Non-binary Tree Reconciliation with Endosymbiotic Gene Transfer

Mathieu Gascon
Département d'informatique et de recherche opérationnelle (DIRO), Université de Montréal, Canada

Nadia El-Mabrouk[1]
DIRO, Université de Montréal, Canada

Abstract

Gene transfer between the mitochondrial and nuclear genome of the same species, called endosymbiotic gene transfer (EGT), is a mechanism which has largely shaped gene contents in eukaryotes since a unique ancestral endosymbiotic event know to be at the origin of all mitochondria. The gene tree-species tree reconciliation model has been recently extended to account for EGTs: given a binary gene tree and a binary species tree, the EndoRex software outputs an optimal DLE-Reconciliation, that is an embedding of the gene tree into the species tree inducing a most parsimonious history of Duplications, Losses and EGT events. Here, we provide the first algorithmic study for DLE-Reconciliation in the case of a multifurcated (non-binary) gene tree. We present a general two-steps method: first, ignoring the mitochondrial-nuclear (or 0-1) labeling of leaves, output a binary resolution minimizing the DL-Reconciliation and, for each resolution, assign a known number of 0s and 1s to the leaves in a way minimizing EGT events. While Step 1 corresponds to the well studied non-binary DL-Reconciliation problem, the complexity of the formal label assignment problem related to Step 2 is unknown. Here, we show it is NP-complete even for a single polytomy (non-binary node). We then provide a heuristic which is exact for the unitary cost of operations, and a polynomial-time algorithm for solving a polytomy in the special case where genes are specific to a single genome (mitochondrial or nuclear) in all but one species.

2012 ACM Subject Classification Applied computing → Molecular evolution

Keywords and phrases Reconciliation, Duplication, Endosymbiotic gene transfer, Multifurcated gene tree, Polytomy

Digital Object Identifier 10.4230/LIPIcs.WABI.2022.5

1 Introduction

Since an initial endosymbiotic event integrating an $\alpha-$proteobacterial genome into an eukaryotic cell, which is known to be at the origin of all extent mitochondria, eukaryote evolution has been marked by episodes of gene transfers, mainly from the mitochondria to the nucleus, resulting in a significant reduction of the mitochondrial genome. Understanding how both nuclear and mitochondrial genomes have been shaped by gene loss, duplication and transfer is important to shed light on a number of open questions regarding the origin, evolution, and characteristics of gene coding capacity of eukaryotes.

From a computational point of view, EndoRex [1] is the first algorithm developed for integrating endosymbiotic events (special cases of gene transfers, but only between the mitochondrial and nuclear genome of the same species) in a gene tree - species tree reconciliation model. Given a gene family with gene copies labeled by 0 or 1 depending on whether they are encoded in the mitochondrial or nuclear genome of a given species, a gene tree for the gene family and a species tree for the considered species, EndoRex infers a most parsimonious scenario of duplications, losses and endosymbiotic gene transfers (EGT)

[1] Corresponding author

explaining the gene tree given the species tree. It is an exact polynomial-time algorithm, which can be used to output all minimum cost solutions, for arbitrary costs of operations.

However, as it has been shown for other evolutionary events [6], the result of a reconciliation model strongly depends on the considered trees. For example, due to potential errors in the trees, some of the plant datasets analysed in [1] produced unrealistic evolutionary histories with unexpected high number of gene duplications and losses. A solution would be to ignore weakly supported parts of the tree, leading to a non-binary tree with multifurcated nodes, also called "polytomies", and simultaneously infer a binary refinement and optimal reconciliation of the multifurcated tree, more precisely, infer an optimal evolutionary scenario leading to a binary refinement of the tree. This strategy has been applied, for example, to infer the evolution of the gene families responsible for alkaloid accumulation in plants [11].

Reconciling a non-binary gene tree by minimizing a DL-Reconciliation cost (history with minimum Duplication/Loss cost) has been considered by many authors [2, 4, 10, 9, 12]. As far as we know, the most efficient algorithm is PolytomySolver [9] which handles unit costs in linear time, improves the best complexity of previous algorithms for the general DL cost model by a linear factor and enables to account for various evolutionary rates across the branches of a species tree, attributing to each taxa its specific duplication and loss cost.

In this paper, we explore the multifurcated gene tree reconciliation problem with a reconciliation model accounting for duplications, losses, but also EGT events. Our method is in two steps: ignoring the 0-1 labeling of the leaves, first output all resolutions minimizing the DL-Reconciliation cost and then, for each resolution (i.e. binary tree), assign a known number of 0s and 1s to the leaves in a way minimizing EGT events. As step one can be done efficiently, we then focus on the second step which consists in assigning an optimal 0-1 labeling for the nodes of a binary tree. We show in Section 3 that this problem is NP-complete, even when the multifurcated tree is restricted to a single polytomy. We then, in Section 4, present a general algorithm solving each polytomy separately, which is shown optimal for a unitary cost of operations.

Except for species conserving the traces of an ancestral eukaryotic origin, few genes are expected to reflect an intermediate endosymbiotic integration of the mitochondrial gene content to the nucleus, with gene copies in both the nuclear and mitochondrial genome. This is the case of the eukaryotes with complete mitochondrial genomes explored in [7] (statistics summarized in [1]): among the 2,486 species, only 52 species have mitochondrial-encoded genes also present in the nuclear genome. This motivates Section 5 where we develop a polynomial-time algorithm for the genome labeling problem in the special case where, in each polytomy, genes are specific to a single genome (mitochondrial or nuclear) in all but one species. We first begin, in the next section, by formally defining our problems.

2 Preliminaries, evolutionary model and definitions

All trees are considered rooted. Given a tree T, we denote by $r(T)$ its root, by $V(T)$ its set of nodes and by $L(T) \subseteq V(T)$ its leafset. A node x is a *descendant* of y if x is on the path from y to a leaf of T and an *ancestor* of y if x is on the path from $r(T)$ to y; x is a *strict descendant* (respect. *strict ancestor*) of x' if it is a descendant (respec. ancestor) of x' different from x'. Moreover, x is the *parent* of $y \neq r(T)$ if it directly precedes y on this path. In this latter case, y is a *child* of x. We denote by $E(T)$ the set of edges of T, where an edge is represented by its two terminal nodes (x, y), with x being the parent of y. More generally, if x is an ancestor of y, (x, y) denotes the path between x and y. The subtree of T rooted at x (i.e. containing all the nodes descendant from x in T) is denoted $T[x]$. The *lowest common*

ancestor (LCA) in T of a subset L' of $L(T)$, denoted $lca_T(L')$, is the ancestor common to all the nodes in L' which is the most distant from the root.

An internal node (a node which is not a leaf) is said to be *unary* if it has a single child, *binary* if it has two children, and a *polytomy* if it has more than two children. We will denote by x_l and x_r the two children of a binary node. The node x_l (respec. x_r) is called *the sibling* of x_r (respec. x_l).

A tree R is *an extension* of a tree T if it is obtained from T by *grafting* unary or binary nodes in T, where grafting a unary node x on an edge (u, v) consists in creating a new node x, removing the edge (u, v) and creating two edges (u, x) and (x, v), and in the case of grafting a binary node, also creating a new leaf y and an edge (x, y). In the latter case, we say that y is a *grafted leaf*.

A *species tree* for a set Σ of species is a tree S with a bijection between $L(S)$ and Σ. In this paper, we assume that the species tree S for a given set of species Σ is known, rooted and binary. A *gene family* is a set Γ of genes where each gene $x \in \Gamma$ belongs to a given species $s(x)$ of Σ. A tree G is a *gene tree* for a gene family Γ if its leafset is in bijection with Γ. We write $\langle G, s\rangle$ when each leaf of G is meant to be fully identified by its *species labeling*, i.e. the species $s(x)$ it belongs to (Figure 1. (3) and (4)). For a subset $G \subseteq \Gamma$ of genes, we write $s(G) = \{s(g) : g \in G\}$ as the set of species containing the genes of G. Then the LCA-mapping of G with S is the function assigning to each node x of G the LCA of $s(V(G[x]))$ in S.

In this paper, we will consider an additional *genome labeling* b for a gene x: $b(x) = 0$ if x belongs to the mitochondrial genome of $s(x)$, and $b(x) = 1$ if x belongs to the nuclear genome of $s(x)$. We write $\langle G, s, b\rangle$ when we want to specify that each leaf of G is fully identified by these two labels (Figure 1. (2) and (5)). To summarize, G, $\langle G, s\rangle$ and $\langle G, s, b\rangle$ are three notations for a gene tree, the two last specifying the way the leaves of G are identified.

A *binary tree* is a tree with all internal nodes being binary. If internal nodes have one or two children, then the tree is said *partially binary*. A *multifurcated tree* is a tree containing at least one polytomy. For example, in Figure 1, the tree (2) is a multifurcated tree with two polytomies.

A *binary refinement* of a multifurcated tree is a binary tree defined as follows.

▶ **Definition 1** (binary refinement). *Let $\langle G, s, b\rangle$ be a multifurcated tree. A binary tree $\langle G^B, s^B, b^B\rangle$ is said to be a* binary refinement *of $\langle G, s, b\rangle$ if $V(G) \subseteq V(G^B)$ and for every $x \in V(G)$, $L(\langle G, s, b\rangle[x]) = L(\langle G^B, s^B, b^B\rangle[x])$. We denote by $\mathcal{B}(\langle G, s, b\rangle)$ the set of binary refinements of $\langle G, s, b\rangle$.*

As for a multifurcated tree $\langle G, s\rangle$, a binary refinement $\langle G^B, s^B\rangle$ and the set of binary refinements $\mathcal{B}(\langle G, s\rangle)$ are defined in the same way, just ignoring the b labeling.

For example, in Figure 1, the tree in (3) and (4) is a binary refinement of $\langle G, s\rangle$ (i.e. the tree in (2) ignoring the 0-1 labeling of leaves), and the tree in (5) is a binary refinement of $\langle G, s, b\rangle$.

We need a final notation. Let $X \subseteq L(\langle G, s, b\rangle)$. The *count matrix* $Count(X)$ for X is a $|\Sigma| \times 2$ matrix defined as follows:

$$\begin{cases} Count(X)[\sigma, 0] = & \text{number of genes } g \in X \ / \ s(g) = \sigma \text{ and } b(g) = 0 \\ Count(X)[\sigma, 1] = & \text{number of genes } g \in X \ / \ s(g) = \sigma \text{ and } b(g) = 1 \end{cases}$$

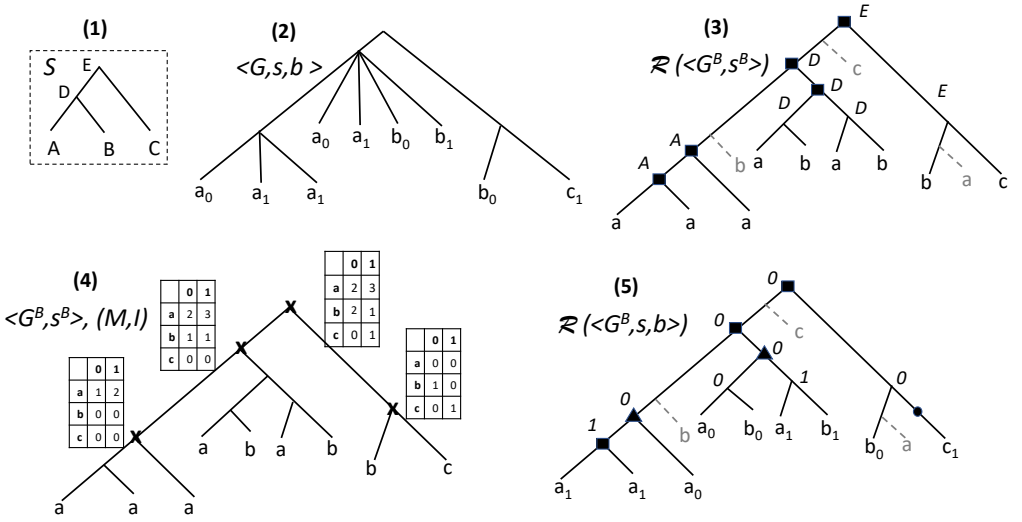


Figure 1 (1) A species tree S on $\Sigma = \{A, B, C\}$; (2) A multifurcated gene tree G where leaves are identified by a species mapping s (a lowercase letter corresponds to the genome identified by the same uppercase letter) and a genome mapping b (the 0-1 index of each leaf); (3) A DL-Reconciliation of the binary refinement $\langle G^B, s^B\rangle$ of $\langle G, s\rangle$. The internal node labeling corresponds to the LCA-mapping with S, squares correspond to duplications and dotted lines to losses (8 events in total). This DL-Reconciliation is optimal for the unitary cost of operations; (4) The tree $\langle G^B, s^B\rangle$ accompanied with an (M, I) b-Constraint, where I is the set of nodes indicated by crosses; (5) A DLE-Reconciliation of $\langle G^B, s^B, b^B\rangle$, where b^B is the genome labeling indicated on leaves, which is consistent with (M, I). The triangle indicates an EGT event and a unary node indicates an EGT-Loss event. For a unitary cost of operations, this reconciliation of cost 9 is optimal for the DLE-BinL problem.

2.1 DLE Reconciliation

Inside the species' genomes, genes undergo *Speciation* (Spe) when the species to which they belong do, but also *Duplication* (Dup) i.e. the creation of a new gene copy, *Loss* of a gene copy, and transfer when a gene is transmitted from a source to a target genome. In this paper, we consider a special case of transfers, called endosymbiotic gene transfers or EGT, only allowing the transmission of genes from the mitochondrial genome to the nuclear genome of the same species, or vice-versa. If the transmission of a gene from a genome A to a genome B is accompanied by the loss of the gene in A, we refer to the event as an *EGT-Loss* ($EGTL$).

Assume that we are given a binary species tree S and a binary gene tree $\langle G, s, b\rangle$. Given an extension R of G, *an extension of* s is a function $\tilde{s}$ from $V(R)$ to $V(S)$ such that, for each leaf x of G, $\tilde{s}(x) = s(x)$. Moreover, an *extension of* b is a function $\tilde{b}$ from $V(R)$ to $\{0, 1\}$ such that, for each leaf x of T, $\tilde{b}(x) = b(x)$. We are now ready to recall the definition of a DLE Reconciliation as introduced in [1].

▶ **Definition 2** (DLE-Reconciliation). *Let $\langle G, s, b\rangle$ be a rooted binary gene tree for a gene family Γ and S be a rooted binary species tree for the species Σ the genes belong to. A* DLE-Reconciliation *of $\langle G, s, b\rangle$ with S (or simply DLE-Reconciliation if no ambiguity) is a quadruplet $\langle R, \tilde{s}, \tilde{b}, e\rangle$ where R is a partially binary extension of G, $\tilde{s}$ is an extension of s, $\tilde{b}$ is an extension of b, and e is an event labeling of the internal nodes of R, such that:*

1. *Each unary node x with a single child y is such that $e(x) = EGTL$, $\tilde{s}(x) = \tilde{s}(y)$ and*

$\tilde{b}(x) \neq \tilde{b}(y)$; *$x$ is an EGT-Loss event with source genome* $\sigma_{\tilde{b}(x)}$ *and target genome* $\sigma_{\tilde{b}(y)}$, *where* $\sigma = \tilde{s}(x)$ *(or equivalently* $\tilde{s}(y)$*).*

2. *For each binary node x of R with two children x_l and x_r, one of the following cases holds:*
   - **a.** $\tilde{s}(x_l)$ *and* $\tilde{s}(x_r)$ *are the two children of* $\tilde{s}(x)$ *in S and* $\tilde{b}(x_l) = \tilde{b}(x_r) = \tilde{b}(x)$, *in which case* $e(x) = Spe$;
   - **b.** $\tilde{s}(x_l) = \tilde{s}(x_r) = \tilde{s}(x) = \sigma$ *and* $\tilde{b}(x_l) = \tilde{b}(x_r) = \tilde{b}(x)$ *in which case* $e(x) = Dup$ *representing a duplication in* $\sigma_{\tilde{b}(x)}$;
   - **c.** $\tilde{s}(x_l) = \tilde{s}(x_r) = \tilde{s}(x) = \sigma$ *and* $\tilde{b}(x_l) \neq \tilde{b}(x_r)$ *in which case* $e(x) = EGT$; *let y be the element of* $\{x_l, x_r\}$ *such that* $\tilde{b}(x) \neq \tilde{b}(y)$, *then* $\tilde{e}(x)$ *is a transfer with source genome* $\sigma_{\tilde{b}(x)}$ *and target genome* $\sigma_{\tilde{b}(y)}$.

Grafted leaves in the extension R correspond to gene losses.

As R is as an extension of G, each node in G has a corresponding node in R. In particular, the $\tilde{s}$, $\tilde{b}$ and e labeling on R induce an $\tilde{s}$, $\tilde{b}$ and e labeling on the nodes of G. The difference between G and R are additional binary nodes with a child being a grafted leaf (a loss), and unary nodes corresponding to EGT-Losses.

A *DL-Reconciliation* of $\langle G, s\rangle$ is defined as in Definition 2, ignoring the binary assignment of genes, i.e. it is a tuple $\langle R, \tilde{s}, e\rangle$ where R is an extension of G.

Optimal reconciliation:

Let c be a function attributing a cost to each event in $DLE = \{Spe, Dup, Loss, EGT, EGTL\}$. As it is usually the case, we will assume a 0 cost for speciations and positive costs for all the other events. Moreover, we assume that $c(Dup) \leq c(EGT) + c(EGTL)$ as otherwise duplications would never be inferred in a most parsimonious reconciliation. Similarly, we assume $c(EGT) \leq c(Dup) + c(EGTL)$ to allow for EGTs and $c(EGTL) \leq c(EGT) + c(Loss)$ to allow for EGT-Losses.

Given a DLE-Reconciliation $\mathcal{R} = \langle R, \tilde{s}, \tilde{b}, e\rangle$ (respec. DL-Reconciliation $\langle R, \tilde{s}, e\rangle$), the cost $C(\mathcal{R})$ of $\mathcal{R}$ is the sum of costs of the events labeling the internal nodes of R plus the sum of costs of the losses, i.e. $C(\mathcal{R}) = \sum_{x \in V(R) \setminus L(R)} c(e(x)) + |L(R)_{Loss}| * c(Loss)$ where $|L(R)_{Loss}|$ is the number of losses in $\mathcal{R}$. In this paper, we seek for a most parsimonious reconciliation, i.e. a reconciliation of minimum cost, also called *optimal reconciliation*. We denote by $DLE(G, S)$ (respec. $DL(G, S)$) the cost of an optimal DLE-Reconciliation (respec. DL-Reconciliation).

From now on, we denote by δ, λ, ρ and τ, respectively, the cost of a duplication, a loss, an EGT-loss and an EGT event. The cost function is said to be *unitary* when $\delta = \lambda = \rho = \tau$.

The following lemma makes the link between an optimal DLE-Reconciliation and the optimal DL-Reconciliation. Notice that such optimal DL-Reconciliation is unique [5].

▶ **Lemma 3.** *An optimal DLE-Reconciliation* $\mathcal{R}_{DLE} = \langle R_{DLE}, \tilde{s}_{DLE}, \tilde{b}_{DLE}, e_{DLE}\rangle$ *of* $\langle G, s, b\rangle$ *can be obtained from the optimal DL-Reconciliation* $\mathcal{R}_{DL} = \langle R_{DL}, \tilde{s}_{DL}, e_{DL}\rangle$ *where* $\mathcal{R}_{DLE}$ *is obtained from* $\mathcal{R}_{DL}$ *by possibly adding unary nodes (corresponding to EGT-loss),* $\tilde{s}_{DLE}$ *is an extension of* $\tilde{s}_{DL}$ *and* e_{DLE} *is obtained from* e_{DL} *by labeling unary nodes as EGT-Losses and possibly converting duplications into EGTs.*

Proof. This result follows from Lemma 2 in [1] proven for a unitary cost of operations, but it is easy to see that the proof of lemma 2 can be generalized to a non unitary cost in the case where $\rho \leq \tau + \lambda$, $\tau \leq \delta + \rho$ and $\delta \leq \tau + \rho$ which, as stated above, are the necessary conditions to ensure that a duplication, EGT and EGT-Loss may be found in an optimal

reconciliation. Note that in [1], *EGTcopy* holds for an *EGT* event and *EGTcut* holds for an *EGTL* event. ◀

From now on, we restrict the discussion to DLE-Reconciliations obtained from a DL-Reconciliation, as stated in Lemma 3. Moreover, given a DLE-Reconciliation $\mathcal{R}_{DLE}$, removing an even number of consecutive EGT-Loss nodes can only lead to a more parsimonious DLE-Reconciliation. Therefore, we assume that a reconciliation does not involve such nodes. This assumption is used in the following definition of a compressed reconciliation, aiming at providing a concise representation of a reconciliation, avoiding to represent losses.

▶ **Definition 4** (Compressed reconciliation)**.** *A* compressed DLE-Reconciliation *of* $\langle G, s, b\rangle$ *is a tuple* $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ *obtained from a DLE-Reconciliation* $\langle R, \tilde{s}, \tilde{b}, e\rangle$ *of* $\langle G, s, b\rangle$*, where* e_V *is simply* e *restricted to the nodes of* G *and* e_E *is a P/A (Presence/Absence) labeling of the edges of* G *corresponding to the unary (EGT-Loss) nodes of* R*, i.e. obtained as follows: Let* G' *be the tree obtained from* R *by removing grafted leaves and their parental nodes (i.e. ignoring losses). For each edge* (x, y) *of* G*, let* x', y' *be the corresponding nodes in* G' *(*G' *differs from* G *only by unary nodes). Then:*

$$e_E(x,y) = \begin{cases} P & \text{if the path } (x',y') \text{ in } G' \text{ contains an unary node} \\ A & \text{if the path } (x',y') \text{ in } G' \text{ contains no unary node} \end{cases}$$

A *compressed DL-Reconciliation* of $\langle G, s\rangle$ is defined similarly, ignoring the binary assignment of genes. For example, in Figure 1, (3) is a DL-Reconciliation of the gene tree in (4) with the species tree S in (1). The compressed DL-Reconciliation is simply that tree $\mathcal{R}(\langle G^B, s^B\rangle)$ where we ignore losses, i.e. dotted lines. Moreover, (5) is a DLE-Reconciliation, and the compressed DLE-Reconciliation is $\mathcal{R}\langle G^B, s^B, b^B\rangle$ where we ignore losses and replace the unary node (EGT-Loss) on the branch leading to c_1 by a label on that branch.

If $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ is a compressed DLE-Reconciliation of an optimal DLE-Reconciliation, then it follows from Lemma 3 that $\tilde{s}$ is the LCA-mapping of G with S. Therefore, from now, we only consider compressed DLE-Reconciliations with $\tilde{s}$ being the LCA-mapping.

For a compressed DLE-Reconciliation $\mathcal{R}^c = \langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ of $\langle G, s, b\rangle$, denote by $|e_{V_{EGT}}|$ the number of EGT nodes, by $|e_E|$ the number of edges labeled P, i.e. the number of EGT-Loss events, and define the cost of $\mathcal{R}^c$ as $C(\mathcal{R}^c) = DL(G, S) + |e_{V_{EGT}}| * (\tau - \delta) + |e_E| * \rho$.

▶ **Lemma 5.** *From a* compressed DLE-Reconciliation $\mathcal{R}^c = \langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ *for* $\langle G, s, b\rangle$*, we can obtain a DLE-Reconciliation* $\mathcal{R}$ *of* $\langle G, s, b\rangle$ *of cost* $C(\mathcal{R}) = C(\mathcal{R}^c)$*.*

Proof. Let $\mathcal{R}^c = \langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ be a compressed DLE-Reconciliation for $\langle G, s, b\rangle$.

Let $\mathcal{R}_{DL} = \langle R_{DL}, \tilde{s}, e_{DL}\rangle$ be the optimal DL-Reconciliation of G with S. We construct a DLE-Reconciliation $\mathcal{R} = \langle R_{DLE}, \tilde{s}_{DLE}, \tilde{b}_{DLE}, e_{DLE}\rangle$ from $\mathcal{R}_{DL}$ and $\mathcal{R}^c$ as follows:

- R_{DLE} is obtained from R_{DL} by grafting a unary node (EGT-Loss) on the edge $(parent(x), x)$ (in R_{DL}) for each node $x \in V(R_{DL}) \cap V(G)$ such that $e_E(parent(x), x) = P$ (in G).
- $\tilde{s}_{DLE}(x)$ is the LCA-mapping of R_{DLE} with S.
- $e_{DLE}(x) = e_{DL}(x)$ for each node $x \in V(R_{DL}) \cap V(R_{DLE})$ and $e_{DLE}(x) = EGTL$ for each unary node of R_{DLE}. For each node $x \in V(G) \cap V(R_{DLE})$, if $e_V(x) = EGT$ then we set $e_{DLE}(x) = EGT$.
- $\tilde{b}_{DLE}(x) = \tilde{b}(x)$ for each node $x \in V(R_{DLE}) \cap V(G)$. For each node $x \in V(R_{DLE}) \setminus V(G)$, let y be the lowest ancestor of x such that $y \in V(R_{DLE}) \cap V(G)$. If y is not an EGT node, then set $\tilde{b}_{DLE}(x) = \tilde{b}(y)$ if there is no EGT-loss event in the path (y, x) (in R_{DLE}),

and set $\tilde{b}_{DLE}(x) = 1 - \tilde{b}(y)$ otherwise. Else if y is an EGT node, set $\tilde{b}_{DLE}(x) = \tilde{b}(y)$ if the EGT node y does not transfer in the direction of x and $\tilde{b}_{DLE}(x) = 1 - \tilde{b}(y)$ otherwise.

As $\mathcal{R}$ is constructed from $\mathcal{R}_{DL}$, it is easy to see that the species labeling of the nodes of R_{DLE} is correct. By construction, the genome labeling of the nodes of R_{DLE} is also correct, as the genome labeling $\tilde{b}$ is assumed correct (thus the genome labeling of the nodes $x \in V(R_{DLE}) \cap V(G)$ is correct) and the genome labeling of the nodes $x \in V(R_{DLE}) \setminus V(G)$ is set according to the definition.

Notice that there are $|e_E|$ EGT-loss events and $|e_{V_{EGT}}|$ EGT events in $\mathcal{R}$. Also, the number of loss events in $\mathcal{R}$ is the same as the number of loss events in $\mathcal{R}_{DL}$. Let $|e_{DL_{Dup}}|$ be the number of duplication nodes in the DL-Reconciliation. As an EGT event in $\mathcal{R}$ may only occur on a node that is a duplication in R_{DL}, there are $|e_{DL_{Dup}}| - |e_{V_{EGT}}|$ duplication events in $\mathcal{R}$. Therefore, the cost of $\mathcal{R}$ is: $C(\mathcal{R}) = DL(G,S) + |e_{V_{EGT}}| * (\tau - \delta) + |e_E| * \rho$ ◀

▶ **Corollary 6.** *From an optimal* compressed DLE-Reconciliation $\mathcal{R}^c = \langle G, \tilde{s}, \tilde{b}, e_V, e_E \rangle$*, an optimal DLE-Reconciliation* $\mathcal{R}$ *of* $\langle G, s, b \rangle$ *can be obtained in linear time.*

Proof. For a compressed DLE-Reconciliation $\mathcal{R}^c = \langle G, \tilde{s}, \tilde{b}, e_V, e_E \rangle$, a DLE-Reconciliation leading to $\mathcal{R}^c$, of the same cost as $\mathcal{R}^c$, can be found in linear-time by the constructive proof of Lemma 5. In particular, a DLE-Reconciliation $\mathcal{R}$ can be obtained from an optimal compressed DLE-Reconciliation $\mathcal{R}^c$, and this DLE-Reconciliation $\mathcal{R}$ is necessarily optimal. In fact, from Lemma 3, there is an optimal DLE-Reconciliation $\mathcal{R}_{DLE}$ obtained from the optimal DL-Reconciliation. Then, by construction of $\mathcal{R}_{DLE}$, $C(\mathcal{R}_{DLE}) = DL(G,S) + |e_{V_{EGT}}| * (\tau - \delta) + |e_E| * \rho$, which is also the cost of its compressed DLE-Reconciliation $\mathcal{R}^c_{DLE}$. But as $\mathcal{R}^c$ is optimal, $C(\mathcal{R}^c) \leq C(\mathcal{R}^c_{DLE})$, and thus $C(\mathcal{R}) \leq C(\mathcal{R}_{DLE})$, but as $\mathcal{R}_{DLE}$ is by definition an optimal DLE-Reconciliation, we have $C(\mathcal{R}) = C(\mathcal{R}_{DLE})$ and thus $\mathcal{R}$ is also optimal. ◀

The problem of finding an optimal DLE-Reconciliation is thus equivalent to that of finding an optimal compressed DLE-Reconciliation. From now on, we only consider compressed reconciliations and, for brevity, simply call them reconciliations.

2.2 Problem statements

The general problem of simultaneously refining and reconciling a multifurcated gene tree under the DLE evolutionary model is formulated as follows.

DLE Non-binary Reconciliation problem:
Input: A binary species tree S, a non-binary gene tree $\langle G, s, b \rangle$ and a cost function c on DLE.
Output: An optimal DLE-Reconciliation $\langle G', \tilde{s}', \tilde{b}', e_V, e_E \rangle$ of $\langle G, s, b \rangle$ where $\langle G', s', b' \rangle \in \mathcal{B}(\langle G, s, b \rangle)$.

The **DL Non-binary Reconciliation problem** is simply the restriction of the previous problem to DL-Reconciliation, namely given a non-binary gene tree $\langle G, s \rangle$, the problem is to find a minimum cost DL-Reconciliation $\langle R, \tilde{s}, e \rangle$ of a binary refinement of G. Notice that in this case, R is a binary rather than partially binary tree, as unary nodes only correspond to EGT-Loss events which are not considered in a DL-Reconciliation.

In this paper, we explore a resolution of the DLE Non-binary Reconciliation problem operating in two steps:

Resolution method:
Step 1: Find a binary refinement $\langle G^B, s^B\rangle$ of $\langle G, s\rangle$ leading to an optimal DL-Reconciliation $\langle G^B, \tilde{s}^B, e\rangle$ by solving the DL Non-binary Reconciliation problem.
Step 2: Given $\langle G^B, s^B\rangle$ obtained above, find a genome labeling b^B such that $\langle G^B, s^B, b^B\rangle$ is a binary refinement of $\langle G, s, b\rangle$, leading to an optimal DLE-Reconciliation $\langle G^B, \tilde{s}^{B'}, \tilde{b}^B, e'_V, e_E\rangle$.

Although not guaranteed to be optimal, this method is a natural greedy heuristic for the DLE Non-binary Reconciliation problem. In fact, as stated in Lemma 3, an optimal DLE binary reconciliation (result of Step 2) can be obtained from a DL binary reconciliation (result of Step 1) by simply converting some duplication nodes into EGT nodes and adding EGT-Loss labels on branches. Moreover, Step 1 which consists in solving the DL non-binary Reconciliation problem, can be done efficiently [9]. Having a binary refinement $\langle G^B, s^B\rangle$ of $\langle G, s\rangle$ leading to an optimal DL-Reconciliation, the problem then reduces (Step 2) to finding a genome labeling for G^B allowing for an optimal DLE-Reconciliation which, in the case of a unitary cost, is equivalent to finding a minimum number of added EGT-Loss events.

Notice however that, in contrast to the species labeling s^B, the genome labeling b^B of the leaves of G^B is unknown after Step 1. The problem is therefore not reduced to generalizing b^B to the internal nodes (extending b^B to $\tilde{b}^B$), but consists in finding an appropriate labeling b^B of the leaves as well. Although unknown, this genome labeling of $V(G^B)$ is constrained by the genome labeling of $V(G)$, as formulated in the next lemma which is directly deduced from the definition of a binary refinement (Definition 1).

▶ **Lemma 7.** *Let $\langle G, s, b\rangle$ be a multifurcated tree and $\langle G^B, s^B\rangle$ be a binary refinement of $\langle G, s\rangle$. Then $\langle G^B, s^B, b^B\rangle$ is a binary refinement of $\langle G, s, b\rangle$ if and only if, for any node x of G^B with a corresponding node (also denoted x) in G, $Count(L(\langle G, s, b\rangle[x])) = Count(L(\langle G^B, s^B, b^B\rangle[x]))$.*

Therefore, in addition to $\langle G^B, s^B\rangle$ corresponding to a binary refinement of $\langle G, s\rangle$, the input of Step 2 also includes a set of constraints induced by the genome labeling of $V(G)$. These constraints can be represented as a set of $|\Sigma| \times 2$ matrices $M(x)$ for each $x \in I$, where I is the subset of $V(G^B) \setminus L(G^B)$ with corresponding nodes in $V(G)$. (M, I) is called the b-Constraint for G^B (Figure 1. (4)).

▶ **Definition 8.** *Given a binary tree $\langle G^B, s^B\rangle$ and a b-Constraint labeling (M, I) for G^B, a labeling b^B is said to be* consistent with *(M, I) if, for any $x \in I$, $Count(L(\langle G^B, s^B, b^B\rangle[x]) = M(x)$.*

The main problem (Step 2) can thus be defined as follows (for simplicity, we avoid the "^{B}" notation). See an example in Figure 1 where (4) is the input of the DLE-BinL problem and its output is (5).

DLE Binary Labeling (or DLE-BinL) Problem:
Input: A binary tree $\langle G, s\rangle$, a b-Constraint (M, I) and a species tree S;
Output: An optimal DLE-Reconciliation $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ of $\langle G, s, b\rangle$ where b is a genome labeling consistent with (M, I).

We call DLE-BinLR the DLE-BinL problem where I is restricted to the root of G (which corresponds to considering a single polytomy in the initial multifurcated tree).

3 Complexity of the DLE-BinL and DLE-BinLR Problems

We show that the decision versions of both DLE-BinL and DLE-BinLR are NP-complete. In this section, the considered cost is unitary. The DLE-BinL problem in its decision version is defined bellow and DVDLE-BinLR is simply DVDLE-BinL when $I = \{r(G)\}$.

Decision version DLE Binary Labeling (or DVDLE-BinL) Problem:
Input: A binary tree $\langle G, s\rangle$, a b-Constraint (M, I), a species tree S and an integer $Cost$;
Question: Is there a DLE-Reconciliation $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ of $\langle G, s, b\rangle$ where b is a genome labeling consistent with (M, I) for which $C(\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle) \leq Cost$?

We first show, by reducing from the Monotone not-all-equal 3-satisfiability problem (Monotone NAE3SAT Problem), that the DVDLE-BinLR problem is NP-complete. First observe that the DVDLE-BinLR problem is in NP. In fact, given a DLE-Reconciliation $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ of $\langle G, s, b\rangle$, we can calculate the cost of the DLE-Reconciliation (to verify if it is less than or equal to $Cost$) and verify if the genome labeling b is consistent with (M, I) in polynomial time by traversing the tree G.

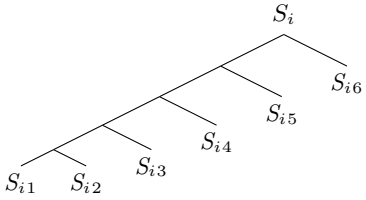
The Monotone NAE3SAT problem is the following (monotone meaning that there are no negation of variables in the clauses)

Monotone NAE3SAT:
Instance: A set of clauses $\mathcal{C} = (C_1 \wedge C_2 \wedge \cdots \wedge C_k)$ on a finite set $Ł = \{\ell_1, \ell_2, \ldots, \ell_m\}$ of variables where each C_i, $1 \leq i \leq k$, is a clause of the form $(x \vee y \vee z)$ with $\{x, y, z\} \subseteq Ł$;
Question: Is there a truth assignment satisfying $\mathcal{C}$ such that the values in each clause are not all equal to each other?

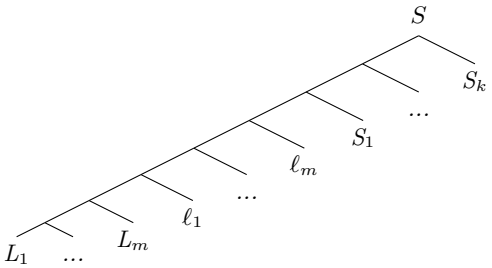
Given an instance $\mathcal{I} = (\mathcal{C}, Ł)$ of the Monotone NAE3SAT problem, we compute, in polynomial time, a corresponding instance $\mathcal{I}' = (\langle G, s\rangle, (M, I), S, Cost)$ of DVDLE-BinLR. First, the set of species Σ is computed as follows:

- For $1 \leq j \leq m$, Σ contains a species ℓ_j and for each clause $C_i \in \mathcal{C}$, $1 \leq i \leq k$ such that ℓ_j is in C_i, Σ contains a species ℓ_{j_i}.
- For each clause $C_i \in \mathcal{C}$, $1 \leq i \leq k$, Σ contains the species S_{i1}, S_{i2}, S_{i3}, S_{i4}, S_{i5} and S_{i6}.

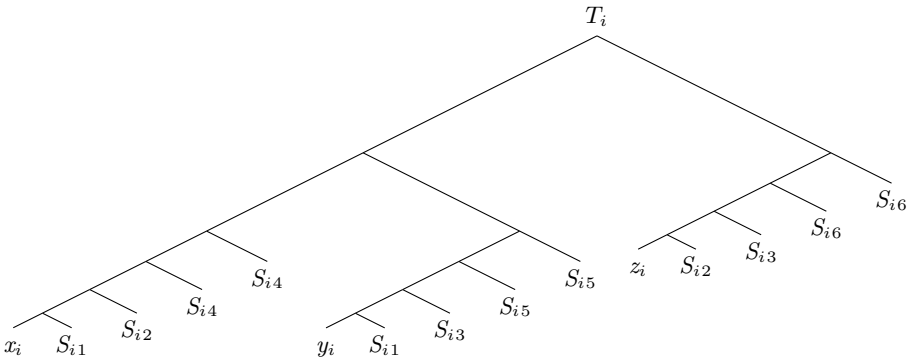
For $1 \leq j \leq m$, let L_j be a caterpillar tree on the leaves ℓ_{j_i} for all i such that ℓ_j is in the clause C_i. For $1 \leq i \leq k$, let S_i be the tree computed as follows:



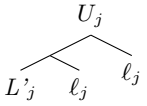
Then, the species tree S is:



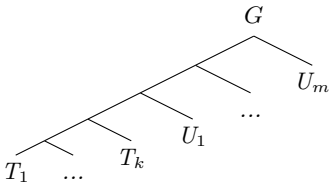
Let now turn to defining the gene tree. For each clause $C_i = (x \vee y \vee z) \in \mathcal{C}$, $1 \leq i \leq k$, let T_i be the following tree:



For $1 \leq j \leq m$, let L'_j be a gene tree which is species label isomorphic to L_j. For $1 \leq j \leq m$, let U_j be the tree computed as follows:



The gene tree G is then:



Notice that for each species $s \in \Sigma$, G contains 2 leaves f such that $s(f) = s$.
We set $M(r(G))$ equal to a matrix of ones of size $|\Sigma| \times 2$. Also recall that $I = \{r(G)\}$. Finally, *Cost* is set to $DL(G, S) + k$.

We next show that $\mathcal{I}$ is a satisfiable instance of the Monotone NAE3SAT problem if (Lemma 9) and only if (Lemma 11) its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR admits a DLE-Reconciliation of cost lower than or equal to *Cost*.

▶ **Lemma 9.** *Let $\mathcal{I}$ be a satisfiable instance of the Monotone NAE3SAT problem. Then its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR admits a DLE-Reconciliation of cost lower than or equal to Cost.*

Proof. See Appendix. ◀

Let $\mathcal{R}'(\langle G, s\rangle, (M, I), S)$ be the optimal DLE-Reconciliation of $\langle G, s, b\rangle$ (with b being a genome labeling consistent with (M, I)) obtained from the optimal DL-Reconciliation of G with S by converting some duplication events into EGT events and by adding some EGT-loss events (this DLE-Reconciliation exists by Lemma 3).

▶ **Lemma 10.** *Let $\mathcal{I}$ be an instance of the Monotone NAE3SAT problem. For its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR, the optimal DLE-Reconciliation $\mathcal{R}'(\langle G, s\rangle, (M, I), S)$ is such that there is at least 1 EGT-loss event in each subtree T_i of G (i.e. $e_E(x, y) = P$ for an edge (x, y) of T_i) for $1 \leq i \leq k$.*

Proof. See Appendix. ◀

▶ **Lemma 11.** *Let $\mathcal{I}$ be an unsatisfiable instance of the Monotone NAE3SAT problem. Then its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR does not admit a DLE-Reconciliation of cost equal or lower than Cost.*

Proof. See Appendix. ◀

Note that, by construction, the instance of DVDLE-BinLR in the reduction contains a gene tree with no more than two leaves having the same species label. From this remark, and since Monotone NAE3SAT is NP-complete, lemmas 9 and 11 lead to the following result.

▶ **Theorem 12.** *The DVDLE-BinLR Problem is NP-complete, even if each species label is present at most 2 times in the leaves of the gene tree G.*

As an instance of the DVDLE-BinLR problem is also an instance of the DVDLE-BinL problem, we conclude that the DVDLE-BinL problem is also NP-complete (we can easily show that it is in NP in the same way we showed that the DVDLE-BinLR Problem is in NP).

▶ **Corollary 13.** *The DVDLE-BinL Problem is NP-complete, even if each species label is present at most 2 times in the leaves of the gene tree G.*

4 A general algorithm for the DLE-BinL Problem

A natural heuristic for the DLE-BinL problem for $\langle G, s\rangle$, where G is a binary resolution of an initial multifurcated tree where initial polytomies are reflected by a b-Constraint (M, I), would be to solve each polytomy, i.e. each subtree rooted at a node x of I, individually. In fact, this strategy leads to an exact algorithm for the DL Non-binary Reconciliation problem. However, in the case of DLE-Reconciliation, the $\tilde{b}$ labeling of internal nodes introduces a dependency between polytomies, avoiding the heuristic to be exact in general, i.e. for an arbitrary cost of operations. In this section, we present the general heuristic (Algorithm 1) and show that it is exact in the case of a unitary cost of operations.

Algorithm 1 traverses the tree G in post-order and each time it encounters a node $x \in I$, it "solves" the corresponding subtree $G[x]$ and replaces it by a single leaf, genome labeled appropriately. Once the tree G has been completely traversed, the subtrees are put back in the tree. Notice that on line 13, the algorithm adds a new species to Σ, but does not extend the species labeling $\tilde{s}$ to this new species. The reason is that the new added species is eventually removed from the tree (line 34), i.e. does not remain in the returned reconciliation. Also notice that on line 9, the algorithm adds a new leaf without genome label for which the algorithm will not consider the genome label at any point in the execution. That leaf is also eventually removed from the tree (line 36).

Algorithm 1 calls a function $DLEBinLR(\langle G, s\rangle[x], M(x), S, Bin)$ where $Bin \in \{0, 1\}$, returning an optimal solution of the DLE-BinLR problem such that $\tilde{b}(x) = Bin$. Recall that the DLE-BinLR problem is also NP-complete. In the next section, we will present $DLEBinLR1Species$ which can be substituted to $DLEBinLR$ in Algorithm 1 for a restriction of the problem, where, for each polytomy, genes are b-labeled identically in all but one species.

▶ **Theorem 14.** *Let $\langle G, s\rangle$ be a binary tree, (M, I) be a b-Constraint for $\langle G, s\rangle$, S be a species tree and c be the unitary cost. Then, with the input $(\langle G, s\rangle, (M, I), S)$, Algorithm 1 returns an optimal DLE-Reconciliation of $\langle G, s, b\rangle$ where b is a genome labeling consistent with (M, I).*

Proof. See Appendix. ◀

Algorithm 1 $DLEBinL(\langle G, s\rangle, (M, I), S)$

```
1  i ← 0
2  for each node x of V(G) \ r(G) in a post-order traversal do
3      M̃(x) ← a zero matrix of size |Σ| × 2
4      if x ∈ I then
5          M'(x) ← M(x) − M̃(x_l) − M̃(x_r)
6          G_{j_0} ← DLEBinLR(⟨G, s⟩[x], M'(x), S, 0)
7          G_{j_1} ← DLEBinLR(⟨G, s⟩[x], M'(x), S, 1)
8          if C(G_{j_0}) == C(G_{j_1}) then
9              Replace the subtree ⟨G, s⟩[x] in G by a new leaf ℓ_i without genome label
10         else
11             label ← arg min_{p∈{0,1}}(C(G_{j_p}));
12             Replace the subtree ⟨G, s⟩[x] in G by a new leaf ℓ_i with s(ℓ_i) ← S_i (where
                S_i is a new species) and b(ℓ_i) ← label;
13             Add the species S_i to Σ;
14             for all x' ∈ I such that x' is a strict ancestor of x do
15                 Add the line [1 − b(ℓ_i), b(ℓ_i)] (corresponding to S_i) to M(x')
16             end
17             for all x'' ∈ I such that x'' is not a strict ancestor of x do
18                 Add the line [0,0] (corresponding to S_i) to M(x'')
19             end
20             if the sibling of x is not in I and has already been visited then
21                 Add the line [0,0] (corresponding to S_i) to M̃(sibling(x))
22             end
23         end
24         M̃(ℓ_i) ← M(x)
25         i ← i + 1
26     else if x is an internal node then
27         M̃(x) ← M̃(x_l) + M̃(x_r)
28     end
29 end
30 M'(r(G)) ← M(r(G)) − M̃(r(G)_l) − M̃(r(G)_r)
31 G ← optimal solution of DLEBinLR(⟨G, s⟩, M'(r(G)), S)
32 for j = i - 1 to 0 do
33     if there is a leaf labeled ℓ_j with a genome label in G then
34         Replace the leaf ℓ_j in G by the tree G_{j_k} where k = b(ℓ_j)
35     else
36         Replace the leaf ℓ_j in G by the tree G_{j_k} where k = b(parent(ℓ_j))
37     end
38 end
39 return ⟨G, s̃, b̃, e_V, e_E⟩
```

5 An exact algorithm for the one-species version of the DLE-BinLR Problem

We consider a restriction of the DLE-BinLR problem where genes are specific to a single genome (the mitochondrial or nuclear genome) in all but one species. In its simplest version

where a single species is present, the problem reduces to assigning a multiset of two labels (a given number of 0s and a given number of 1s) to the leaves of a tree-shape (i.e. a tree with no leaf labels), in a way minimizing 0-1 transitions in the tree. Similar problems on assigning leaves to tree-shapes or to multilabeled trees (MUL-trees) have been considered in the context of other tree distances (Robinson Foulds distance, path distance, maximum agreement subtree), most of them being NP-complete [3, 8]. Here we present an exact polynomial-time algorithm for this restricted version of the DLE-BinLR problem, which we call the DLE-BinLR1Species problem.

Let $\sigma \in \Sigma$ be the only species for which the genes belonging to it are not specific to a single genome. We will call the leaves $\ell \in L(G)$ for which $s(\ell) = \sigma$ *free leaves* and the leaves $\ell \in L(G)$ for which $s(\ell) \neq \sigma$ *fixed leaves*. For a fixed leaf ℓ, $b(\ell)$ is fixed and known in advance, as all leaves whose species label is $s(\ell)$ have the same genome label which is known from the matrix M. The DLE-BinLR1Species problem is then reduced to finding an optimal DLE-Reconciliation for which exactly k free leaves are labeled by 0, where $k = M(r(G))[\sigma, 0]$.

Let $\mathcal{R}_{DL} = \langle G, \tilde{s}, e\rangle$ be an optimal DL-Reconciliation for $\langle G, s\rangle$. From Lemma 3, an optimal DLE-Reconciliation $\mathcal{R}_{DLE} = \langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$ with exactly k free leaves labeled by 0 can be obtained from $\mathcal{R}_{DL}$ by converting some duplications into EGTs and adding EGT-Loss events, i.e. a P/A labeling on edges. We define $minCostTransfer(\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle) = |e_{V_{EGT}}| * (\tau - \delta) + |e_E| * \rho$. Then recall from Section 2 that, by construction of $\mathcal{R}_{DLE}$, we have:

$$C(\mathcal{R}_{DLE}) = DL(G, S) + minCostTransfer(\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle)$$

The problem thus reduces to minimizing $minCostTransfer(\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle)$.

We will need to consider the two possible genome labeling $i \in \{0, 1\}$ for the root of G. We therefore denote by $minCostTransfer(\langle G, \tilde{s}, e\rangle, k, i)$ the $minCostTransfer$ function for an optimal DLE-Reconciliation $\mathcal{R}_{DLE}$ with exactly k free leaves labeled by 0 and with the additional constraint that $\tilde{b}(r(G)) = i$.

We are now ready to present Algorithm 2. It proceeds in two phases: (1) a bottom-up phase (Algorithm 3) in which we assign an array of size $2 \times (k + 1)$ to each node x of G where the (i, j)th entry equals $minCostTransfer(\langle G[x], \tilde{s}, e\rangle, j, i)$ where $\langle G[x], \tilde{s}, e\rangle$ is the optimal DL-Reconciliation of $G[x]$ with S; (2) a top-down phase (not given in pseudo-code) in which the algorithm assigns the $\tilde{b}$ labeling of nodes and locates the EGT and EGT-Loss events in the optimal solution. For this purpose, for each entry of $x.array$ of each internal node x, the bottom-up algorithm keeps in memory pointers to the entries of the arrays of the children of x from which the value of the entry was obtained.

▶ **Theorem 15.** *The output of Algorithm 2 is a solution of the DLE-BinLR1Species problem.*

Proof. See Appendix. ◀

▶ **Theorem 16.** *Algorithm 2 computes the solution of the DLE-BinLR1Species problem in $O(nk^2)$ time, where n is the number of leaves of G.*

Proof. For each leaves of G, the associated array is computed in time $O(k)$. For each internal node of G, the associated array is computed in time $O(k^2)$. The time complexity to compute the arrays for all the nodes is then $O(nk^2)$.

Once all the arrays are computed, the algorithm finds the optimal assignation of the internal nodes with a preorder traversal of G in time $O(n)$

We conclude that the time complexity of Algorithm 2 is $O(nk^2)$. ◀

Algorithm 2 $BinLR1Species(\langle G, s\rangle, (M, I), S)$

$k \leftarrow M(r(G))[\sigma, 0]$
$\langle G, \tilde{s}_{DL}, e_{DL}\rangle \leftarrow$ Optimal DL-Reconciliation of $\langle G, s\rangle$ with S
$Bottom - up(\langle G, s\rangle, e_{DL}, k)$
$Top - down(\langle G, s\rangle, e_{DL}, k)$
return $\langle G, \tilde{s}, \tilde{b}, e_V, e_E\rangle$

Algorithm 3 $Bottom - up(\langle G, s\rangle, e, k)$

for *each node* x *of* G *in a post-order traversal* **do**
  $x.array \leftarrow$ Array of size $2 \times (k+1)$
  if x *is a leaf* **then**
    if x *is fixed to "0"* **then**
      $x.array(0,0) \leftarrow 0$
      $x.array(i,j) \leftarrow \infty$ for every values of $(i,j) \neq (0,0)$
    else if x *is fixed to "1"* **then**
      $x.array(1,0) \leftarrow 0$
      $x.array(i,j) \leftarrow \infty$ for every values of $(i,j) \neq (1,0)$
    // Case where x is a free leaf
    else
      $x.array(0,1) \leftarrow 0$
      $x.array(1,0) \leftarrow 0$
      $x.array(i,j) \leftarrow \infty$ for every values of $(i,j) \neq (1,0)$ and $(i,j) \neq (0,1)$
    end
  else
    for $j = 0$ *to* k **do**
      $T_{00}, T_{01}, T_{10}, T_{11} \leftarrow$ Arrays of size $(j+1)$
      for $\ell = 0$ *to* j **do**
        // ℓ is the number of free leaves labeled "0" under x_l and $j-\ell$ is the number of free leaves labeled "0" under x_r
        $T_{00}(\ell) \leftarrow x_l.array(0,\ell) + x_r.array(0, j-\ell)$
        $T_{01}(\ell) \leftarrow x_l.array(0,\ell) + x_r.array(1, j-\ell)$
        $T_{10}(\ell) \leftarrow x_l.array(1,\ell) + x_r.array(0, j-\ell)$
        $T_{11}(\ell) \leftarrow x_l.array(1,\ell) + x_r.array(1, j-\ell)$
      end
      // Cost of the first transfer
      $cost \leftarrow ((e(x) == Dup)\ ?\ \tau - \delta : \rho)$
      // Case where x is labeled "0"
      $x.array(0,j) \leftarrow$
       $\min(\min(T_{00}), cost + \min(T_{01}), cost + \min(T_{10}), cost + \rho + \min(T_{11}))$
      // Case where x is labeled "1"
      $x.array(1,j) \leftarrow$
       $\min(cost + \rho + \min(T_{00}), cost + \min(T_{01}), cost + \min(T_{10}), \min(T_{11}))$
    end
  end
end

See Figure 2 for an example of an execution of Algorithm 2.

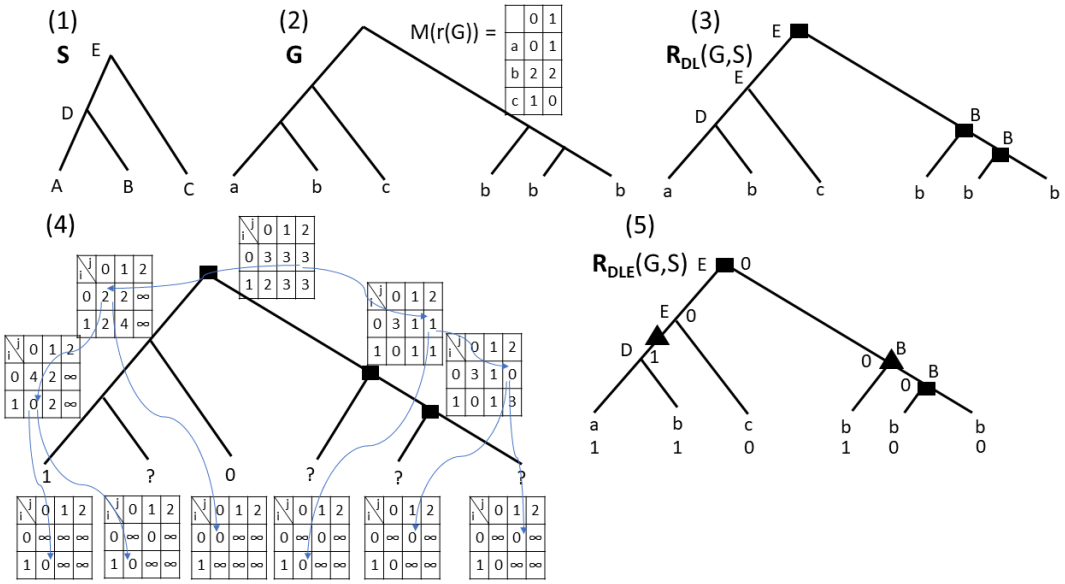


Figure 2 (1) A species tree S on $\Sigma = \{A, B, C\}$; (2) A binary gene tree G where leaves are identified by a species mapping s, and a b-Constraint (M, I) where $I = r(G)$; (3) An optimal DL-Reconciliation of G with S; (4) The tree G accompanied with the arrays computed by Algorithm 3 (we consider here the costs $\delta = \lambda = 1$ and $\rho = \tau = 2$) and the pointers for an optimal solution; (5) The optimal DLE-Reconciliation $\mathcal{R}_{DLE}(G, S)$ of $\langle G, s, b\rangle$ (where b is consistent with (M, I)) returned by Algorithm 2. The cost $minCostTransfer(\mathcal{R}_{DLE}(G, S))$ is 3. Events are represented as in Figure 1.

6 Conclusion

Endosymbiotic gene transfers (EGTs) are important events to be considered in a reconciliation model aiming to infer the evolution of a gene family, given a gene tree for the gene family and a species tree for the species containing the genes. As it is usually difficult, or impossible, to infer a well supported binary tree based on sequence data, it is also important to be able to account for non-binary gene trees. In this paper, we present the first method for DLE reconciliation, i.e. reconciliation accounting for duplications, losses, but also EGTs, for a multifurcated gene tree. It is a natural extension of the DL reconciliation of a multifurcated tree, where we first consider a solution for this problem, i.e. an optimal DL-Reconciliation, which can be obtained efficiently, and then appropriately assign the binary genome labeling (0/1 for mitochondrial/nuclear) to the nodes of the tree, accounting for EGT transfers, in a way minimizing a total DLE (Duplications, Losses and EGT) cost.

We show that the optimal genome labeling assignment step is NP-complete for an arbitrary binary refinement, even for a single polytomy, and even when genes are present in only two copies in each species. We then present two natural heuristics for the general and one-polytomy versions of the problem which are shown to be exact for some restrictions on the model (unitary cost of operations and/or free leaves belonging to a single genome). As explained in the introduction, we argue that these restrictions are biologically relevant. The next step will be to apply our method to the orthologous mitochondrial protein-coding genes (MitoCOGs) dataset [1, 7].

From a theoretical and algorithmic point of view, which is the focus of this paper, many open questions remain. Apart from the fact that a heuristic combining accuracy and time-

efficiency should be developed for both the DLE-BinL and DLE-BinLR problems in the case of a general cost function and an arbitrary number of species presenting an intermediate endosymbiotic integration, a more fundamental question is whether an exact one-step method, considering all the events at once, can be developed. In fact, the complexity results obtained here do not allow to conclude on the complexity of the DLE Non-binary Reconciliation problem. It is indeed not excluded that the polynomial-time PolytomySolver algorithm [9] can be extended for solving a multifurcated tree with a binary labeling of leaves, at least in special cases. In the near future, we will first explore the extension of PolytomySolver to the one species restriction of the model, before considering generalization to an arbitrary number of species.

References

1 Y. Anselmetti, N. El-Mabrouk, M. Lafond, and A. Ouangraoua. Gene tree and species tree reconciliation with endosymbiotic gene transfer. *Bioinformatics*, 37(SI-1):i120–i132, 2021.

2 W.C. Chang and O. Eulenstein. Reconciling gene trees with apparent polytomies. In D.Z. Chen and D. T. Lee, editors, *Proceedings of the 12th Conference on Computing and Combinatorics (COCOON)*, volume 4112 of *Lecture Notes in Computer Science*, pages 235–244, 2006.

3 C. Colijn and G. Plazzotta. A metric on phylogenetic tree shapes. *Systematic Biology*, 67(1):113–126, 2018.

4 D. Durand, B.V. Haldórsson, and B. Vernot. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, 13:320–335, 2006.

5 N. El-Mabrouk and E. Noutahi. *Bioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret*, chapter Gene Family Evolution—An Algorithmic Framework, pages 87–119. Springer International Publishing, t. warnow edition, 2019.

6 M.W. Hahn. Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. *Genome Biology*, 8(7):R141, 2007.

7 S. Kannan, I. Rogozin, and E. Koonin. MitoCOGs: clusters of orthologous genes from mitochondria and implications for the evolution of eukaryotes. *BMC Evolutionary Biology*, 14(11):1–16, 2014.

8 M. Lafond, N. El-Mabrouk, K.T. Huber, and V. Moulton. The complexity of comparing mutiply-labelled trees by extending phylogenetic-tree metrics. *Theoretical Computer Science*, 760:15–34, 2018.

9 M. Lafond, E. Noutahi, and N. El-Mabrouk. Efficient Non-Binary Gene Tree Resolution with Weighted Reconciliation Cost. In Roberto Grossi and Moshe Lewenstein, editors, *27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016)*, volume 54 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:12, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

10 M. Lafond, K.M. Swenson, and N. El-Mabrouk. An optimal reconciliation algorithm for gene trees with polytomies. In *LNCS*, volume 7534 of *WABI*, pages 106–122, 2012.

11 J. Sabir, R. Jansen, D. Arasappan, et al. The nuclear genome of Rhazya stricta and the evolution of alkaloid diversity in a medically relevant clade of Apocynaceae. *Scientific Reports*, 6(1):33782, 2007.

12 Y. Zheng and L. Zhang. Reconciliation with nonbinary gene trees revisited. *J. ACM*, 64(4), aug 2017.

Appendix

Proof of Lemma 9

Proof. Let $R_{DL} = \langle G, \tilde{s}, e \rangle$ be the optimal DL-Reconciliation of G with S. We recall that, by definition, $C(R_{DL}) = DL(G, S)$. We will show that we can obtain a DLE-Reconciliation

R_{DLE} of cost lower than or equal to *Cost* from R_{DL} by converting some duplication events into EGT events and by adding EGT-Loss events. Notice that because the costs are unitary, converting a duplication event into an EGT event does not change the cost of the reconciliation. Thus, the cost of R_{DLE} is $C(R_{DL})$ plus the number of EGT-loss events in R_{DLE}.

Let TA be a truth assignment satisfying $\mathcal{C}$ such that the values in each clause are not all equal to each other (we know that such truth assignment exists because $\mathcal{I}$ is a satisfiable instance).

We now construct the genome labeling $\tilde{b}$ (and b) and the mappings e_V and e_E as follows: Let $e_V = e$. Let $e_E(x, y) = A$ for all edge (x, y) of G.

For all j, $1 \leq j \leq m$, such that ℓ_j is True (resp. False) in TA, we set $\tilde{b}(x) = 1$ (resp. $\tilde{b}(x) = 0$) for each nodes x of the left subtree of U_j.

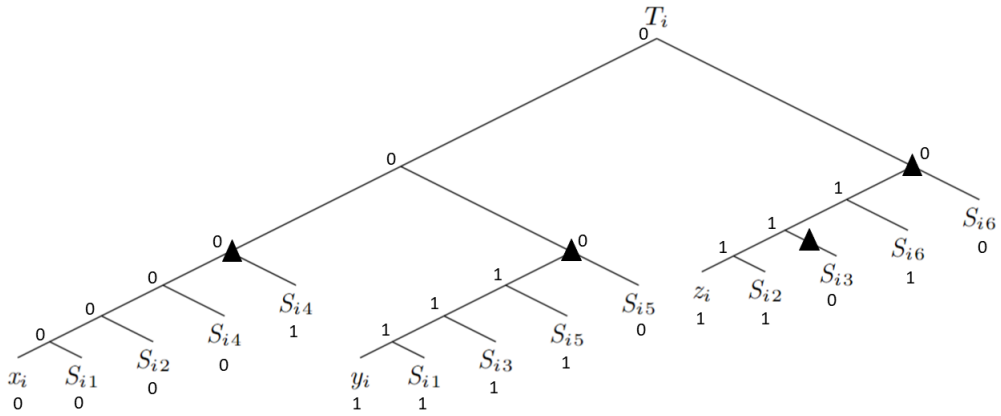
Notice that for each $\sigma \in \Sigma \setminus \{S_{ij} | 1 \leq i \leq k, 1 \leq j \leq 6\}$, we have set the genome label of exactly one of the two leaves of G for which the species label is σ. For each $\sigma \in \Sigma \setminus \{S_{ij} | 1 \leq i \leq k, 1 \leq j \leq 6\}$, we then set the genome label of the leaf with species label σ whose genome label have not been set yet to $1 - i$ where i is the genome label of the other leaf with species label σ.

For each nodes x on the path from the parent of $r(T_1)$ to $r(G)$, we set $\tilde{b}(x) = 0$. We set $\tilde{b}(r(T_i)) = 0$ for $1 \leq i \leq k$ and we set $\tilde{b}(r(U_j)) = 0$ for $1 \leq j \leq m$.

Therefore, there is no EGT-loss event on edges that are not in the subtrees U_j $(1 \leq j \leq m)$ or T_i $(1 \leq i \leq k)$, as all the nodes connected by those edges are labeled by 0.

We now show that no EGT-loss event is required in the subtree U_j of G, for $1 \leq j \leq m$. by construction, all the nodes in the left subtree of U_j have the same genome label i $(i \in \{0, 1\})$ and the node in the right subtree of U_j has the genome label $1-i$. Thus, $\tilde{b}(r(U_j)_l) \neq \tilde{b}(r(U_j)_r)$. Notice that $r(U_j)$ is a duplication node in R_{DL} and recall that $\tilde{b}(r(U_j)) = 0$. We then set $e_V(r(U_j)) = EGT$ which is a transfer from 0 to 1. Therefore, there is no EGT-loss event in the subtree U_j.

We now show that exactly one EGT-loss event is required in the subtree T_i of G, for $1 \leq i \leq k$. Notice that for a clause $C_i = (x \vee y \vee z) \in \mathcal{C}$, x, y and z can't be all equal to each other in TA (because TA is a solution of the instance) and so, by construction, the genome labels of x_i, y_i and z_i in T_i are not all equal to each other. Without loss of generality, let assume that $\tilde{b}(x_i) = 0$, $\tilde{b}(y_i) = 1$ and $\tilde{b}(z_i) = 1$ (the other possible cases are very similar). Then, the following genome labeling $\tilde{b}$ of T_i is correct and requires exactly one EGT-loss event:



We set $e_E(x, y) = P$ where (x, y) is the edge with a triangle on it in the tree above. We also set $e'(lca_{T_i}(\{x_i, S_{i4}\})) = EGT$, $e'(lca_{T_i}(\{y_i, S_{i5}\})) = EGT$ and $e'(lca_{T_i}(\{z_i, S_{i6}\})) = EGT$

(those are the nodes represented by a triangle in the tree above). We can do so because those nodes are duplication nodes in R_{DL}.

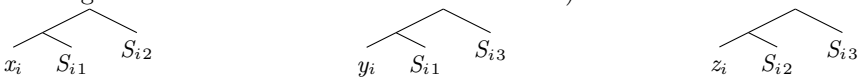
There is then exactly k EGT-loss events in R_{DLE}. Thus, the cost of R_{DLE} is $DL(G,S)+k$ and $C(R_{DLE}) \leq Cost$.

For each leaf x of G, we set $b(x) = \tilde{b}(x)$. Notice that the genome labeling b we construct is consistent with (M,I) as for each $\sigma \in \Sigma$, there is one leaf labeled σ whose genome label is 1 and one leaf labeled σ whose genome label is 0, as needed.

We then obtain a DLE-Reconciliation $R_{DLE} = \langle G, \tilde{s}, \tilde{b}, e_V, e_E \rangle$ of $\langle G, s, b \rangle$ where b is a genome labeling consistent with (M,I) for which $C(R_{DLE}) \leq Cost$ and we conclude that the instance $\mathcal{I}'$ of DVDLE-BinLR admits a DLE-Reconciliation of cost lower than or equal to $Cost$. ◀

Proof of Lemma 10

Proof. For the optimal DLE-Reconciliation $\mathcal{R} = \mathcal{R}'(\langle G, s \rangle, (M,I), S)$, for each clause $C_i = (x \vee y \vee z) \in \mathcal{C}$, $1 \leq i \leq k$, for any genome labeling labeling b consistent with (M,I), there will be at least one EGT-loss event in the three following subtrees of T_i (regardless of the labeling $\tilde{b}$ of the internal nodes of these subtrees) :



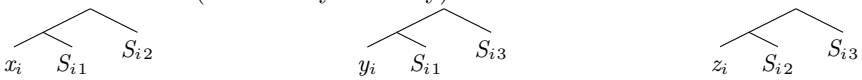
This is the case because there are no duplication node in the DL reconciliation of these subtrees with S (so no EGT events can occur in these subtrees by construction of $\mathcal{R}$) and we know that at least one of these subtrees will not have all its leaves labeled by the same genome label (because two leaves with the same species label can't have the same genome label by construction of the instance) so at least one EGT-loss will be required. ◀

Proof of Lemma 11

Proof. By contradiction, let us suppose that for an unsatisfiable instance $\mathcal{I}$ of the Monotone NAE3SAT problem, its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR does admit a compressed DLE-Reconciliation of cost equal or lower than $Cost$. Let $\mathcal{R} = \mathcal{R}'(\langle G, s \rangle, (M,I), S)$. By lemma 3, $\mathcal{R}$ is optimal and thus $C(\mathcal{R}) \leq DL(G,S) + k$ as $\mathcal{I}'$ does admit a compressed DLE-Reconciliation of cost equal or lower than $Cost = DL(G,S) + k$. By lemma 10, $\mathcal{R}$ is such that there is at least 1 EGT-loss event in each subtree T_i of G for $1 \leq i \leq k$. There is then at least k EGT-loss events in the reconciliation $\mathcal{R}$. As the cost of $\mathcal{R}$ is equal to $DL(G,S)$ plus the number of EGT-loss events in $\mathcal{R}$ (from lemma 4 in [1]), $C(\mathcal{R})$ must be higher than or equal to $DL(G,S)+k$ and we conclude that $C(\mathcal{R}) = DL(G,S)+k$. Thus, there is exactly one EGT-loss event in each subtree T_i of G for $1 \leq i \leq k$ and no EGT-loss event elsewhere in the tree as otherwise $C(\mathcal{R})$ would be higher than $DL(G,S) + k$. In particular, there is no EGT-loss event in the subtrees U_j, $1 \leq j \leq m$, and we can conclude that all nodes in the subtree L'_j, $1 \leq j \leq m$, have the same genome label (there is no EGT event in the subtree L'_j as there is no duplication in the DL-Reconciliation of L'_j with S).

We now define a truth assignment TA as follows: for all $1 \leq j \leq m$, let the variable ℓ_j be true if the genome label of the nodes in L'_j is 1, and let the variable ℓ_j be false otherwise. We now show that TA satisfies $\mathcal{I}$. For each clause $C_i = (x \vee y \vee z) \in \mathcal{C}$, $1 \leq i \leq k$, we need to show that x, y and z are not all equal to each other. Let us suppose by contradiction that this is false, and that there exists a clause $C_i = (x \vee y \vee z) \in \mathcal{C}$ such that x, y and z are all equal to each other. Then, by construction, the genome labels of the leaves x_i, y_i and z_i in the corresponding subtrees T_i are all equal to each other. Then, there is at least 2

EGT-losses events in T_i, as at least two of the following three subtrees of T_i will not have all their leaves labeled by the same genome label and there are no EGT events in those subtrees (by construction) because there are no duplication node in the DL reconciliation of these subtrees with S (this is easy to verify) :



This is a contradiction, as there must be exactly one EGT-loss event in the subtree T_i. We then conclude that for each clause $C_i = (x \vee y \vee z) \in \mathcal{C}$, $1 \leq i \leq k$, x, y and z are not all equal to each other. Thus, the truth assignment TA satisfies $\mathcal{I}$, and we conclude by contradiction that if $\mathcal{I}$ is an unsatisfiable instance of the Monotone NAE3SAT problem, then its corresponding instance $\mathcal{I}'$ of DVDLE-BinLR does not admit a compressed DLE-Reconciliation of cost equal or lower than $Cost$. ◀

Proof of Theorem 14

Proof. The proof is by induction on the number of node $x \in V(G)$ such that $x \in I$.

Notice that the DLE-Reconciliation $\langle G, s, b\rangle$ returned by Algorithm 1 is such that b is a genome labeling consistent with (M, I) by construction.

If there is only one node $x \in V(G)$ such that $x \in I$, then this node x is the root of G by definition. The algorithm then returns an optimal solution, as assume that we can solve $DLEBinLR(\langle G, s\rangle, M'(r(G)), S, i)$ (where $M'(r(G)) = M(r(G))$) for $i \in \{0, 1\}$.

If there is more than one node $x \in V(G)$ such that $x \in I$, then the root of G is in I by definition. By induction, we may assume that for each node $x \in V(G) \setminus r(G)$ such that $x \in I$, the resolution of $G[x]$ computed by the algorithm is exact. For each of those subtrees $G[x]$, we then know the possible genome label(s) at the root leading to an optimal resolution of $G[x]$ and the corresponding optimal resolution of $G[x]$. We now give the index 1 to $|I| - 1$ to the elements of $I \setminus r(G)$. For all $1 \leq j \leq |I| - 1$, there is then two cases for $x_j \in I \setminus r(G)$:

Case 1: $G[x_j]$ is such that both $\tilde{b}(x_j) = 0$ and $\tilde{b}(x_j) = 1$ can lead to an optimal resolution of $G[x_j]$.

In that case, Algorithm 1 will remove $G[x_j]$ from G and replace it by a new leaf without genome label. It solves $G(x_j)$ separately and then replace the new leaf in G by the solved $G[x_j]$ (after the rest of G is solved). $G[x_j]$ can be solved separately in that case, because regardless of the genome label of the parent of $G[x_j]$ in an optimal resolution of (the rest of) G we can obtain an optimal resolution of $G[x_j]$ with $r(G[x_j])$ having the same genome label as its parent (and thus we can obtain an optimal solution to the problem by putting the solved $G[x_j]$ with $r(G[x_j])$ having the same genome label as its parent back in G).

Case 2: $G[x_j]$ is such that only $\tilde{b}(x_j) = i_j$ (where $i_j \in \{0, 1\}$) can lead to an optimal resolution of $G[x_j]$. In that case, Algorithm 1 will remove $G[x_j]$ from G and replace it by a new leaf labeled by i.

Then, Algorithm 1 solves $DLEBinLR(\langle G', s\rangle, M'(r(G)), S)$ where G' is the tree obtained by removing $G[x_j]$ for x_j in Case 1, and by replacing $G[x_j]$ for x_j in Case 2 by a new leaf with the appropriate genome labeling and where $M'(r(G))$ is the appropriate matrix of constraints for the genome labeling of the leaves of G'. By construction, it will then return the solution of lowest cost such that $\tilde{b}(x_j) = i_j$, for all x_j in Case 2.

Let's show that this solution is optimal. By contradiction, suppose that there is $x_j \in I$ (x_j in Case 2) such that there is no optimal solution of the problem for which $\tilde{b}(x_j) = i_j$. Then, the optimal solution $\mathcal{R}^*$ of the problem is such that $\tilde{b}(x_j) \neq i_j$. In $\mathcal{R}^*$, if we set

$\tilde{b}(x_j) = i_j$ and replace the resolved subtree $G[x_j]$ by the optimal resolution of $G[x_j]$ (that we can obtain because $\tilde{b}(x_j) = i_j$), we obtain a new solution $\mathcal{R}'$ of the problem with at most one more EGT-loss event (on the edge $(parent(x), x)$) and such that the resolution of $G[x_j]$ in $\mathcal{R}'$ has a strictly lower cost than the resolution of $G[x_j]$ in $\mathcal{R}^*$. There is then at least one less event in the resolution of $G[x_j]$ in $\mathcal{R}'$ and as the cost are unitary, the solution $\mathcal{R}'$ we obtain is such that $C(\mathcal{R}') \leq C(\mathcal{R}^*)$ and thus $\mathcal{R}'$ is optimal. Contradiction. We then conclude that there is an optimal solution of the problem for which $\tilde{b}(x) = i$.

Thus, Algorithm 1 returns an optimal solution for the input $(\langle G, s\rangle, \tilde{s}, (M, I), S)$.

We conclude, by induction, that the solution returned by Algorithm 1 is optimal.

◀

Proof of Theorem 15

Proof. Once the optimal arrays are computed for all nodes, the optimal solution is easily reconstructed from the entry $min(r(G).array(0, k), r(G).array(1, k))$ by following the pointers from the root to the leaves.

The key point is therefore showing that the arrays computed by Algorithm 3 are exact, i.e., for each node x, $x.array(i, j)$ is equal to $minCostTransfer(\langle G[x], \tilde{s}, e\rangle, j, i)$ where $\langle G[x], \tilde{s}, e\rangle$ is the optimal DL-Reconciliation of $G[x]$ with S.

The proof is by induction on the height of $G(x)$.

If x is a leaf (either free or fixed), it is easy to see that $x.array$ is correct.

Now if x is an internal node, we assume by induction that $x_l.array$ and $x_r.array$ are correct. By contradiction, let's assume that there is (i, j) such that $x.array(i, j) \neq minCostTransfer(\langle G[x], \tilde{s}, e\rangle, j, i)$. Let $\mathcal{R}$ be the optimal DLE-Reconciliation obtained from the optimal DL-Reconciliation by converting some duplication events into EGT events events and by adding some EGT-loss events (this DLE-Reconciliation exists by Lemma 3) leading to $minCostTransfer(\langle G[x], \tilde{s}, e\rangle, j, i)$. Then, in $\mathcal{R}$, $\tilde{b}(x) = i$, $\tilde{b}(x_l) = \ell_1$ where $\ell_1 \in \{0, 1\}$ and $\tilde{b}(x_r) = \ell_2$ where $\ell_2 \in \{0, 1\}$. Also, as there are j free leaves labeled by 0 under x, the sum of the numbers of free leaves labeled by 0 under x_l and x_r must be equal to j. If the genome labels of the children of x are not the same as i, x is converted as an EGT event if x is a duplication node in the DL-Reconciliation (and possibly an EGT-Loss event is added) and if x is not a duplication node then some EGT-Loss events may be added on the edges between x and its children. As the algorithm considers all possibilities of genome labels for x_l and x_r and all possibilities of number of free leaves labeled by 0 under x_r and x_l leading to j free leaves under x (and considers the optimal assignation of EGT and EGT-loss events for the transfer(s) needed from x to its children), the particular possibility leading to $\mathcal{R}$ will be considered and then $x.array(i, j) = minCostTransfer(\langle G[x], \tilde{s}, e\rangle, j, i)$. Contradiction. Thus, there is no such (i, j) and $x.array$ is exact.

We conclude, by induction, that the arrays computed by Algorithm 3 are exact. ◀