removing less than $4|V| + |E| + 1$ leaves, one of them being a leaf of T_Z .*

Proof. Let T^* be a solution of MinLeafRem over instance (T, S) obtained from T by removing less than $4|V| + |E| + 1$ leaves. Notice that, since $|Z| = 4|V| + |E| + 1$, at least one leaf with a label in the set Z must be in T^* . Assume that a leaf with label z_h is removed from T^* . It is easy to see that inserting this leaf in T^* does not affect other nodes of T^* , that is the insertion of the leaf with label z_h does not cause any AD node to become a NAD node.

We are now ready to show the two main technical results of the reduction.

Lemma 4 *Let $G = (V, E)$ be an instance of MVCC and let (T, S) be the corresponding instance of $\text{MinLeafRem}(2)$. Then, starting from a vertex cover V' of G , we can compute in polynomial time a solution of $\text{MinLeafRem}(2)$ over instance (T, S) that is obtained by removing $3|V| + |V'| + |E|$ leaves from T .*

Proof. (Sketch) Let $V' \subseteq V$ be a vertex cover of $G = (V, E)$. Then we define a solution T^* by removing some leaves of the subtrees of T . We will denote by $T_{v_i}^*$ the subtree of T^* obtained from T_{v_i} , and by $T_{e_{i,j}}^*$ the subtree of T^* obtained from $T_{e_{i,j}}$. The solution T^* is defined as follows:

- for each $v_i \in V'$, remove from the subtree T_{v_i} the set of leaves labelled by $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ (hence the subtree $T_{v_i}^*$ has its leaf-set labelled by $\{v_i^j, v_i^h, v_i^k\}$);
- for each $v_i \in V \setminus V'$, remove from the subtree T_{v_i} the set of leaves labelled by $\{v_i^j, v_i^h, v_i^k\}$ (hence the subtree $T_{v_i}^*$ has its leaf-set labelled by $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$);
- for each $\{v_i, v_j\} \in E$, if $v_i \in V'$, then remove from $T_{e_{i,j}}$ the leaf labelled by v_j^i (hence the subtree $T_{e_{i,j}}^*$ has its leaf-set labelled by $\{e_{i,j}, v_i^j\}$), else remove from $T_{e_{i,j}}$ the leaf labelled by v_i^j (hence the subtree $T_{e_{i,j}}^*$ has its leaf-set labelled by $\{e_{i,j}, v_j^i\}$).

It is easy to see that the tree T^* is MD-consistent with S and that it is obtained by removing $3|V| + |E| + |V'|$ leaves from T .

Lemma 5 *Let $G = (V, E)$ be an instance of MVCC and let (T, S) be the corresponding instance of $\text{MinLeafRem}(2)$. Then starting from a solution of $\text{MinLeafRem}(2)$ over instance (T, S) that is obtained by removing at most $3|V| + |E| + c$ leaves from T , with $1 \leq c \leq |V|$, we can compute in polynomial time a vertex cover V' of G such that $|V'| \leq c$.*

Proof. (Sketch) Let T^* be a solution of $\text{MinLeafRem}(2)$ over instance (T, S) obtained by removing at most $3|V| + |E| + c$ leaves from T , with $1 \leq c \leq |V|$. Let $T_{v_i}^*$, with $v_i \in V$, be the subtree of T^* obtained from T_{v_i} after removing

appropriate leaves. Let $T_{e_{i,j}}^*$, with $\{v_i, v_j\} \in E$, be the subtree of T^* obtained from $T_{e_{i,j}}$ after removing appropriate leaves.

We can show (using Remark 3 and Lemma 1) that $T_{v_i}^*$, for each $v_i \in V$, must be leaf-labelled either by the set $\{v_i^j, v_i^h, v_i^k\}$, or by the set $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$. Moreover, by Lemma 3, T^* contains all the leaves with labels in Z .

On the other hand, using Lemma 2, we can prove that $T_{e_{i,j}}^*$ must contain the leaf labelled by $e_{i,j}$ (otherwise the parent of the leaf labelled by $e_{i,j}$ on the spine of T , which is an AD node in T , becomes a NAD node) and exactly one leaf with label in $\{v_i^j, v_j^i\}$ (otherwise the parent of the subtree $T_{e_{i,j}}$ on the spine of T , which is an AD node in T , becomes a NAD node). Moreover, if $T_{e_{i,j}}^*$ contains a leaf labelled by v_i^j , then $T_{v_i}^*$ must be leaf-labelled by the set $\{v_i^j, v_i^h, v_i^k\}$ (otherwise the parent of the subtree $T_{e_{i,j}}$ on the spine of T becomes a NAD node), while if $T_{e_{i,j}}^*$ contains a leaf labelled by v_j^i , then T_{v_j} is leaf-labelled by the set $\{v_j^i, v_j^x, v_j^y\}$ (same reason as above).

It follows that the set

$$V' = \{v_i : T_{v_i}^* \text{ is leaf-labelled by } \{v_i^j, v_i^h, v_i^k\}\}$$

is a vertex cover of G of minimum size, as for each edge $e_{i,j} \in E$, exactly one of v_i and v_j is contained in V' . It is easy to see that $|V'| \leq c$.

Theorem 1 *MinLeafRem(2) is APX-hard.*

Proof. It follows from Lemma 4 and from Lemma 5, that we have designed an L-reduction from MVCC to MinLeafRem(2). Since MVCC is APX-hard [1], it follows that also MinLeafRem(2) is APX-hard.

4 Fixed-Parameter Algorithms

Since the MinLeafRem problem is APX-hard, it is interesting to see if the problem becomes tractable under some biological meaningful parameterizations (for an introduction to parameterized complexity see [23]). In this section we focus on the two following parameterizations: (1) the size of the solution of MinLeafRem (that is the number of leaves removed from T in order to obtain a tree MD-consistent with S), and (2) the number of labels in Γ associated with multiple leaves of T (i.e. the number of genomes containing multiple gene copies). We will give two fixed-parameter algorithms for MinLeafRem under these two parameterizations.

Notice that a third natural parameter would be the maximum number of leaves in T associated with a single label of Γ (i.e. the maximum number of gene copies in a given genome). However, we have already proved in the last section that the MinLeafRem problem is already APX-hard when each label has at most two occurrences in the gene tree T .

4.1 MinLeafRem Parameterized by the Number of Leaves Removed

In this section, we investigate the parameterized complexity of MinLeafRem, when the problem is parameterized by the size of the solution, that is the number of leaves removed from T . We present a fixed-parameter algorithm that is based on the depth-bounded search tree technique. Denote by c the size of the solution, that is the number of leaves that have to be removed from T in order to get a tree T^* which is MD-consistent with the species tree S .

If T does not contain NAD nodes, then T is MD-consistent with S and it requires no leaf removal. Hence in what follows we assume that T contains at least one NAD node.

Now, consider a NAD node v of T . Let s be the node of S where v is mapped. Since v is a NAD node, it follows that at least one of its children, denoted as v_l and v_r , is mapped by $\text{lca}_{T,S}$ in s . Assume w.l.o.g. that v_l is mapped in s , that is $\text{lca}_{T,S}(v_l) = s$. Denote by s_l and s_r the left child and the right child respectively of s . Since $\text{lca}_{T,S}(v_l) = s$, it follows that $\mathcal{C}(v_l) \subseteq \mathcal{C}(s)$, $\mathcal{C}(v_l) \cap \mathcal{C}(s_l) = X_1 \neq \emptyset$ and $\mathcal{C}(v_l) \cap \mathcal{C}(s_r) = X_2 \neq \emptyset$. It follows that either the leaves of $T(v_l)$ having labels in X_1 , or the leaves of $T(v_l)$ having labels in X_2 , or the leaves of $T(v_r)$ must be deleted from T . We formally prove this property in the following lemma.

Lemma 6 *Let v be a NAD node of a gene tree T , and let v_l, v_r be the children of v , such that $\text{lca}_{T,S}(v) = \text{lca}_{T,S}(v_l) = s$. Let s_l, s_r be the children of s . Then, there is no subtree included in T that is MD-consistent with S and that contains a leaf of $T(v_l)$ with a label in $X_1 = \mathcal{C}(v_l) \cap \mathcal{C}(s_l)$, a leaf of $T(v_l)$ with a label in $X_2 = \mathcal{C}(v_l) \cap \mathcal{C}(s_r)$, and a leaf of $T(v_r)$.*

Due to Lemma 6, we can design a fixed-parameter algorithm for MinLeafRem parameterized by c as follows. Let $\text{Dup}(T) = \langle v^1, \dots, v^z \rangle$ be the ordered list of NAD nodes of T in a breadth-first visit of T . The algorithm at each step chooses the first node v^1 of $\text{Dup}(T)$. Let $\text{lca}_{T,S}(v^1) = s$, and let s_l and s_r be the two children of s . Consider a child v_x^1 , with $v_x^1 \in \{v_l^1, v_r^1\}$, of v^1 that is mapped in s , and let $v_{\bar{x}}^1$ be the other child of v^1 . Let $\mathcal{C}(v_x^1) \cap \mathcal{C}(s_l) = X_1 \neq \emptyset$, $\mathcal{C}(v_x^1) \cap \mathcal{C}(s_r) = X_2 \neq \emptyset$.

Now, the algorithm branches in the following cases:

1. Remove the leaves of $T(v_x^1)$ with label in X_1 from $L(T)$ and suppress the resulting degree two nodes;
2. Remove the leaves of $T(v_x^1)$ with label in X_2 from $L(T)$ and suppress the resulting degree two nodes;
3. Remove the subtree $T(v_{\bar{x}}^1)$ from T , and suppress the resulting degree two node.

After the branching, the algorithm outputs a subtree T' of T . Then the lca mapping $\text{lca}_{T',S}$ between T' and S is computed (in polynomial time), and the ordered list $\text{Dup}(T')$ of NAD nodes of T' is computed (again in polynomial time). The algorithm stops either when it finds a subtree T' of T that is MD-consistent with S , or when there is no subtree included in T that can be obtained with c leaf removals.

Theorem 2 *The algorithm computes if there exists a solution of size at most c for MinLeafRem in time $O(3^c \text{poly}(|V(T)|, |V(S)|))$.*

Proof. The correctness of the algorithm follows from Lemma 6.

Now, we focus on the time complexity of the algorithm. At each step the algorithm branches in three possible cases, and for each of these cases at least one leaf is removed. As the depth of the search tree is bounded by c , the size of the search tree is bounded by 3^c . Since after each branching we require at most time $O(\text{poly}(|V(T)||V(S)|))$ to compute T' , $\text{lca}_{T',S}$, and $\text{Dup}(T')$, it follows that the overall time complexity of the algorithm is $O(3^c \text{poly}(|V(T)||V(S)|))$.

4.2 MinLeafRem Parameterized by the Number of Labels with Multiple Copies

In this section we give a fixed-parameter algorithm for MinLeafRem, when the parameter is the number of labels associated with multiple leaves of T . Denote by $\Gamma_D \subseteq \Gamma$, the subset of labels associated with multiple leaves of T .

Let x be a node of T , having children x_l, x_r , and let y be a node of S , with children y_l, y_r . Given $\Gamma'_D \subseteq \Gamma_D$, we define $M[T(x), S(y), \Gamma'_D]$ as the minimum number of leaves that have to be removed to obtain a tree T' included in $T(x)$ such that (1) T' is MD-consistent with $S(y)$ and (2) the subset $\Gamma'_D \subseteq \Gamma(T')$. We can compute $M[T(x), S(y), \Gamma'_D]$ applying the following recurrence:

$$M[T(x), S(y), \Gamma'_D] = \min_{\substack{\Gamma'_{1,D} \subseteq \Gamma'_D, \\ \Gamma'_{2,D} \subseteq \Gamma'_D, \\ \Gamma'_{1,D} \cup \Gamma'_{2,D} = \Gamma'_D}} \begin{cases} M[T(x_l), S(y_l), \Gamma'_{1,D}] + M[T(x_r), S(y_r), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y_r), \Gamma'_{1,D}] + M[T(x_r), S(y_l), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y), \Gamma'_{1,D}] + M[T(x_r), S(y), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset \\ M[T(x_l), S(y), \Gamma'_D] + |L(T(x_r))| \\ M[T(x_r), S(y), \Gamma'_D] + |L(T(x_l))| \\ M[T(x), S(y_l), \Gamma'_D] \\ M[T(x), S(y_r), \Gamma'_D] \end{cases} \quad (1)$$

Now, we define the basic cases of the recurrence, when each of $T(x)$ and $S(y)$ is a single leaf, with $\Gamma(T(x)) = \lambda_G$ and $\Gamma(S(y)) = \lambda_S$. If $\lambda_G = \lambda_S$, then $M[T(x), S(y), \Gamma'_D] = 0$ if $\Gamma'_D = \{\lambda_G\}$, $M[T(x), S(y), \Gamma'_D] = 0$ if $\Gamma'_D = \emptyset$, else $M[T(x), S(y), \Gamma'_D] = +\infty$. If $\lambda_G \neq \lambda_S$, then $M[T(x), S(y), \Gamma'_D] = 1$ if $\Gamma'_D = \emptyset$, else $M[T(x), S(y), \Gamma'_D] = +\infty$.

The correctness of Recurrence 1, is proved in the following lemma.

Lemma 7 *Let T be a gene tree, let S be a species tree, and let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T . Let x be a node of T and y be a node of S , and consider a subset $\Gamma'_D \subseteq \Gamma_D$. Then $M[T(x), S(y), \Gamma'_D] = c$ if and*

only if there exists a tree T' included in $T(x)$ such that (i) T' is MD-consistent with $S(y)$; (ii) T' is obtained by removing c leaves; (iii) $\Gamma'_D \subseteq \Gamma(T')$.

Theorem 3 Given a gene tree T and a species tree S , let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T . Then an optimal solution of *MinLeaf* over instance (T, S) can be computed in time $O(4^{|\Gamma_D|} \text{poly}(|V(T)||V(S)|))$.

Proof. By Lemma 7 a solution of *MinLeaf* over instance (T, S) , is obtained looking for the minimum of the values $M[T(r_T), S(r_S), \Gamma'_D]$, for each $\Gamma'_D \subseteq \Gamma_D$, where r_T (r_S respectively) is the root of T (S respectively).

Now, we prove in the following that the time complexity of the algorithm is $O(4^{|\Gamma_D|} \text{poly}(|V(T)||V(S)|))$. It is easy to see that the time complexity to compute Recurrence 1 is dominated by case 3. The entries $M[T(x), S(y), \Gamma'_D]$ are $O(2^{|\Gamma_D|} |V(T)||V(S)|)$. For each pair of nodes $x \in V(T)$, $y \in V(S)$, we have to consider $O(4^{|\Gamma_D|})$ possible combinations. Indeed, the number of subsets $\Gamma'_{1,D}, \Gamma'_{2,D} \subseteq \Gamma'_D$, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$, is $4^{|\Gamma_D|}$, since we have to consider all possible subsets Γ'_D of Γ_D and, for each subset Γ'_D , we have to consider all possible subsets $\Gamma'_{1,D}, \Gamma'_{2,D} \subseteq \Gamma'_D$, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$. It follows that we have to consider $4^{|\Gamma_D|}$ combinations, since there are $4^{|\Gamma_D|}$ possible ways to split set Γ_D into four disjoint subsets (in this case the subsets are $\Gamma_D \setminus \Gamma'_D$, $\Gamma'_{1,D} \setminus \Gamma'_{2,D}$, $\Gamma'_{2,D} \setminus \Gamma'_{1,D}$, and $\Gamma'_{1,D} \cap \Gamma'_{2,D}$). For each combination, the recursion can be computed in constant time.

Finding the minimum value in the entries $M[T(r_G), S(r_S), \Gamma'_D]$ requires time $O(2^{|\Gamma_D|} |V(T)||V(S)|)$, hence the overall time complexity to find an optimal solution of *MinLeafRem* over instance (T, S) , is $O(4^{|\Gamma_D|} \text{poly}(|V(T)||V(S)|))$.

5 Conclusion

We presented complexity results and gave two parameter tractable versions of the Minimum Leaf Removal Problem. This problem has been shown to be a natural one to consider for preprocessing gene trees prior to reconciliation [8]. Even though the problem is proved to be APX-hard, a polynomial-time heuristic, showing a good performance on simulated data sets, has already been developed [12]. The fixed-parameter algorithms presented in this paper nicely complement those in [12].

In the case of species tree inference, it has been shown in [8] that deciding whether a gene tree T is an MD-tree, i.e. a tree that is MD-consistent with at least one species tree, can be done in polynomial time and space, as well as computing a parsimonious species tree. In the case of a tree T being not an MD-tree, a natural extension of the Minimum Leaf Removal Problem would be to find the minimum number of leaves that have to be removed from a given gene tree T in order for T to be an MD-tree. Having appropriate solutions for this problem would give natural ways for correcting gene trees prior to species tree inference. We are presently studying the theoretical complexity of this problem.

Acknowledgements

We thank Krister M. Swenson for his careful reading of the proofs, and his advices on notations and presentation of the paper.

References

1. Alimonti, P., Kann, V.: Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.* 237(1–2), 123–134 (2000)
2. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. *J.Mol.Biol.* 215(3), 403–410 (1990)
3. Arvestad, L., Berglung, A.C., Lagergren, J., Sennblad, B.: Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In: Gusfield, D. (ed.) *RECOMB 2004*. pp. 326–335. ACM, New York (2004)
4. Blin, G., Bonizzoni, P., Dondi, R., Rizzi, R., Sikora, F.: Complexity insights of the minimum duplication problem. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) *SOFSEM 2012*. LNCS, vol. 7147, pp. 153–164. Springer, Heidelberg (2012)
5. Blomme, T., Vandepoele, K., Bodt, S.D., Silmillion, C., Maere, S., van de Peer, Y.: The gain and loss of genes during 600 millions years of vertebrate evolution. *Genome Biology* 7, R43 (2006)
6. Bonizzoni, P., Della Vedova, G., Dondi, R.: Reconciling a gene tree to a species tree under the duplication cost model. *Theoretical Computer Science* 347, 36–53 (2005)
7. Chang, W., Eulenstein, O.: Reconciling gene trees with apparent polytomies. In: Chen, D., Lee, D.T. (eds.) *COCOON 2006*. LNCS, vol. 4112, pp. 235–244. Heidelberg (2006)
8. Chauve, C., El-Mabrouk, N.: New perspectives on gene family evolution: losses in reconciliation and a link with supertrees. In: Batzoglou, S. (ed.) *RECOMB 2009*. LNCS, vol. 5541, pp. 46–58. Springer, Heidelberg (2009)
9. Chen, K., Durand, D., Farach-Colton, M.: Notung: Dating gene duplications using gene family trees. *Journal of Computational Biology* 7, 429–447 (2000)
10. Cotton, J., Page, R.: Rates and patterns of gene duplication and loss in the human genome. *Proceedings of the Royal Society of London. Series B* 272, 277–283 (2005)
11. Demuth, J., Bie, T.D., Stajich, J., Cristianini, N., Hahn, M.: The evolution of mammalian gene families. *PLoS ONE* 1:e85 (2006)
12. Doroftei, A., El-Mabrouk, N.: Removing noise from gene trees. In: Przytycka, T.M., Sagot, M.F. (eds.) *WABI 2011*. LNBI, vol. 6833, pp. 76–91. Springer, Heidelberg (2011)
13. Durand, D., Haldórsson, B., Vernet, B.: A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology* 13, 320–335 (2006)
14. Eichler, E., Sankoff, D.: Structural dynamics of eukaryotic chromosome evolution. *Science* 301, 793–797 (2003)
15. Goodman, M., Czelusniak, J., Moore, G., Romero-Herrera, A., Matsuda, G.: Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology* 28, 132–163 (1979)

16. Górecki, P., Eulenstein, O.: A linear time algorithm for error-corrected reconciliation of unrooted gene trees. In: Chen, J., Wang, J., Zelikovsky, A. (eds.) ISBRA 2012. LNCS, vol. 6674, pp. 148- 159. Springer, Heidelberg (2011)
17. Gorecki, P., Tiuryn., J.: DLS-trees: a model of evolutionary scenarios. *Theoretical Computer Science* 359, 378–399 (2006)
18. Guigó, R., Muchnik, I., Smith, T.: Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution* 6, 189–213 (1996)
19. Hahn, M.: Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. *Genome Biology* 8(R141) (2007)
20. Hahn, M., Han, M., Han, S.G.: Gene family evolution across 12 *drosophila* genomes. *PLoS Genetics* 3:e197 (2007)
21. Kristensen, D., Wolf, Y., Mushegian, A., Koonin, E.: Computational methods for gene orthology inference. *Briefings in Bioinformatics* 12(5), 379- 391 (2011)
22. Ma, B., Li, M., Zhang, L.: From gene trees to species trees. *SIAM J. on Comput.* 30, 729–752 (2000)
23. Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford (2006)
24. Ohno, S.: *Evolution by gene duplication*. Springer, Berlin (1970)
25. Page, R.: Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology* 43, 58–77 (1994)
26. Page., R.: Genetree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics* 14, 819–820 (1998)
27. Page, R., Charleston, M.: Reconciled trees and incongruent gene and species trees. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 37, 57–70 (1997)
28. Page, R., Cotton, J.: Vertebrate phylogenomics: reconciled trees and gene duplications. In: *Pacific Symposium on Biocomputing*. pp. 536–547 (2002)
29. Sanderson, M., McMahon, M.: Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evolutionary Biology* 7, S3 (2007)
30. Vernet, B., Stolzer, M., Goldman, A., Durand, D.: Reconciliation with non-binary species trees. *Journal of Computational Biology* 15, 981–1006 (2008)