EasyChair Preprint

№ 636

Minimising the Deep Coalescence

Dawid Dabkowski and Pawel Gorecki

November 15, 2018

Minimising the Deep Coalescence

Dawid Dąbkowski and Paweł Górecki

Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
Warsaw, mazowieckie, 00-927, Poland
dd345130@students.mimuw.edu.pl, gorecki@mimuw.edu.pl

Abstract

Metagenomic studies identify the species present in an environmental sample usually by using procedures that match molecular sequences, e.g., genes, with the species taxonomy. Here, we formulate the problem of gene-species matching in the parsimony framework using phylogenetic gene and species trees under the deep coalescence cost and the assumption that each gene is paired uniquely with one species. In particular, we solve the problem in the cases when one of the trees is *caterpillar*. Next, we generalize the solution and propose several heuristic algorithms. Finally, we present the results of computational experiments on simulated and empirical datasets.

keywords: deep coalescence, metagenomics, species taxonomy, gene tree

1 Introduction

One of the primary goals of metagenomic studies is to identify the species present in an environmental sample. Such identification from metagenomics data is usually computationally demanding and requires complex workflows in which molecular sequences identified in the sample, i.e., reads, genes or contigs, can be matched with the species taxonomy. The gene-species matching can be expressed by using a partially labeled phylogenetic tree, where a gene tree inferred from a set of homologous sequences extracted from the sample is matched with the known species taxonomy. Here, we formulate the problem in the parsimony framework using gene and species trees under the deep coalescence model and the assumption that each gene is paired uniquely with one species.

Deep coalescence is a major process that can lead to a discordance between a gene tree and its species tree. It occurs when the time at which lineages of alleles coalesce, predates speciation events of the alleles' species [1, 2]. The discordance caused by deep coalescence can be measured by using *the deep coalescent cost* which, given two labeled trees, is efficiently computable in linear time [3]. Consequently, the cost has been studied in the context of classical problems in computational biology, e.g., gene tree parsimony [4–7], tree reconciliation [8], error correction [9] or tree rooting [10, 11].

In this article, we analyze the deep coalescence cost for a pair of bijectively labeled gene and species trees. We investigate into gene-species matching problem expressed as the *minimisation problem*, that is, *given two unlabeled trees find bijective leaf-labelings for these trees that minimise the deep coalescence cost.* While several variants of the dual maximising problem can be solved in polynomial time [12–14], little is known about the minimising problem. The closest is the minimisation problem for the general leaf-labelings, i.e., without the requirement of bijectivity. Usually, such a problem can be solved by a dynamic programming in polynomial time [15–17].

In this article, we solve the gene-species problem for bijectively labeled leaves in the cases when one of the trees is *caterpillar* by using two tree ordering operations. Next, we generalize the solution and propose several heuristic algorithms the problem for any gene and species tree topology. Finally, we present the results of computational experiments on simulated and empirical datasets.

2 Basic definitions

A *tree* in this article is a rooted binary tree $T = \langle V(T), E(T) \rangle$ such that the edges of T are directed towards leaves, i.e., if $\langle v, w \rangle \in E(T)$ then v is the parent of w. The edges incident to the *root* are called *top*. By $T(v)$ we denote a subtree of T rooted at a node v. A cluster of v, denoted $C_T(v)$, is the set of all leaves of $T(v)$. By $|T|$ we denote the *size* of T, that is, the number of its leaves. By $h(T)$ we denote the *height* of T, i.e., the maximal number of edges on the path from the root to a leaf of T. If v is a non-root node, then v_P is the parent of v and v_S is the sibling of v.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a fixed set of $n > 1$ taxa.

A *labeled tree* over X is a tree having exactly n leaves bijectively labeled by the elements from X. A labeled tree is often *ordered*. In such a case, each internal node v has the left and the right child, denoted v_L and v_R, respectively. An ordered tree T induces a *labeling* $\Lambda_T\colon [n] \to X$,[1] such that $\Lambda_T(1), \Lambda_T(2), \ldots, \Lambda_T(n)$ are the taxons obtained from the leaves by the left to right traversal of T. A labeling that satisfies $\Lambda_T(i) = x_i$, for $i \in [n]$ will be called *simple*. For a node v, the subtree $T(v)$ has the labeling inherited from T.

Each edge $e \in E(T)$ can be uniquely identified by the child, therefore, in notation, we often use an edge and its terminating node (the child) alternatively. For instance, the subtree $S(v)$ can be denoted as $S(e)$ if $e = \langle v_P, v\rangle$.

In computational biology, we recognize two types of trees: a *gene tree* and a *species tree*. In this article, they are both labeled trees over the same set of taxa. Now we introduce the least common ancestor mapping, in short, LCA-mapping, from a gene tree G to a species tree S. An example is depicted in Fig. 1.

Definition 1. LCA-*mapping from a gene tree G to a species tree S is a function* $\mathsf{LCA}_S\colon G \to S$ *such that for a node v of G,* $\mathsf{LCA}_S(v)$ *is the lowest node s of S, such that each taxon from $G(v)$ is present in $S(s)$.*

Based on the LCA-mapping we can embed a gene tree G into S by mapping every edge $\langle v, w\rangle$ of G to the path in S whose endpoints are $\mathsf{LCA}_S(v)$ and $\mathsf{LCA}_S(w)$. The edges of such paths are called *lineages*. Embeddings can be visualized in the form depicted in Fig. 1. If both trees are equal, then the LCA-mapping is a bijection, and an edge is bijectively mapped to an edge. Otherwise, the embedding has some number of *extra lineages* present on edges of the species tree. For an edge $e \in E(S)$, the number, denoted $\mathsf{xl}(G, e)$, can be defined formally as [13]

$$\mathsf{xl}(G, e) = |C_S(e)| - c_e - 1,$$

where c_e is the number of internal nodes of G that are mapped to nodes of $S(e)$.

Having this, we define the deep coalescence cost.

Definition 2. *For a gene tree G and a species tree S the deep coalescence cost is defined as* $\mathsf{dc}(G, S) = \sum_{e\in E(S)} \mathsf{xl}(G, e)$.

Equivalently, it can be shown that $\mathsf{dc}(G, S) = \sum_{\langle v,w\rangle\in E(S)} \|\mathsf{LCA}_S(v), \mathsf{LCA}_S(w)\| - 1$, where $\|s, s'\|$ denote the number of edges on the shortest path connecting s and s'.

Now, we will investigate into the minimal deep coalescence cost for fixed tree topologies. For a given tree T (labeled or not) by $\mathbb{S}(T)$ we denote the set of all labeled trees T' over X such that $V(T) = V(T')$ and $E(T) = E(T')$. In other words, the elements of $\mathbb{S}(T)$ share the tree topology.

[1] $[n] = \{1, 2, \ldots, n\}$

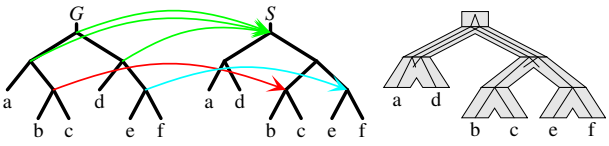


Figure 1: *Left:* An example of a gene tree G and a species tree S over $X = \{a, b, c, d, e, f\}$ with the LCA-mappings of internal nodes of G. *Right:* The embedding, or evolutionary scenario [18], explains the differences between G and S by drawing G inside S. Here, $\mathsf{dc}(G, S) = 2$ as each top edge of S has one extra lineage.

Problem 1 (MinDC)**.** *Given trees G and S. Find the minimal* $\mathsf{dc}(G', S')$*, denoted* $\widetilde{\mathsf{dc}}(G, S)$*, in the set of all pairs* $\langle G', S'\rangle$ *from* $\mathbb{S}(G) \times \mathbb{S}(S)$.

From the practical point of view, the most critical problem is to infer the minimal labelings, which encode the gene-species mappings. This can be expressed by seeking for the optimal gene tree as follows.

Problem 2 (Gene-Species Matching)**.** *Given a tree G and a species tree S. Find $G^* \in \mathbb{S}(G)$ such that* $\mathsf{dc}(G^*, S) = \widetilde{\mathsf{dc}}(G, S)$.

3 Results

In this section, we show how to solve our problems when one of the trees is a caterpillar, i.e., the maximum-height tree $\mathcal{C}_n$.

3.1 Caterpillar species tree

We say that an ordered tree T is size-*ordered* if for each internal node v we have $|T(v_L)| \leq |T(v_R)|$.

Theorem 1. *Given a size-ordered gene tree G and a size-ordered species tree $\mathcal{C}_n$. If both are simply labeled then* $\widetilde{\mathsf{dc}}(G, \mathcal{C}_n) = \mathsf{dc}(G, \mathcal{C}_n)$*. Furthermore,*

$$\widetilde{\mathsf{dc}}(G, \mathcal{C}_n) = \widetilde{\mathsf{dc}}(G_L, \mathcal{C}_{|G_L|}) + \widetilde{\mathsf{dc}}(G_R, \mathcal{C}_{|G_R|}) + |G_L| - 1, \quad (1)$$

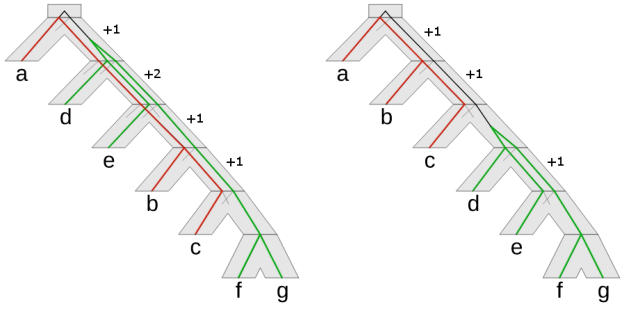
where G_L and G_R are the left and the right subtree of G, respectively.

Proof. Without loss of generality, we assume that the labeling of G is simple. Let us consider any labeling of $\mathcal{C}_n$. First, we embed G_L and G_R separately, and then,

we join them by embedding the top edges. We can write that,

$$\mathsf{dc}(G, \mathcal{C}_n) = \mathsf{dc}(G_L, \mathcal{C}_{|G_L|}) + \mathsf{dc}(G_R, \mathcal{C}_{|G_R|}) + K, \quad (2)$$

where $K > 0$ is the number of additional extra lineages from the top edges and the intersection of subtrees in the embedding. Here, the labelings of $\mathcal{C}_{|G_L|}$ and $\mathcal{C}_{|G_R|}$ are inherited from $\mathcal{C}_n$. An example is depicted in Fig. 2.



$K = 4$, $\mathsf{dc}(G, S_1) = 5$ $\quad$ $K = 2$, $\mathsf{dc}(G, S_2) = 3$

Figure 2: Embeddings of a simple labeled gene tree $G = ((a,(b,c)),((d,e),(f,g)))$ into $\mathcal{C}_n$ with two labelings. The lineages of $G_L = (a,(b,c))$ are red, $G_R = ((d,e),(f,g))$ are green, while the lineages of top edges of G are black. The contribution to dc is marked next to edges. The right embedding is optimal, as the tree is simple labeled and the lineages of G_L and G_R are disjoint.

We show that the best labeling of $\mathcal{C}_n$ is simple. The proof is by induction on the size of T. For $n = 1$ it is trivial. Assume that the statement holds true for trees of the size smaller than n. We want to minimize the value of $\mathsf{dc}(G, \mathcal{C}_n)$.

Let Λ be the labeling of the species tree $\mathcal{C}_n$ and let $m = |G_L| \leq |G_R|$. Let A and the set of indices of taxons mapped to $\mathcal{C}_n$ from G_L [2]. If Λ is simple, then $A = [m]$ and $K = m-1$ as the lineages of G_L and G_R are disjoint and there are $m-1$ lineages of the top edges (see Fig. 2). In general, for any A, let $0 = l_0 < l_1 < l_2 < \ldots < l_k = n$ be the maximal sequence such that for $1 \leq j < k$ either $l_j \in A \wedge l_j + 1 \notin A$ or $l_j \notin A \wedge l_j + 1 \in A$. E.g. for the left tree from Fig. 2, $k = 4$, $n = 7$, $l_1 = 1$, $l_2 = 3$, $l_3 = 5$ and $l_4 = 7$. Now we have $[n]$ split into k parts $P_j := \{l_{j-1} + 1, \ldots, l_j\}$ for $j \in [k]$. We can imagine a tree G_j as a tree contracted to the set of taxons from $\Lambda[P_j]$. When embedding G into $\mathcal{C}_n$, we can inductively embed G_L (and similarly G_R) as proposed in formula (2). This can be done by using every second G_j's tree and calculating only additional extra lineages in embeddings of G_j and G_{j+2} that are separated by the embedding of G_{j+1}. Including the lineages of top edges, to calculate K we have the following observations. Let s_j be the LCA-mapping of the root of G_j. For every $3 \leq j \leq k$, G_j has to be connected with G_{j-2}, which requires $|G_{j-1}| - 1$ lineages, located on the path connecting s_{j-1} with the parent of s_j, shared with the lineages of G_{j-1}. Similarly, G_2 needs $|G_1| - 1$ lineages between the parent of s_2 and the root of S. Next, if $2 \leq j < k$, there is one more lineage, i.e., the edge whose terminating node is s_j, shared with the lineages connecting G_{j-1} and G_{j+1}. We have that $|G_j| = l_j - l_{j-1}$, hence $K = k - 2 + \sum_{j=2}^{k}(|G_{j-1}| - 1) = l_{k-1} - 1 \geq \min(m, n-m) - 1 = m - 1$. So, for every labeling of $\mathcal{C}_n$, K is bounded, and this boundary is achieved only when $k = 2$. By the inductive assumption, this statement joined with the previous observations, completes the proof. □

[2] Formally, $A = \Lambda^{-1}(\Lambda_G[\{1, 2, \ldots, m\}])$.

The next theorem shows that to compute the minimal dc for the caterpillar species tree, it is sufficient to order the gene tree by size.

Theorem 2. *If G is a* size*-ordered gene tree then*

$$\widetilde{\mathsf{dc}}(G, \mathcal{C}_n) = \sum_{e \in \mathsf{Lft}(G)} |G(e)| - 1, \quad (3)$$

where $\mathsf{Lft}(G)$ is the set of all edges in G that connect a node with its left child.

Proof. It follows immediately from Thm. 1. □

Note that we cannot fully classify the minimal cost trees by writing that for a gene tree G, $\widetilde{\mathsf{dc}}(G, \mathcal{C}_n) = \mathsf{dc}(G, \mathcal{C}_n)$ if and only if G is a size-ordered tree and $\Lambda_G = \Lambda_{\mathcal{C}_n}$. This statement does not hold in general, e.g., we can swap leaves of f and g in the left species tree from Fig. 2 and the minimal cost will be preserved.

The compact formula (3), or the recursive formula (1), allows us to compute $\widetilde{\mathsf{dc}}$ in linear time. Now let us also fix the topology of G to be a *complete balanced* tree of height h, i.e., a tree of the size 2^h with all 2^h leaves on the same depth, and calculate deep coalescence cost. We have: $\mathsf{dc}(\mathcal{B}_n, \mathcal{C}_n) = \sum_{i=1}^{h} 2^{i-1} \cdot (2^{h-i} - 1) = \frac{1}{2}(\sum_{i=1}^{h} 2^h - 2^i) = \frac{1}{2}(h \cdot 2^h - 2 \cdot \frac{1-2^h}{1-2}) = \frac{1}{2}(n \log_2 n + 2(1-n)) = \frac{n}{2}(\log_2 n - 2) + 1$

It shows that $\widetilde{\mathsf{dc}}$ between a complete binary and a caterpillar tree is $\sim \frac{n}{2} \log_2 n$. Having this, one may conjecture, that the maximal $\widetilde{\mathsf{dc}}$ for any gene tree versus a caterpillar species tree, both of the size n, is $\sim \frac{n}{2} \log_2 n$.

3.2 Caterpillar gene tree

In this Section, we show how to solve our Problems in the case when a gene tree is a caterpillar. The solution is

similar to the previous case, with the difference that we need a new type of order. For a node $v \in V(T)$ a *saving* of v, denoted $\mathsf{sav}(v)$, is defined recursively: $\mathsf{sav}(v) = \max(\mathsf{sav}(v_L), \mathsf{sav}(v_R)) + |T(v)| - 1$, where $\mathsf{sav}(v) = 0$ if v is a leaf. We say that a tree is sav-ordered, if, for every internal node v we have $\mathsf{sav}(v_L) \leq \mathsf{sav}(v_R)$. An example of a sav-ordered tree is depicted in Fig. 3.

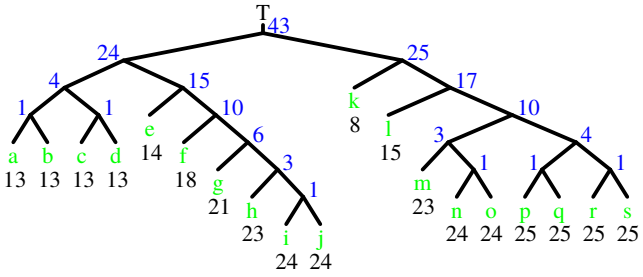


Figure 3: An example of a sav-ordered tree T with the decoration showing the values of sav for the internal nodes, and the weights $\omega_i(T)$ each leaf. This is one of the two smallest trees (to obtain the second one replace $((a,b),(c,d))$ by $(a,(b,(c,d)))$) showing that sav-ordeding is significantly different than size-ordering. Furthermore, this example shows that sav-ordering cannot be replaced by a potentially simpler ordering based on the height of subtrees.

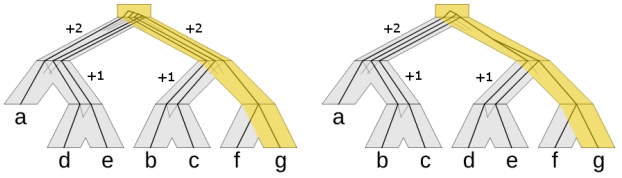
Let $E_i(T)$ denote the set of edges on the path connecting the root of T with the i-th leaf. Let $\omega_T(i) = \sum_{e \in E_i(T)} |T(e)| - 1$ be the *weight* of the i-th path. First, we show that, for a sav-ordered tree T, $\omega_i(T)$ is maximized by the rightmost path.

Lemma 1. *For any* sav*-ordered tree* T, $\max_i \omega_T(i) = \mathsf{sav}(\mathsf{root}(T)) - |T| + 1$. *Moreover, the maximum is reached by the rightmost path, i.e., for* $i = n$.

Proof. The proof is by induction on the size of T. For $n = 1$ it is trivial. We assume that the statements hold for trees of the size smaller than n. First, we partition the set of paths: $\max_i \omega_T(i) = \max(\max_i \omega_{T_L}(i) + |T_L| - 1, \max_i \omega_{T_R}(i) + |T_R| - 1)$. Now, from the induction assumption and the definition of saving this value equals $\max(\mathsf{sav}(\mathsf{root}(T_L)), \mathsf{sav}(\mathsf{root}(T_R)))$ $= \mathsf{sav}(\mathsf{root}(T)) - |T| + 1$, which completes the first part of the proof. For the second path, observe, that the tree is sav-ordered, hence $\max(\mathsf{sav}(\mathsf{root}(T_L)), \mathsf{sav}(\mathsf{root}(T_R)))$ $= \mathsf{sav}(\mathsf{root}(T_R))$. Finally, by induction assumption, we have $\mathsf{sav}(\mathsf{root}(T_R)) = \omega_T(n)$. □

For a tree S and $i \in [n]$, by S^i we denote the tree obtained from S as follows.

- Let $v_1, v_2 \ldots v_k$ be nodes on the path from the root to the i-th leaf.
- For $j = 1, 2, \ldots, k$, if the left child of v_j is v_{j+1} then swap the subtrees of v_j.



$dc(G, S_1) = 6$ $\qquad$ $dc(G, S_2) = 4$

Figure 4: Embeddings of a simple labeled gene tree $G = (a,(b,(c,(d,(e,(f,g))))))$ into species trees S_1 and S_2. The rightmost paths are colored in yellow. The number of extra lineages is shown near the corresponding edges. S_2 has simple labeling, therefore the right embedding is optimal. In particular, its rightmost path has no extra lineages.

Obviously, this transformation does not change the deep coalescence so $\widetilde{\mathsf{dc}}(G, S) = \widetilde{\mathsf{dc}}(G, S^i)$ for any i, G and S. Also, if S is sav-ordered then $S^n = S$. Now we can formulate our main theorem for the case of caterpillar gene trees.

Theorem 3. *Let* $\mathcal{C}_n$ *be a caterpillar gene tree, and* S *be a species tree. Assume that both are* sav*-ordered. If both have simple labeling, then* $\widetilde{\mathsf{dc}}(\mathcal{C}_n, S) = \mathsf{dc}(\mathcal{C}_n, S)$.

Proof. For simplicity, let $\overline{E}_n(S) = E(S) \setminus E_n(S)$. Note that the n-th leaf in a simple labeled $\mathcal{C}_n$ is the deepest node in $\mathcal{C}_n$ and $\Lambda_{\mathcal{C}_n}(n) = x_n$.[3] By $\widetilde{\mathsf{dc}}_i(\mathcal{C}_n, S)$ we denote the minimal $\mathsf{dc}(\mathcal{C}_n, S)$ in the set of all species trees S with the labeling satisfying $\Lambda_S(i) = x_n$. We split the proof into two parts. First, we show that $\widetilde{\mathsf{dc}}_i(\mathcal{C}_n, S)$ is determined by the simple labeling of S^i and equals $\sum_{e \in \overline{E}_n(S^i)} (|S^i(e)| - 1)$. Secondly, we prove that this sum is minimal if $i = n$.

Part I. Let $\Lambda_S(i) = x_n$. Clearly, $\widetilde{\mathsf{dc}}_i(\mathcal{C}_n, S) = \widetilde{\mathsf{dc}}_n(\mathcal{C}_n, S^i)$, i.e., we consider S^i with the n-th leaf labeled by x_n. Note, that every internal node of $\mathcal{C}_n$ maps to a node from the path $E_n(S^i)$ as x_n is the label of the n-th node from S^i and $\mathcal{C}_n$. Hence, if $e = \langle v, w \rangle$ is an edge from $\overline{E}_n(S^i)$, e is a lineage for every taxon (leaf) below v when embedding $\mathcal{C}_n$ into S^i. Thus, e is exactly $|S^i(e)|$ times a lineage, which gives $|S^i(e)| - 1$ extra lineages. We conclude that $\widetilde{\mathsf{dc}}_n(\mathcal{C}_n, S^i) \geq \sum_{e \in \overline{E}_n(S^i)} (|S^i(e)| - 1)$. Now, we show that this boundary is reached by the simple labeling of S^i. In such a case, for $j < n$, the lineages of the edge adjacent to the j-th leaf of $\mathcal{C}_n$ are disjoint with the rightmost path of S^i. Moreover, there is no extra

[3] Recall that x_n is the last taxon from the taxon set X.

lineage in $E_n(S^i)$ (see the right embedding in Fig. 4). This completes the proof of the first part.

Part II. Let $W(S) = \sum_{e \in S}(|S(e)| - 1)$. Note that $W(S) = W(S^i)$. It follows from the first part that $\widetilde{\mathsf{dc}}_n(\mathcal{C}_n, S^i) = \sum_{e \in \overline{E}_n(S^i)}(|S^i(e)| - 1) = W(S^i) - \sum_{e \in E_n}(|S^i(e)| - 1) = W(S) - \omega_n(S^i) = W(S) - \omega_i(S)$. Hence, we have $\widetilde{\mathsf{dc}}(\mathcal{C}_n, S) = \min_i \widetilde{\mathsf{dc}}_i(\mathcal{C}_n, S) = W(S) - \max_i \omega_i(S)$. Finally, by Lemma 1 we have that $\widetilde{\mathsf{dc}}(\mathcal{C}_n, S) = W(S) - \omega_n(S)$, i.e., when $i = n$ and $S^n = S$ is simply labeled. This completes the proof. □

Theorem 4. *If S is* sav*-ordered then*

$$\widetilde{\mathsf{dc}}(\mathcal{C}_n, S) = \sum_{e \in E(S) \setminus E_n(S)} |S(e)| - 1. \qquad (4)$$

Proof. Under the notation from the second part of the proof of Thm. 3 we have $\widetilde{\mathsf{dc}}(\mathcal{C}_n, S) = W(S) - \omega_n(S)$. The rest follows by expanding $W(S)$ and $\omega_n(S)$. □

The formula (4) allows us to compute the minimal deep coalescence cost in $O(n)$ time. Now we can compute easily the minimal cost for the complete balanced species tree when $n = 2^k$. We have $\widetilde{\mathsf{dc}}(\mathcal{C}_n, \mathcal{B}_n) = \sum_{i=1}^{k}(2^i - 1)(2^{k-i} - 1) = \sum_{i=1}^{k}(2^k - 2^i - 2^{k-i} + 1) = k2^k - (2^{k+1} - 2) - 2^k(1 - 2^{-k}) + k = 2^k(k - 2 - 1) + 2 + 1 + k = n(\log_2 n - 3) + \log_2 n + 3$.

It shows that $\widetilde{\mathsf{dc}}$ for the caterpillar and the complete binary tree is $\sim n \log_2 n$, which is similar to the results obtained in the previous section. Having this, one may also conjecture, that the maximal $\widetilde{\mathsf{dc}}$ for a caterpillar gene tree versus any species tree, both of the size n, is $\sim \frac{n}{2} \log_2 n$.

3.3 Algorithms for Gene-Species Matching

Here, we propose several heuristic algorithms for solving our problems. The algorithms, given the input consisting of two unlabeled trees of the same size, alter the ordering of nodes and infer labelings that approximate the minimal deep coalescence cost. Then, to compute the dc cost for such trees, we use the classical $O(n)$ algorithm based on LCA queries [3].

Algorithm 1: The simple algorithm

1: **Input:** Trees G and S of the same size.
2: **Output:** Approximation of $\widetilde{\mathsf{dc}}(G, S)$.
3: Order G by size and S by saving.
4: Add simple labelings to G and S.
5: **Return** $\mathsf{dc}(G, S)$.

Alg. 1 has a linear time and space complexity. Next, it follows from Thm. 1 and 3, that the simple algorithm is exact if one of the input trees is caterpillar as for caterpillar trees ordering by size and by saving are equivalent.

Algorithm 3: Extended greedy algorithm

1: **Input/Output:** see Alg. 2.
2: *Notation:* For a tree T, let $K(T)$ be the set of maximal nodes v, such that the left and the right subtree of v are isomorphic.
3: Order G by size and S by saving.
4: **return** $\min(d(G, S), \min_{g \in K(G), s \in K(S)} d(G^g, S^s))$, where G^g (and similarly S^s) is a tree obtained from G by swapping subtrees of v_j if, for each $j < k$, v_{j+1} is the left child of v_j, where $v_1, v_2, \ldots, v_k$ is the path from the root of G to g.
5: **Function** $d(G, S)$: all lines ≥ 4 from Alg. 2

Although the simple algorithm fits our theorems perfectly, one could find even small counterexample when the output cost is not optimal. Therefore, we propose another approach (see Alg. 2), in which we first try to match *cherries*, which are nodes with precisely two leaves beneath. Empirical evaluation shows, see Fig. 5 that the greedy algorithm performs better than Alg. 1 in terms of the returned cost. Alg. 2 has a quadratic time complexity. It is also more difficult to find a counterexample which does not give the lowest cost.

Extending algorithms. To further improve the performance of our algorithms we propose to apply different kinds of orderings in some nodes of the input trees. The details how to extend Alg. 2 are depicted in Alg. 3. Analogously, we extend the simple algorithm. Both extended algorithms are never worse than the original ones, and we still have the exact solution for caterpillar trees. For the other trees, extended algorithms tend to perform better, which is summarized in Fig. 5. As the set of nodes $K(T)$ in Alg. 3 can be computed by using an $O(n \log n)$ the solution proposed by Campbell et al. [19], the time complexity of the extended greedy algorithm is $O(n^4)$, while the extended variant of the simple algorithm requires $O(n^3)$ time.

The evaluation of all proposed algorithms is depicted in Fig. 5.

4 Experimental Results

We have performed two computational experiments on empirical and simulated datasets. In the first experiment, we present a comparative study of the reconstruction algorithms, while in the second, we tested the quality of labeling inference.

Experiment I. To verify which algorithm yields

Algorithm 2: The greedy algorithm

Input: Trees G and S of the same size. **Output:** Approximation of $\widetilde{\mathsf{dc}}(G, S)$.
Notation: For a tree T and a set of nodes $Z \subseteq V(T)$ and $i \in \{1, 2, \ldots, |Z|\}$, by $Z[i]$ we denote i-th node from Z when T is traversed in post-order. By I_T we denote the set of internal nodes of a tree T.
Order G by size and S by saving.
Add the simple labeling to G. Let $i := j := 1$. Initialize sets $M_G := M_S := \emptyset$.
$F := C_G(\mathsf{root}(G))$ - the set of unmapped leaves from G; $U := C_S(\mathsf{root}(S))$ - the set of unlabeled leaves in S.
While $j \leq n - 1$
    $A := |F \cap C_G(I_G[i])|$ and $B := |U \cap C_G(I_S[j])|$
    If $|A| = |B|$ **Then** $\mathsf{map}(A, B)$; i+=1; j+=1;
    Else If $|A \cup M_G| = |B \cup M_S|$ **Then** $\mathsf{map}(A \cup M_G, B \cup M_S)$; $M_G := M_S := \emptyset$; i+=1; j+=1;
    Else If $|A \cup M_G| = |B|$ **Then** $\mathsf{map}(A \cup M_G, B)$; $M_G := \emptyset$; i+=1; j+=1;
    Else If $|A| = |B \cup M_S|$ **Then** $\mathsf{map}(A, B \cup M_S)$; $M_S := \emptyset$; i+=1; j+=1;
    Else If $|A| < |B|$ **Then** $M_G := M_G \cup A$; i+=1;
    Else $M_S := M_S \cup B$; j+=1;
return $\mathsf{dc}(G, S)$.
Function $\mathsf{map}(P, Q)$:
    For $k = 1, 2, \ldots, |P|$, set the label of $Q[k]$ to be the label of $P[k]$.
    $F := F \setminus P$, $U := U \setminus Q$.

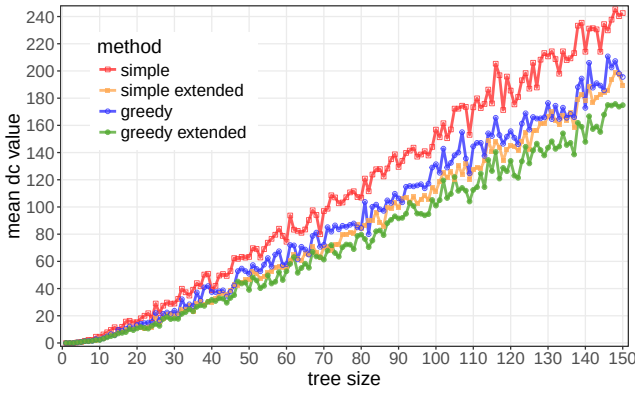


Figure 5: Averaged minimal deep coalescence, computed by all four heuristics.

the lowest cost, we generated random trees from the Yule model [20]. For each tree size between 1 and 150 we generated 7 random tree pairs. Then, we computed the approximation of the minimal cost by using our four algorithms. The result, averaged over sizes of trees, is depicted 5. We observe that the greedy extended algorithm is the best performing among all our algorithms.

Experiment II. In practice, we often have some partial information on the labeling of leaves. Therefore, we introduce a more practical variant of our problems:

Given a gene tree G with a partial labeling, i.e., some leaves of G are unlabeled, and a species tree S. Find the total labeling for G that minimize the deep coalescence cost $\mathsf{dc}(G, S)$.

The greedy algorithm can be easily modified to solve the above problem. In line 5 of Alg. 2, it is sufficient to remove labeled leaves from F and used taxons being labels from U.

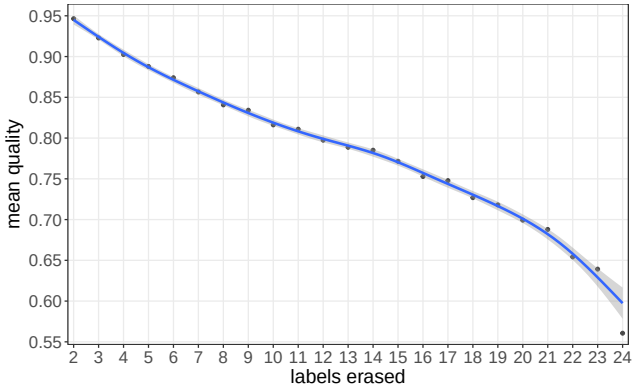


Figure 6: Averaged quality, computed by the modified greedy algorithm on the trees from *TreeFam* dataset [21].

The test was performed on TreeFam dataset, which consists of 1274 curated gene family trees from TreeFam v7.0 [21] spanning 25 mostly animal species. The species tree was based on the NCBI taxonomy. First, we extracted 295 gene trees with bijectively labeled leaves. Next, for each such a gene tree G, we contracted a species tree to the taxons present in G. Hence, we obtained 295 pairs of bijectively labeled trees, with the average size of 17.66 taxons. Finally, for every pair of trees $\langle G, S \rangle$, and for each $i = \{2, 3, \ldots, |G|\}$, we removed labels of i random leaves from G and then applied the modified greedy algorithm to infer total labeling. The quality of a reconstruction is determined by the number of properly reconstructed

labels divided by i. The experiment was repeated 10 times. The result, averaged over i, is depicted in Fig. 6.

5 Conclusion

In this article, we gave a closer look at an open question of the minimal deep coalescence cost for the fixed tree topologies and bijective labelings. The particular cases that we have solved provide a better understanding of the properties of the deep coalescence cost. While the complexity of our problems remains open, the methods presented seem to be a good starting point to the solve the problems in the general case which we plan to study in future. Also, we plan to test our solutions on more complex empirical and simulated datasets, including simulations with more realistic models.

Acknowledgements

The support was provided by NCN grant #2015/19/B/ST6/00726.

References

[1] W. P. Maddison. "Gene Trees in Species Trees". In: *Syst Biol* 46 (1997), pp. 523–536.

[2] R. D. M. Page and E. C. Holmes. *Molecular evolution: a phylogenetic approach*. Blackwell Science, 1998.

[3] L. Zhang. "From Gene Trees to Species Trees II: Species Tree Inference by Minimizing Deep Coalescence Events". In: *IEEE/ACM TCBB* 8 (2011), pp. 1685–1691.

[4] B. C. Carstens and L. L. Knowles. "Estimating species phylogeny from gene-tree probabilities despite incomplete lineage sorting: an example from Melanoplus grasshoppers". In: *Syst Biol* 56.3 (2007), 400–11.

[5] W. Jennings and S. Edwards. "Speciational history of Australian grass finches (Poephila) inferred from thirty gene trees". In: *Evolution* 59.9 (2005), 2033–47.

[6] C. Than and L. Nakhleh. "Species tree inference by minimizing deep coalescences". In: *PLoS Comput Biol* 5.9 (2009), e1000501.

[7] C. V. Than and N. A. Rosenberg. "Consistency Properties of Species Tree Inference by Minimizing Deep Coalescences". In: *J Comput Biol* 18.1 (2011), pp. 1–15.

[8] Y.-C. Wu, M. D. Rasmussen, M. S. Bansal, and M. Kellis. "Most parsimonious reconciliation in the presence of gene duplication, loss, and deep coalescence using labeled coalescent trees". In: *Genome research* 24.3 (2014), pp. 475–486.

[9] R. Chaudhary, J. G. Burleigh, and O. Eulenstein. "Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence". In: *BMC bioinformatics* 13.10 (2012), S11.

[10] P. Górecki and O. Eulenstein. "Deep coalescence Reconciliation with Unrooted Gene Trees: Linear Time Algorithms". In: *LNCS* 7434 (2012), pp. 531–542.

[11] P. Górecki, O. Eulenstein, and J. Tiuryn. "Unrooted Tree Reconciliation: A Unified Approach". In: *IEEE/ACM TCBB* 10.2 (2013), pp. 552–536.

[12] P. Górecki and O. Eulenstein. "Maximizing Deep Coalescence Cost". In: *IEEE/ACM TCBB* 11.1 (2014), pp. 231–242.

[13] C. V. Than and N. A. Rosenberg. "Mathematical properties of the deep coalescence cost". In: *IEEE/ACM TCBB* 10.1 (2013), pp. 61–72.

[14] P. Górecki and O. Eulenstein. "Gene Tree Diameter for Deep Coalescence". In: *IEEE/ACM TCBB* 1 (2015), pp. 155–165.

[15] A. Mykowiecka, P. Szczesny, and P. Gorecki. "Inferring gene-species assignments in the presence of horizontal gene transfer". In: *IEEE/ACM TCBB* (2017).

[16] A. Betkier, P. Szczęsny, and P. Górecki. "Fast Algorithms for Inferring Gene-Species Associations". In: *International Symposium on Bioinformatics Research and Applications*. Springer. 2015, pp. 36–47.

[17] L. Zhang and Y. Cui. "An Efficient Method for DNA-Based Species Assignment via Gene Tree and Species Tree Reconciliation". In: *WABI*. Springer. 2010, pp. 300–311.

[18] P. Górecki and J. Tiuryn. "DLS-trees: A model of evolutionary scenarios". In: *Theor Comput Sci* 359.1-3 (2006), pp. 378–399.

[19] D. M. Campbell and D. Radford. "Tree isomorphism algorithms: speed vs. clarity". In: *Mathematics Magazine* 64.4 (1991), pp. 252–261.

[20] G. U. Yule. "A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS". In: *Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character* 213 (1925), pp. 21–87.

[21] J. Ruan et al. "TreeFam: 2008 Update". In: *Nucleic Acids Res* 36 (2008), pp. D735–40.