On the Complexity of Duplication-Transfer-Loss Reconciliation with Non-Binary Gene Trees

Misagh Kordi and Mukul S. Bansal

Abstract—Duplication-Transfer-Loss (DTL) reconciliation has emerged as a powerful technique for studying gene family evolution in the presence of horizontal gene transfer. DTL reconciliation takes as input a gene family phylogeny and the corresponding species phylogeny, and reconciles the two by postulating speciation, gene duplication, horizontal gene transfer, and gene loss events. Efficient algorithms exist for finding optimal DTL reconciliations when the gene tree is binary. However, gene trees are frequently non-binary. With such non-binary gene trees, the reconciliation problem seeks to find a binary resolution of the gene tree that minimizes the reconciliation cost. Given the prevalence of non-binary gene trees, many efficient algorithms have been developed for this problem in the context of the simpler Duplication-Loss (DL) reconciliation model. Yet, no efficient algorithms exist for DTL reconciliation with non-binary gene trees and the complexity of the problem remains unknown. In this work, we resolve this open question by showing that the problem is, in fact, NP-hard. Our reduction applies to both the dated and undated formulations of DTL reconciliation. By resolving this long-standing open problem, this work will spur the development of both exact and heuristic algorithms for this important problem.

1 Introduction

Duplication-Transfer-Loss (DTL) reconciliation is one of the most powerful techniques for studying gene and genome evolution in microbes and other non-microbial species engaged in horizontal gene transfer. DTL reconciliation accounts for the role of gene duplication, gene loss, and horizontal gene transfer in shaping gene families and can infer these evolutionary events through the systematic comparison and reconciliation of gene trees and species trees. *Gene trees* represent the evolutionary histories of gene families, while *species trees* represent the evolutionary histories of the corresponding species. Given a gene tree and a species tree, DTL reconciliation shows the evolution of the gene tree inside the species tree, and explicitly infers duplication, transfer, and loss events. Accurate knowledge of gene family evolution has many uses in biology, including inference of orthologs, paralogs and xenologs for functional genomic studies, e.g., [1], [2] , reconstruction of ancestral gene content, e.g., [3], [4], and accurate gene tree and species tree construction, e.g., [2], [5], [6], [7], [8], as well as potential application to error-correcting taxonomic assignments of metagenomic reads. Consequently, the DTL reconciliation problem has been widely studied, e.g., [4], [9], [10], [11], [12], [13], [14], [15], [16].

DTL reconciliation is typically formulated using a parsimony framework where each evolutionary event is assigned a cost and the goal is to find a reconciliation with minimum total cost. The resulting optimization problem is called the *DTL-reconciliation problem*. DTL-reconciliations can sometimes be *time-inconsistent*; i.e, the inferred transfers may induce contradictory constraints on the dates for the internal nodes of the species tree. The problem of finding an optimal *time-consistent* reconciliation is known to be NP-hard [11], [17]. Thus, in practice, the goal is to find an optimal (not necessarily time-consistent) DTL-reconciliation [4], [11], [12], [14], [16] and this problem can be solved in $O(mn)$ time [12], where m and n denote the number of nodes in the gene tree and species tree, respectively. Interestingly, the problem of finding an optimal time-consistent reconciliation actually becomes efficiently solvable [10], [18] in $O(mn^2)$ time if the species tree is fully dated. Thus, these two efficiently solvable formulations, regular and dated, are the two standard formulations of the DTL-reconciliation problem.

Both these formulations of the DTL-reconciliation problem assume that the input gene tree and species tree are binary. However, while relatively accurate species trees can

- *Misgah Kordi is with the Department of Computer Science and Engineering at the University of Connecticut, Storrs, USA.* misagh.kordi@uconn.edu
- *Mukul S. Bansal is with the Department of Computer Science & Engineering and the Institute for Systems Genomics at the University of Connecticut, Storrs, USA.* mukul@engr.uconn.edu

be obtained through the use of well-behaved orthologous gene families or multi-gene species tree reconstruction methods [6], [19], [20], gene tree inference is confounded by the fact that there is often insufficient information in the underlying gene sequences to fully resolve gene tree topologies. As a result, gene trees are frequently non-binary in practice. When the input consists of a non-binary gene tree, the reconciliation problem seeks to find a binary resolution of the gene tree that minimizes the reconciliation cost. Given the prevalence of non-binary gene trees, many efficient algorithms have been developed for this problem in the context of the simpler Duplication-Loss (DL) reconciliation model [5], [21], [22], [23], with the most efficient of these algorithms having an optimal $O(m + n)$ time complexity [23]. However, the DTL reconciliation model is more general and significantly more complex than the DL reconciliation model. Consequently, no efficient algorithms exist for DTL reconciliation with non-binary gene trees and the complexity of the problem remains unknown. As a result, DTL reconciliation is currently inapplicable to non-binary gene trees, significantly reducing its utility in practice.

In this work, we settle this open problem by proving that the DTL-reconciliation problem on non-binary gene trees is, in fact, NP-hard. Our proof is based on a reduction from the minimum 3-set cover problem and applies to both formulations of the DTL-reconciliation problem. An especially desirable feature of our reduction is that it implies NP-hardness for biologically relevant settings of the event cost parameters, showing that the problem is difficult even for biologically meaningful scenarios. By settling this question, our work will spur the development of both exact (better than brute-force) and efficient approximation and heuristic algorithms for this important problem.

A preliminary version of this work, without any proofs and with only some of the lemmas, appeared in the proceedings of ISBRA 2015 [24]. The current manuscript substantially expands upon [24] and contains an improved and more detailed exposition, many additional lemmas, and all proofs.

We develop our NP-hardness proof in the context of the regular (undated) DTL-reconciliation formulation, and revisit dated DTL-reconciliation later in Section 4. The next section introduces basic definitions and preliminaries, and we present the NP-hardness proof for the optimal gene tree resolution problem in Section 3. Concluding remarks appear in Section 5.

2 Definitions and Preliminaries

We follow the basic definitions and notation from [12]. Given a tree T, we denote its node, edge, and leaf sets by $V(T)$, $E(T)$, and $Le(T)$ respectively. If T is rooted, the root node of T is denoted by $rt(T)$, the parent of a node $v \in V(T)$ by $pa_T(v)$, its set of children by $Ch_T(v)$, and the (maximal) subtree of T rooted at v by $T(v)$. The set of *internal nodes* of T, denoted $I(T)$, is defined to be $V(T) \setminus Le(T)$. We define $\leq_T$ to be the partial order on $V(T)$ where $x \leq_T y$ if y is a node on the path between $rt(T)$ and x. The partial order $\geq_T$ is defined analogously, i.e., $x \geq_T y$ if x is a node on the path between $rt(T)$ and y. We say that y is an *ancestor* of x, or that x is a *descendant* of y, if $x \leq_T y$ (note that, under this definition, every node is a descendant as well as ancestor of itself). We say that x and y are *incomparable* if neither $x \leq_T y$ nor $y \leq_T x$. Given a non-empty subset $L \subseteq Le(T)$, we denote by $lca_T(L)$ the last common ancestor (LCA) of all the leaves in L in tree T; that is, $lca_T(L)$ is the unique smallest upper bound of L under $\leq_T$. Given $x, y \in V(T)$, $x \rightarrow_T y$ denotes the unique path from x to y in T. We denote by $d_T(x, y)$ the number of edges on the path $x \rightarrow_T y$; note that if $x = y$ then $d_T(x, y) = 0$. Throughout this work, the *term* tree refers to rooted trees. A tree is *binary* if all of its internal nodes have exactly two children, and *non-binary* otherwise. We say that a tree T' is a *binary resolution* of T if T' is binary and T can be obtained from T' by contracting one or more edges. We denote by $\mathcal{BR}(T)$ the set of all binary resolutions of a non-binary tree T.

Gene trees may be either binary or non-binary while the species tree is always assumed to be binary. Throughout this work, we denote the gene tree and species tree under consideration by G and S, respectively. If G is restricted to be binary we refer to it as G^B and as G^N if it is restricted to be non-binary. We assume that each leaf of the gene tree is labeled with the species from which that gene was sampled. This labeling defines a *leaf-mapping* $\mathcal{L}_{G,S}\colon Le(G) \rightarrow Le(S)$ that maps a leaf node $g \in Le(G)$ to that unique leaf node $s \in Le(S)$ which has the same label as g. Note that gene trees may have more than one gene sampled from the same species. We will implicitly assume that the species tree contains all the species represented in the gene tree.

2.1 Reconciliation and DTL-scenarios

A binary gene tree can be reconciled with a species tree by mapping the gene tree into the species tree. Next, we define what constitutes a valid reconciliation; specifically, we define a Duplication-Transfer-Loss scenario (DTL-scenario) [11], [12] for G^B and S that characterizes the mappings of G^B into S that constitute a biologically valid reconciliation. Essentially, DTL-scenarios map each gene tree node to a unique species tree node in a consistent way that respects the immediate temporal constraints implied by the species tree, and designate each gene tree node as representing either a speciation, duplication, or transfer event. For any gene tree node, say g, that represents a transfer event, DTL-scenarios also specify which of the two edges (g, g') or (g, g''), where g', g'' denote the children

of g, represents the transfer edge on S, and identify the recipient species of the corresponding transfer.

Definition 1 (DTL-scenario)**.** *A DTL-scenario for G^B and S is a seven-tuple $\langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$, where $\mathcal{L}\colon Le(G^B) \rightarrow Le(S)$ represents the leaf-mapping from G^B to S, $\mathcal{M}\colon V(G^B) \rightarrow V(S)$ maps each node of G^B to a node of S, the sets Σ, Δ, and Θ partition $I(G^B)$ into speciation, duplication, and transfer nodes respectively, Ξ is a subset of gene tree edges that represent transfer edges, and $\tau\colon \Theta \rightarrow V(S)$ specifies the recipient species for each transfer event, subject to the following constraints:*

1) *If $g \in Le(G^B)$, then $\mathcal{M}(g) = \mathcal{L}(g)$.*
2) *If $g \in I(G^B)$ and g' and g'' denote the children of g, then,*
   a) *$\mathcal{M}(g) \not\leq_S \mathcal{M}(g')$ and $\mathcal{M}(g) \not\leq_S \mathcal{M}(g'')$,*
   b) *At least one of $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ is a descendant of $\mathcal{M}(g)$.*
3) *Given any edge $(g, g') \in E(G^B)$, $(g, g') \in \Xi$ if and only if $\mathcal{M}(g)$ and $\mathcal{M}(g')$ are incomparable.*
4) *If $g \in I(G^B)$ and g' and g'' denote the children of g, then,*
   a) *$g \in \Sigma$ only if $\mathcal{M}(g) = lca(\mathcal{M}(g'), \mathcal{M}(g''))$ and $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ are incomparable,*
   b) *$g \in \Delta$ only if $\mathcal{M}(g) \geq_S lca(\mathcal{M}(g'), \mathcal{M}(g''))$,*
   c) *$g \in \Theta$ if and only if either $(g, g') \in \Xi$ or $(g, g'') \in \Xi$.*
   d) *If $g \in \Theta$ and $(g, g') \in \Xi$, then $\mathcal{M}(g)$ and $\tau(g)$ must be incomparable, and $\mathcal{M}(g')$ must be a descendant of $\tau(g)$, i.e., $\mathcal{M}(g') \leq_S \tau(g)$.*

Constraint 1 above ensures that the mapping $\mathcal{M}$ is consistent with the leaf-mapping $\mathcal{L}$. Constraint 2a imposes on $\mathcal{M}$ the temporal constraints implied by S. Constraint 2b implies that any internal node in G^B may represent at most one transfer event. Constraint 3 determines the edges of T that are transfer edges. Constraints 4a, 4b, and 4c state the conditions under which an internal node of G^B may represent a speciation, duplication, and transfer respectively. Constraint 4d specifies which species may be designated as the recipient species for any given transfer event.

DTL-scenarios correspond naturally to reconciliations and it is straightforward to infer the reconciliation of G^B and S implied by any DTL-scenario. Figure 1 shows an example of a DTL-scenario. Given a DTL-scenario α, one can directly count the minimum number of gene losses, $Loss_\alpha$, in the corresponding reconciliation. For brevity, we refer the reader to [12] for further details on how to count losses in DTL-scenarios.

Let P_Δ, P_Θ, and P_{loss} denote the non-negative costs associated with duplication, transfer, and loss events, respectively. The reconciliation cost of a DTL-scenario is defined as follows.

Definition 2 (Reconciliation cost of a DTL-scenario)**.** *Given a DTL-scenario $\alpha = \langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$ for G^B and S, the* reconciliation cost *associated with α is given by $\mathcal{R}_\alpha = P_\Delta \cdot |\Delta| + P_\Theta \cdot |\Theta| + P_{loss} \cdot Loss_\alpha$.*

A most parsimonious reconciliation is one that has minimum reconciliation cost.

Definition 3 (Most Parsimonious Reconciliation (MPR))**.** *Given G^B and S, along with P_Δ, P_Θ, and P_{loss}, a* most parsimonious reconciliation (MPR) *for G^B and S is a DTL-scenario with minimum reconciliation cost.*

2.2 Optimal gene tree resolution

Non-binary gene trees cannot be directly reconciled against a species tree. Thus, given a non-binary gene tree G^N, the problem is to find a binary resolution of G^N whose MPR with S has the smallest reconciliation cost.

Problem 1 (Optimal Gene Tree Resolution (OGTR))**.** *Given G^N and S, along with P_Δ, P_Θ, and P_{loss}, the* Optimal Gene Tree Resolution (OGTR) *problem is to find a binary resolution G^B of G^N such that the MPR of G^B and S has the smallest reconciliation cost among all $G^B \in \mathcal{BR}(G^N)$.*

An example of a non-binary gene tree and a binary resolution is shown in Figure 1.

3 NP-HARDNESS OF THE OGTR PROBLEM

We claim that the OGTR problem is NP-hard; specifically, that the corresponding decision problem is NP-Complete. The decision version of the OGTR problem is as follows:

Problem 2 (D-OGTR)**.**

Instance: *G^N and S, event costs P_Δ, P_Θ, and P_{loss}, and a non-negative integer l.*

Question: *Does there exist a $G^B \in \mathcal{BR}(G^N)$ such that the MPR of G^B and S has reconciliation cost at most l?*

Theorem 1. *The D-OGTR problem is NP-Complete.*

The D-OGTR problem is clearly in NP. In the remainder of this section we will show that the D-OGTR problem is NP-hard using a poly-time reduction from the decision version of the NP-hard *minimum 3-set cover* problem [25].

3.1 Reduction from minimum 3-set cover

The decision version of minimum 3-set cover can be stated as follows.

Problem 3 (M3SC)**.**

Instance: *Given a set of n elements $U = \{u_1, u_2, \ldots, u_n\}$, a set $A = \{A_1, A_2, \ldots, A_m\}$ of m subsets of U such that $|A_i| = 3$ for each $1 \leq i \leq m$, and a nonnegative integer $k \leq m$.*

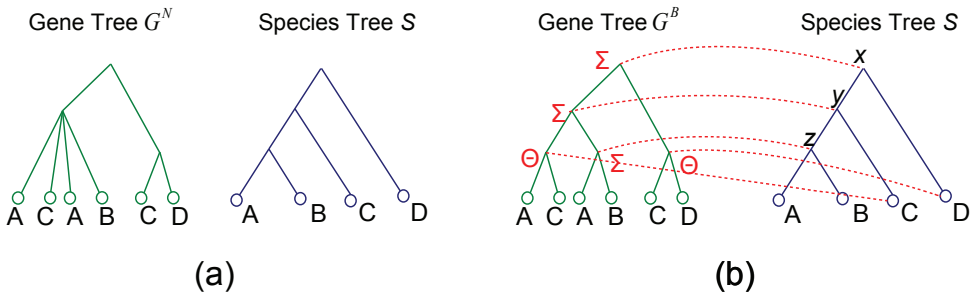

Fig. 1. **DTL reconciliation and OGTR problem.** Part (a) shows a non-binary gene tree G^N and binary species tree S. Part (b) shows a DTL reconciliation between a possible binary resolution G^B of G^N and species tree S. The dotted arcs show the mapping $\mathcal{M}$ (with the leaf mapping being specified by the leaf labels on the gene tree), and the label at each internal node of G^B specifies the type of event represented by that node. This reconciliation invokes two transfer events.

Question: *Is there a subset of A of size at most k whose union is U?*

We point out that the M3SC problem as defined above is a slight variation of the traditional minimum 3-set cover problem: In our formulation the subsets of U in A are restricted to have *exactly* three elements each while the traditional formulation allows for the subsets to have *less than or equal to* three elements [25]. However, it is easy to establish that the NP-Completeness of the traditional version immediately implies the NP-completeness of our formulation of the M3SC problem.

We will also assume, without any loss of generality, that each element u_i appears in at least two subsets from A. Elements that only appear in one subset imply necessary inclusion of that subset and so M3SC instances where an element occurs in a single subset can be trivially reduced to instances where each element appears in at least two subsets from A.

Consider an instance ϕ of the M3SC problem with $U = \{u_1, u_2, \ldots, u_n\}$, $A = \{A_1, A_2, ..., A_m\}$, and k given. We now show how to transform ϕ into an instance λ of the D-OGTR problem by constructing G^N and S and setting the three event costs in such a way that there exists a YES answer to the M3SC instance ϕ if and only if there exists a YES answer to the D-OGTR instance λ with $l = k + 48m - 12n$.

3.2 Gadget

Gene tree. We first show how to construct the gene tree G^N. Note that each element of U occurs in at least two of the subsets from A. We will treat each of the occurrences of an element separately and will order them according to the indices p of the A_p's which contain that element. More precisely, for an element $u_i \in U$, we denote by $x_{i,j}$ the j^{th} occurrence of u_i in A. For instance, if element u_5 occurs in the subsets A_2, A_4, A_{10}, and A_{25}, then $x_{5,2}$ refers to the occurrence of u_5 in A_4, while $x_{5,4}$ refers to the occurrence of u_5 in A_{25}.

Let c_i denote the cardinality of the set $\{A_p : u_i \in A_p, \text{ for } 1 \leq p \leq m\}$. Then, $x_{i,j}$ is well defined as long as $1 \leq i \leq n$ and $1 \leq j \leq c_i$. Each $x_{i,j}$ will correspond to exactly four leaves, $x_{i,j,1}$, $x_{i,j,2}$, $x_{i,j,3}$, and $x_{i,j,4}$ in the gene tree G^N. In addition, the leaf set of G^N also contains a special node that we label *start*, provided for orienting the reconciliation.

Thus, $\mathit{Le}(G^N) = \{x_{i,j,1}, x_{i,j,2}, x_{i,j,3}, x_{i,j,4} : 1 \leq i \leq n \text{ and } 1 \leq j \leq c_i\} \cup \{\mathit{start}\}$. The overall structure of G^N is shown in Figure 2(a). As shown, the root node of the gene tree is unresolved and has $3m + 3n + 1$ children consisting of (i) the *start* node, (ii) the $\sum_{i=1}^{n} c_i = 3m$ leaf nodes, collectively called *blue* nodes, and (iii) the $3n$ internal nodes labeled g_i, g_i', and g_i'', for each $1 \leq i \leq n$. These internal nodes represent the n elements in U and the subtrees rooted at those nodes have the structure shown in Figure 2(a). Note that the number of children for each of the internal nodes labeled g_i, g_i', and g_i'', for $1 \leq i \leq n$, is c_i. These nodes may thus be either binary or non-binary. The leaves labeled $x_{i,j,3}$ appear in the node g_i', those labeled $x_{i,j,4}$ appear in g_i'', and those labeled $x_{i,j,1}$ or $x_{i,j,2}$ appear in g_i. The $x_{i,j,1}$'s also appear in the collection of blue nodes and thus appear twice in the gene tree. Note, also, that all the children of a node g_i, for $1 \leq i \leq n$, are themselves internal nodes (and binary) and are labeled as $y_{i,j}$, where $1 \leq j \leq c_i$.

Species tree. Next, we show how to construct the species tree S. The tree S is binary and consists of m subtrees whose root nodes are labeled $s_1, \ldots s_m$, each corresponding to a subset from A, connected together through a backbone tree as shown in Figure 2(b). The exact structure of this backbone tree is unimportant, as long as each s_i is sufficiently separated from the roots of the rest of the subtrees. For concreteness, we will assume that this backbone consists of a "caterpillar" tree as shown Figure 2(b), and that $9m$ extraneous leaves (not present in the gene tree) have been added to this backbone as shown in the figure to ensure that each pair of subtrees is sufficiently separated.

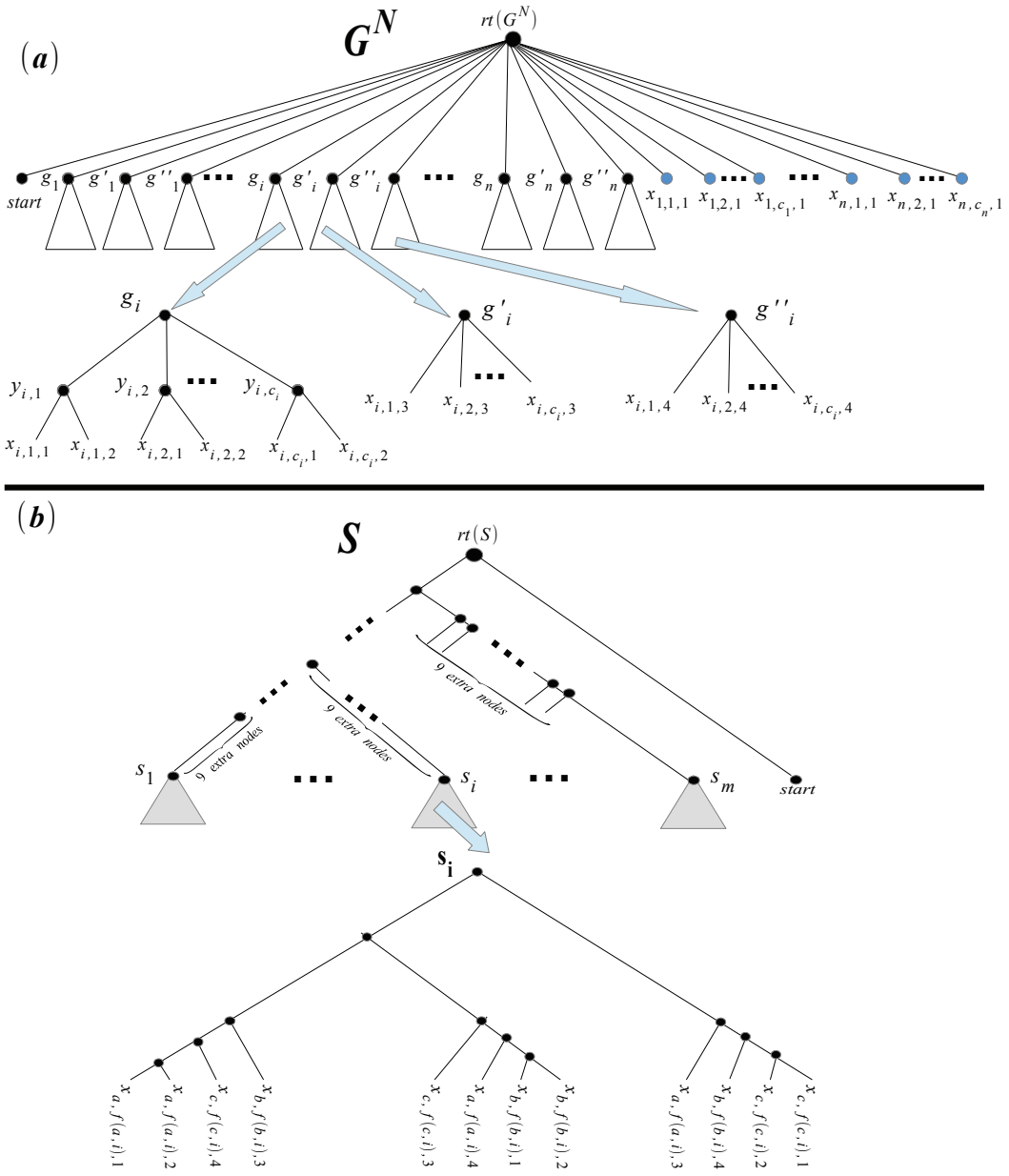

Fig. 2. **Construction of non-binary gene tree and species tree.** (a) Structure of the non-binary gene tree G^N. (b) Structure of the species tree S.

Recall that we use $x_{i,j}$ to denote the j^{th} occurrence of u_i in A. Assuming that $u_i \in A_p$ and that $x_{i,j}$ refers to the occurrence of u_i in A_p, we define $f(i,p)$ to be j. In other words, if the j^{th} occurrence of an element u_i is in the subset A_p, then we assign $f(i,p)$ to be j. Each S_i corresponds to the subset A_i and has the structure depicted in Figure 2(b). In particular, if A_i contains the three elements u_a, u_b, and u_c, then S_i contains the 12 leaves labeled $x_{a,f(a,i),j}$, $x_{b,f(b,i),j}$, and $x_{c,f(c,i),j}$, for $1 \leq j \leq 4$.

Event costs. We assign the following event costs for problem instance λ: $P_\Delta = 2$, $P_\Theta = 4$, and $P_{loss} = 1$.

Note that the D-OGTR instance λ can be constructed in time polynomial in m and n.

Claim 1. *There exists a YES answer to the M3SC instance ϕ if and only if there exists a YES answer to the D-OGTR instance λ with $l = k + 48m - 12n$.*

The remainder of this section is devoted to proving

this claim which, in turn, would complete our proof for Theorem 1. We begin by explaining the main idea of the reduction and describing the association between the instances ϕ and λ, and then prove the forward and reverse directions of the claim.

3.3 Key insight

The main idea behind our reduction can be explained as follows: In the gene tree G^N, subtrees $G^N(g_i)$, $G^N(g'_i)$ and $G^N(g''_i)$ correspond to the element u_i, for each $1 \leq i \leq n$, while in the species tree the subtree $S(s_j)$ corresponds to the subset A_j, for each $1 \leq j \leq m$. Let G^B be any binary resolution of G^N. It can be shown that in any MPR of any optimal binary resolution G^B of G^N the following must hold: For each $i \in \{1, \ldots, n\}$, g_i (along with g'_i and g''_i) must map to an $S(s_j)$ for which $u_i \in A_j$. Under these restrictions on the mappings, observe that if we were to solve the OGTR problem on G^N and S and then choose all those A_j's for which the subtree $S(s_j)$ has at least one of the g_i's mapping into it, then the set of chosen A_j's would cover all the elements of U.

The source of the optimization is that, due to the specific construction of the gene tree and species tree, it is more expensive (in terms of reconciliation cost) to use more $S(s_j)$'s for the mapping. Thus, all the g_i's (along with g'_i's and g''_i's) must map to as few of the subtrees, $S(s_j)$'s, as possible. Recall that the OGTR problem optimizes the topology of the binary resolution G^B in such a way that its MPR with S has minimum reconciliation cost. Thus, the OGTR problem effectively optimizes the topology of G^B in a way that minimizes the total number of $S(s_j)$'s receiving mappings from the g_i's, g'_i's, or g''_i's, yielding a set cover of smallest possible size. This is the key idea behind our reduction and we develop this idea further in the next two subsections.

3.4 Proof of Claim 1: forward direction

Let us assume that we have a YES answer for the M3SC instance ϕ. We will show how to create a binary resolution G^B of G^N whose MPR with S has reconciliation cost at most $k + 48m - 12n$.

We first show how to resolve the subtrees $G^N(g_i)$, $G^N(g'_i)$, and $G^N(g''_i)$, for $1 \leq i \leq n$. Recall that, for any fixed i, these three subtrees correspond to element u_i of U. The $y_{i,j}$'s in $G^N(g_i)$ correspond to the different occurrences of element u_i in the subsets from A. The same holds for the $x_{i,j,3}$'s in $G^N(g'_i)$ and the $x_{i,j,4}$'s in $G^N(g''_i)$.

Suppose a solution to instance ϕ consists of the k subsets $A_{r(1)}, A_{r(2)}, \ldots, A_{r(k)}$. Since every element in U must be covered by at least one of these k subsets, we can designate a *covering subset* for each element $u_i \in U$, $1 \leq i \leq n$, chosen arbitrarily from among those subsets in the solution that contain u. Suppose that element u_i is assigned the covering subset A_j (so we must have $u_i \in A_j$ and $A_j \in \{A_{r(1)}, A_{r(2)}, \ldots, A_{r(k)}\}$). The subtree $G^N(g_i)$ will then be resolved as follows: The $y_{i,j}$ corresponding to the occurrence of u_i in A_j, i.e., $y_{i,f(i,j)}$, will be separated out as one of the two children of g_i. The other child of g_i will be the root of an arbitrary caterpillar tree on all the remaining $y_{i,j}$'s in $G^N(g_i)$. This is depicted in Figure 3(d). The subtrees $G^N(g'_i)$ and $G^N(g''_i)$ are resolved similarly, except that in $G^N(g'_i)$ the leaf node $x_{i,f(i,j),3}$ is separated out and in $G^N(g''_i)$ the leaf node $x_{i,f(i,j),4}$ is separated out. Thus, the resolution of $G^N(g_i)$, $G^N(g'_i)$, and $G^N(g''_i)$ is done based on the assigned covering subset of element u_i. This is repeated for all i, where $1 \leq i \leq n$.

Next, we show how to resolve the root node of G^N to obtain G^B. The *start* node will become an outgroup to the rest of G^B. The backbone of the rest of G^B consists of an arbitrary caterpillar tree on k "leaf" nodes as shown in Figure 3(a). These k nodes are labeled $h_{r(1)}, \ldots h_{r(k)}$ and are the root nodes of k subtrees. Each of the k subtrees corresponds to one of the subsets $A_{r(1)}, A_{r(2)}, \ldots, A_{r(k)}$. In particular, subtree $G^B(h_{r(i)})$, for $1 \leq i \leq k$ corresponds to the subset $A_{r(i)}$. Each of the blue nodes and the subtrees rooted at the g_i's, g'_i's, and g''_i's, for $1 \leq i \leq n$ will be included in one of these k subtrees. Specifically, the subtree $G^B(h_{r(j)})$ will include all those g_i's, g'_i's, and g''_i's for which the covering subset of the corresponding u_i is $A_{r(j)}$. Since there may be 0, 1, 2, or 3 i's for which the covering subset of u_i is $A_{r(j)}$, the sizes of different $G^B(h_{r(j)})$ subtrees may vary. The structure of $G^B(h_{r(j)})$ when there are 3 i's is depicted in Figure 3(b). The structure of $G^B(h_{r(j)})$ when there are only 1 or 2 such i's is similar and is the induced subtree, on the relevant i's, of the full subtree for all 3 i's. As shown in the figure, note that each subtree $G^B(h_{r(j)})$ also includes at least three blue nodes, corresponding to the three elements in $A_{r(j)}$. These three blue nodes are included even for cases where there are fewer than 3 i's. Thus, when there are 0 such i's, which can happen when the size of the minimum set cover for instance ϕ is less than k, the subtree $G^B(h_{r(j)})$ consists of the three blue nodes.

This results in the assignment of all g_i's, g'_i's, and g''_i's, for $1 \leq i \leq n$ to one of the subtrees $G^B(h_{r(j)})$, for $1 \leq j \leq k$. As discussed above, $3k$ out of the $3m$ blue nodes also get assigned in this process. The remaining $3m - 3k$ of the blue nodes are organized into an arbitrary caterpillar tree and added to the subtree $G^B(h_{r(k)})$ as shown in Figure 3(c).

This finishes our description of G^B. The next lemma follows directly from this construction of G^B.

Lemma 1. *Gene tree G^B is a binary resolution of G^N.*

Proof. From the construction of G^B from G^N above, it is easy to verify that all edges (or, more accurately, clusters) in

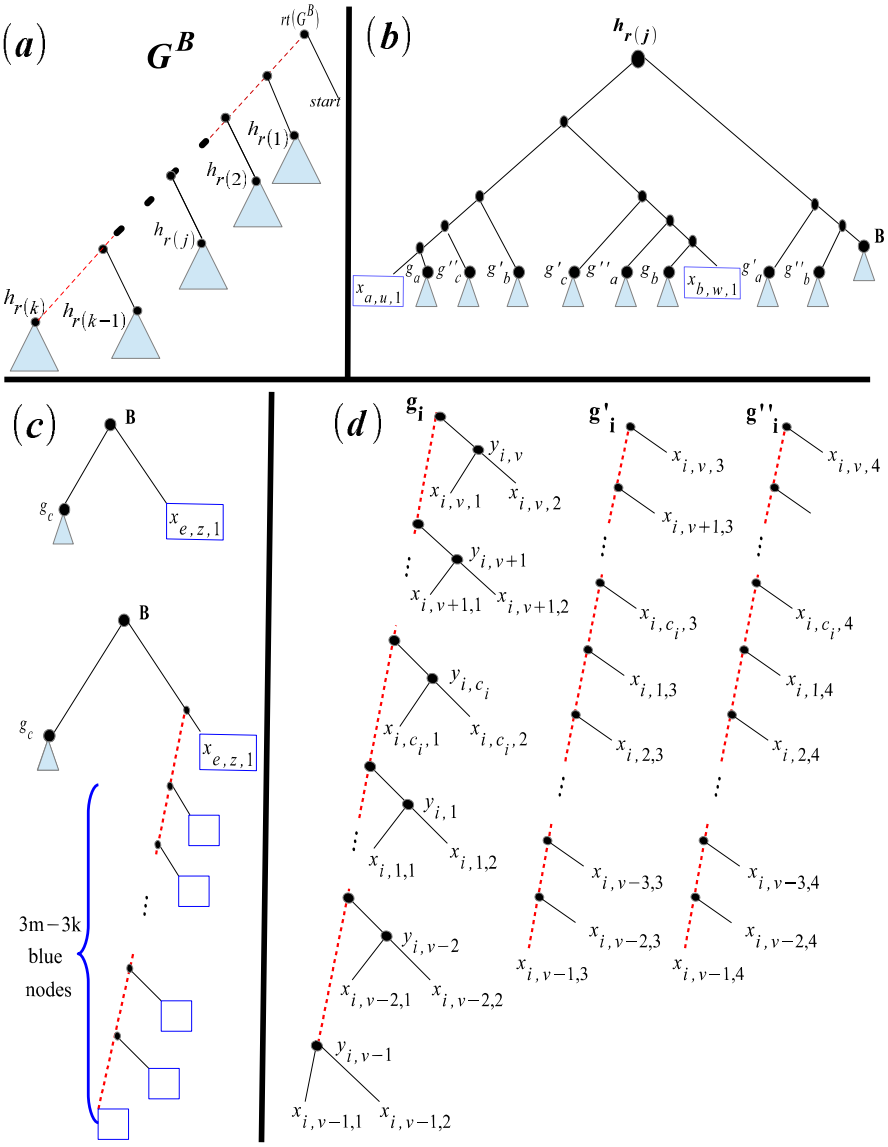

Fig. 3. **Resolution of G^N into G^B.** (a) The structure of the backbone of the gene tree G^B. (b) Structure of the subtree $h_{r(j)}$ for any $j \in \{1, \ldots, k\}$. (c) The two possible structures of the subtree with root B in $h_{r(j)}$. For any $j \in \{1, \ldots, k-1\}$, this subtree is as shown at the top of part (c) while, for $j = k$, it is as shown at the bottom and includes all the "remaining" $3m - 3k$ blue nodes. (d) The resolution of the g_i's, g'_i's, g''_i's. In the figure, u_a, u_b, and u_c represent the three elements in $A_{r(j)}$, with $u = f(a, r(j))$, $w = f(b, r(j))$, and $z = f(c, r(j))$. In part (d), if the covering subset of element u_i is A_p, then v represents $f(i, p)$. The labels inside the blue boxes represent blue nodes.

G^N also appear in G^B. By construction, G^B is also binary. Thus, G^B is a binary resolution of G^N. □

Next, we show how to construct a DTL-scenario for G^B and S with cost at most $k + 48m - 12n$.

DTL-scenario for G^B and S. All leaves of the gene tree, G^B, map to the corresponding leaves on the species tree S. Consider the depiction of G^B as shown in Figure 3. For each i such that $1 \leq i \leq k-1$, $h_{r(i)}$ and $pa(h_{r(i)})$ map to s_i. The node $pa(h_{r(i)})$ represents a transfer event and $h_{r(i)}$ a speciation event. Finally, $h_{r(k)}$ maps to s_k and represents a speciation event.

For each internal node a in subtree B, if only one child of a is a leaf node then a has the same mapping as its unique leaf-child. If both children of a are leaf nodes, then it has the same mapping as any one of them. Thus, all internal nodes of B are transfer nodes.

For each i, consider subtree $G^B(h_{r(i)})$. For each el-

ement j represented in that subtree, g'_j and g''_j are all transfer nodes and map to leaves $x_{j,v,3}$ and $x_{j,v,4}$ on $S(s_i)$, respectively. Consider any internal node a in the subtrees $G^B(g'_j)$ and $G^B(g''_j)$. If only one child of a is a leaf node then a has the same mapping as its unique leaf-child. If both children of a are leaf nodes, then it has the same mapping as any one of them. Thus, all internal nodes of $G^B(g'_j)$ and $G^B(g''_j)$ are transfers. In the subtree $G^B(g_j)$, each node labeled $y_{,\cdot}$ is a speciation node and maps to the LCA of the mapping of its two children. Consider any other internal node a in the subtree $G^B(g_j)$. If only one child of a is a $y_{,\cdot}$ node then a has the same mapping as its unique $y_{,\cdot}$-child. If both children of a are $y_{,\cdot}$ nodes, then it has the same mapping as any one of them. Thus, all nodes along the spine of $G^B(g_j)$ are transfers. Furthermore, $pa(g_j)$ is a duplication node, while $pa(g'_j)$ and $pa(g''_j)$ are both speciation nodes.

The root of G^B, maps to the *start* node on the species tree S and is a transfer node. All other nodes of G^B are speciation nodes. We denote the resulting DTL-scenario for G^B and S by α. It is not difficult to verify that α is a valid DTL-scenario.

The following two lemmas help bound the cost of the reconciliation implied by α.

Lemma 2. *Under DTL-scenario α, the reconciliation cost of any subtree $G^B(g_j)$, $G^B(g'_j)$, or $G^B(g''_j)$, for $1 \leq j \leq n$, with S is $(c_j - 1) \times P_\Theta$.*

Proof. Based on the reconciliation implied by α, each internal node along the spine of any subtree $G^B(g_j)$, $G^B(g'_j)$, or $G^B(g''_j)$, for $1 \leq j \leq n$, is a transfer node. Note that each of the nodes in $G^B(g_j)$ labeled $y_{j,\cdot}$ is a speciation node and the subtrees rooted at the $y_{j,\cdot}$'s do not invoke any losses. Thus, none of the subtrees $G^B(g_j)$, $G^B(g'_j)$, or $G^B(g''_j)$, for $1 \leq j \leq n$, invoke any duplications or losses. Since the number of internal nodes along the spines of each of $G^B(g_j)$, $G^B(g'_j)$, or $G^B(g''_j)$, for $1 \leq j \leq n$, is $c_j - 1$, the lemma follows. □

Recall that, since there may be 0, 1, 2, or 3 i's for which the covering subset of u_i is $A_{r(j)}$, the sizes of different $G^B(h_{r(j)})$ subtrees may vary. The next two lemmas shows that, under α, the reconciliation cost of any subtree $G^B(h_{r(j)})$ behaves predictably. the next lemma applies to all $G^B(h_{r(j)})$ where $1 \leq j \leq k-1$. We separate out the case of $j = k$ as a separate lemma since all the unassigned blue nodes get attached to $G^B(h_{r(k)})$.

Lemma 3. *For each j, $1 \leq j \leq k-1$, the total reconciliation cost of subtree $G^B(h_{r(j)})$ with S under DTL-scenario α is as follows:*

1) *If there exist exactly three distinct subtrees g_a, g_b, and g_c, where $1 \leq a, b, c \leq n$, within subtree $G^B(h_{r(j)})$, then the reconciliation cost is $12 \times (c_a + c_b + c_c - 3) + 9$.*
2) *If there exist exactly two distinct subtrees g_a and g_b, where $1 \leq a, b \leq n$, within subtree $G^B(h_{r(j)})$, then the reconciliation cost is $12 \times (c_a + c_b - 2) + 9$.*
3) *If there exists exactly one subtree g_a, where $1 \leq a \leq n$, within subtree $G^B(h_{r(j)})$, then the reconciliation cost is $12 \times (c_a - 1) + 9$.*
4) *If there do not exist any subtrees of the form g_a, where $1 \leq a \leq n$, within subtree $G^B(h_{r(j)})$, then the reconciliation cost is 9.*

Proof. Consider the first case of the lemma. Based on Lemma 3.4, the reconciliation cost of any subtree $G^B(g_i)$, $G^B(g'_i)$, $G^B(g''_i)$, for each $1 \leq i \leq n$, with S is $P_\Theta \times (c_i - 1)$. Thus, the total reconciliation cost contributed by all such subtrees is $P_\Theta \times 3 \times (c_a + c_b + c_c - 3)$, which is $12 \times (c_a + c_b + c_c - 3)$. Also, as shown in Figure 4, nodes x, y, and z are duplication nodes that each also invoke one loss, and all the other nodes of $G^B(h_{r(j)})$ are speciations without any losses. Thus, the total reconciliation cost of $G^B(h_{r(j)})$ under DTL-scenario α is $12 \times (c_a + c_b + c_c - 3)$ plus the cost of three duplications and three losses, which is $12 \times (c_a + c_b + c_c - 3) + 9$.

For the other cases, note that for each set of "missing" subtrees g_i, g'_i, and g''_i, for $i \in \{a, b, c\}$, the reconciliation of $G^B(h_{r(j)})$ with S invokes two additional losses for the missing g'_i, and g''_i, and one less duplication for the missing g_i. Since $P_{loss} = 1$ and $P_\Delta = 2$, there is no net change on the total additive cost of 9. Thus, in cases 2, 3, and 4, the total cost is the sum of the reconciliation costs for the subtrees g_i, g'_i, and g''_i that are in $G^B(h_{r(j)})$, plus the additive cost of 9. □

Lemma 4. *The total reconciliation cost of subtree $G^B(h_{r(k)})$ with S under DTL-scenario α is the same as given in Lemma 3 but with an additional additive cost of $4 \times (3m - 3k)$.*

Proof. The proof for this lemma proceeds identically to that of Lemma 3, depending on whether $G^B(h_{r(k)})$ falls under case 1, 2, 3, or 4. However, $G^B(h_{r(k)})$ contains a additional subtree of $(3m - 3k)$ unassigned blue nodes (see Figure 3) and there is an additional cost associated with that subtree. As shown in Figure 3c, this subtree introduces $3m - 3k$ additional internal nodes to $G^B(h_{r(k)})$. Under DTL-scenario α, each of these $3m - 3k$ internal nodes is a transfer node (and there are no duplications or losses). This contributes an additive reconciliation cost of $P_\Theta \times (3m - 3k)$ to the reconciliation cost of $G^B(h_{r(k)})$. □

Thus, the reconciliation cost of any subtree $G^B(h_{r(j)})$ depends only on the total reconciliation cost of the subtrees $G^B(g_i)$, $G^B(g'_i)$, and $G^B(g''_i)$, for each $1 \leq i \leq n$, within $G^B(h_{r(j)})$ plus an additive cost of 9. In addition, there is an added cost of $4 \times (3m - 3k)$ for the subtree $G^B(h_{r(k)})$.

The following lemma implies the forward direction of Claim 1.

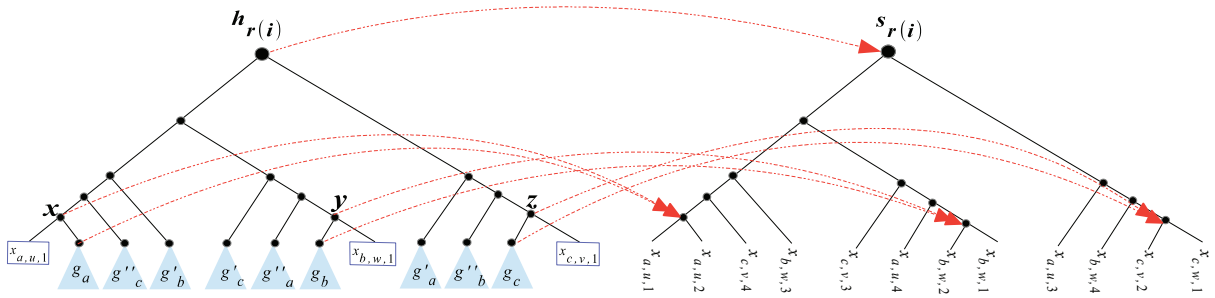

Fig. 4. **Mapping of subtree $G^B(h_{r(j)})$ to $S(s_{r(j)})$.** As the figure shows, nodes x, y, and z are duplication nodes that each invoke one loss. All the other nodes of $G^B(h_{r(j)})$ are speciation nodes without any losses.

Lemma 5. *Any MPR of G^B with S must have reconciliation cost at most $k + 48m - 12n$.*

Proof. Since α is a valid DTL-scenario, an MPR of G^B with S cannot have reconciliation cost more than that implied by α. Thus, it suffices to show that the DTL-scenario α has a reconciliation cost of exactly $k+48m-12n$. The total reconciliation cost under α is the sum of the reconciliation costs for each subtree $G^B(h_{r(j)})$, for $1 \leq j \leq k$, and the reconciliation cost implied by the backbone of G^B that connects these k subtrees.

Consider the k $G^B(h_{r(j)})$'s. Note that there are exactly n g_i's, g_i''s and g_i'''s distributed among these k subtrees. Thus, by Lemmas 3 and 4, the total reconciliation cost of the k subtrees is $12\times\sum_{i=1}^{n}(c_i-1)+9\times k+4\times(3m-3k)$. Since $\sum_{i=1}^{n} c_i = 3m$, this evaluates to $48m - 12n - 3k$.

Now consider the backbone of G^B that connects the k $G^B(h_{r(j)})$'s (see Figure 3). According to DTL-scenario α, for each $j \in \{1, \ldots, k-1\}$, the node $pa(h_{r(j)})$ is a transfer node. In addition, the root node of G^B is also a transfer node. Moreover, according to the mapping defined by α, this backbone does not invoke any losses. Thus, the backbone contributes a total of $P_\Theta \times k$, which is $4k$, to the total reconciliation cost.

The total reconciliation cost of G^B with S under DTL-scenario α is thus $48m - 12n - 3k + 4k$, which is $k + 48m - 12n$. □

3.5 Proof of Claim 1: reverse direction

Conversely, let us assume that we have a YES answer for the OGTR instance λ with $l = k + 48m - 12n$. We will show that there exists a solution of size at most k for the set cover instance ϕ. We first characterize the structure of optimal resolutions and their most parsimonious reconciliations.

Lemma 6. *For any optimal binary resolution G^B of G^N, all MPRs of G^B with S must satisfy the following:*

1) *Each node in $I(G^B)$ maps to either the start node or to a node in the subtree $S(s_j)$, for some $j \in \{1, \ldots, m\}$.*
2) *Each subtree $G^B(g_i)$, $G^B(g_i')$, or $G^B(g_i'')$, where $1 \leq i \leq n$, has at least $(c_i - 1)$ transfer nodes.*

Proof. Part (1). Suppose there exists a minimum-cost DTL-scenario α for G^B and S such that, under α, there exists a node in $I(G^B)$ that does not map to the *start* node or to a node in the subtree $S(s_j)$, for any $j \in \{1, \ldots, m\}$. We will show how to construct an alternative DTL-scenario β with lower reconciliation cost, leading to a contradiction.

Note that the set $V(S)\setminus(\cup_{i=1}^{m} V(S(s_i))\cup start)$ consists of three types of nodes: (i) the set of extra leaves added to each species tree branch (9 per branch), (ii) the set of internal nodes created by adding the extra leaves, and (iii) the rest of the nodes (each representing a branching point in the induced species tree without the added extra leaves). We will refer to these as *extra-leaf* node, *extra* nodes, and *backbone* nodes, respectively. Note that, by the definition of DTL scenarios, none of the nodes of $I(G^B)$ can map to an extra-leaf node. They may, however, map to extra nodes or backbone nodes. We will first show how to modify α into a new DTL-scenario α' with the same or lower reconciliation cost such that no node of $I(G^B)$ maps to an extra node.

Modifying mappings to extra nodes. Suppose $I(G^B)$ contains nodes that map to extra nodes under the DTL-scenario α. Let a denote such a node. If there is more than one such node of G^B, then a is chosen to be a node that does not have any descendants that map to extra nodes. Let b denote the node of S to which a maps. Let c denote the closest descendant of b that is not an extra node (or an extra-leaf node). Thus, c must either be an s_i, for $1 \leq i \leq m$, or a backbone node. Note that, by definition, a cannot be a speciation node. However, it may be a duplication or a transfer, yielding the following two cases.

Case 1. a is a duplication: Since no descendant of a maps to an extra node, we can change its mapping from b to c. The node a still remains a duplication node, and this change does not create any additional duplications, transfers, or losses. In fact, the number of losses is reduced by at least one since there are no longer any losses of the duplicated lineage along the path from b to c.

Case 2. a is a transfer: As in the previous case, since no descendant of a maps to an extra node, we can change its mapping from b to c. The node a remains a transfer node,

and this change does not create any additional duplications, transfers, or losses. Note that, if the node $pa(a)$ exists and maps either to b or an ancestor of b, then there is no reduction in the number of losses. And similarly, if the node $pa(a)$ does not exist or does not map either to b or to an ancestor of b, then the number of losses reduces by at least one.

Thus, in both cases, there is no increase in the reconciliation cost. We can apply this procedure iteratively to each node a in G^B that maps to an extra node, resulting in a new DTL-scenario α' that has either the same or lower reconciliation cost, and in which none of the nodes of G^B map to an extra node. If the reconciliation cost of α' is smaller than that of α, then we have a contradiction and the proof finishes. If the two costs are the same, one of the following two cases must hold: (i) There were no nodes in $I(G^B) \setminus \{rt(G^B)\}$ that mapped to an extra node under α (and thus $\alpha' = \alpha$, or (ii) all the candidate a's were transfer events and moreover, each a has a parent $pa(a)$ that maps to a node along the path from b to $rt(S)$. In either case, there must be at least one node in $I(G^B) \setminus \{rt(G^B)\}$ that maps to a backbone node under α'.

Next, we show how to further modify DTL-scenario α' into DTL-scenario β by modifying the mappings to the backbone nodes.

Modifying mappings to backbone nodes. Let a be a node from $I(G^B)$ that maps to a backbone node under DTL-scenario α'. If there is more than one such node of G^B, then a is chosen to be a node that does not have any descendants that map to backbone nodes. Let b denote the backbone node of S to which a maps. We now have three cases depending on whether a is a speciation, duplication, or transfer.

Case 1. a is a speciation: In this case, one child of a must map to a node in subtree $S(s_i)$ and the other child to a node in the subtree $S(s_j)$, where $1 \leq i, j \leq m$, and $i \neq j$. Moreover s_i and s_j must both be descendants of b. We will change the mapping to a from b to s_i. The node a now becomes a transfer node and the DTL-scenario remains valid. With this change, the number of transfers increases by 1, and the number of losses decreases by at least 9 (since there is one fewer loss at each of the extra nodes along the path from b to s_i). Thus, overall, the reconciliation cost decreases by at least $9 \times P_{loss} - 1 \times P_{\Theta}$, which is 5.

Case 2. a is a duplication: In this case, one child of a must map to a node in subtree $S(s_i)$ and the other child to a node in the subtree $S(s_j)$, where $1 \leq i, j \leq m$, and i may be the same as j. Moreover s_i and s_j must both be descendants of b. We will change the mapping to a from b to s_i. The node a now becomes either a transfer node, if $i \neq j$, or remains a duplication node if $i = j$, and the DTL-scenario remains valid. With this change, the number of losses decreases by at least 9 (since there is one fewer loss at each of the extra nodes along the path from b to s_i), while the number of transfers may increase by one with a corresponding decrease in one duplication. Thus, overall, the reconciliation cost decreases by at least $9 \times P_{loss} - 1 \times (P_{\Theta} - P_{\triangle})$, which is 7.

Case 3. a is a transfer: In this case, one child of a must map to a node in subtree $S(s_i)$ and the other child to a node in the subtree $S(s_j)$, where $1 \leq i, j \leq m$ and $i \neq j$, such that s_i is a descendant of b while s_j is neither a descendant nor an ancestor of b. We will change the mapping to a from b to s_i. The node a remains a transfer node and the DTL-scenario remains valid. In this case, if the node $pa(a)$ exists and maps either to b or an ancestor of b, then there is no reduction in the number of losses. But if the node $pa(a)$ does not exist or does not map either to b or to an ancestor of b, then the number of losses, and the reconciliation cost, reduces by at least 9.

We can apply this procedure iteratively to each node a in G^B that maps to a backbone node, resulting in a new DTL-scenario β that has reconciliation cost no greater than that of α. In particular, if any of the a's are duplications or speciations, then the new DTL-scenario β has a cost smaller than that of α and we have a contradiction. Similarly, if any of the a's are transfers such that their parent node does not map to b or its ancestor, then β must have cost smaller than that of α. Therefore, assume that none of the a's is a speciation or duplication, and that the parent of any given a maps to b or its ancestor. Under this assumption, as we iterate through all the candidate a's we eventually reach an a for which $pa(a)$ is $rt(G^B)$. If $rt(G^B)$ maps to the *start* node then, we are done, since then updating a's mapping will reduce the reconciliation cost by at least 9. Otherwise, if $rt(G^B)$ maps to either b or its ancestor, then we can update the mapping of $rt(G^B)$ to be the same as the mapping of a (i.e., to s_i). With this change, $rt(G^B)$ becomes a transfer node, irrespective of its previous event-type, and the DTL-scenario remains valid. This would result in a reduction of at least $9 - P_{\Theta} = 5$ in the reconciliation cost.

Thus, the reconciliation cost under β would be strictly smaller than the reconciliation cost under α, leading to a contradiction.

Part (2). Consider any g'_i, for $1 \leq i \leq n$. $G^B(g'_i)$contain c_i leaves and $(c_i - 1)$ internal nodes, and each of the c_i leaves maps to a different subtree $S(s_j)$, for $1 \leq j \leq m$. We will show that all $(c_i - 1)$ internal nodes of $G^B(g'_i)$ must be transfers. Suppose not. Then there must be an internal node a in $G^B(g'_i)$ that is not a transfer node. Without loss of generality assume that that a is such that all of its internal node descendants are transfers. By the part (1) of this lemma, we know that each node of G^B maps either to a node in $S(s_j)$, for $1 \leq j \leq m$ or to the *start* node. Now, since each leaf node maps to a different $S(s_j)$, for $1 \leq j \leq m$, the two children of a must also map to two different subtrees $S(s_j)$, for $1 \leq j \leq m$. Therefore, if a is either a speciation or duplication, it must map to a node that

is neither in one of the $S(s_j)$'s nor the *start* node, which is a contradiction.

The proof for g_i'' is identical to the one for g_i'. For g_i, observe that there are c_i of the $y_{i,\cdot}$'s and each of the $y_{i,\cdot}$'s contains exactly two leaves that both map to the same subtree $S(s_j)$, for $1 \leq j \leq m$. Moreover, the two leaves of each distinct $y_{i,\cdot}$ both map to a distinct subtree $S(s_j)$, for $1 \leq j \leq m$. Thus, each of the $y_{i,\cdot}$'s must themselves map to distinct subtrees $S(s_j)$, for $1 \leq j \leq m$. Based on this observation, the proof for g_i also follows along the same lines as the proof for g_i'. □

For the next few lemmas we need the following two definitions:

Definition 4 (Most recent Ancestral Transfer)**.** *Given a DTL-scenario α for G^B and S, and any node $a \in V(G^B)$, we define the* Most Recent Ancestral Recipient *node of a, denoted MRAR(a), to be the first node x along the path from a to rt(G^B) that $(pa(x), x) \in \Xi$ (i.e., x is the recipient of a transfer event). Note that not all $a \in V(G^B)$ have an MRAR node.*

Definition 5 (Canonical optimal resolution and MPR)**.** *Consider an optimal resolution G^B of G^N and an MPR, represented by DTL-scenario α, of G^B with S. We say that G^B and the MPR implied by α are both* canonical *if the node rt(G^B) maps to the start node in S.*

Not all optimal resolutions G^B and their MPRs are canonical. However, as we show next, any given optimal resolution G^B and its MPR α that are not canonical can be converted into a canonical resolution $G^{B'}$ and canonical MPR α', without any change in reconciliation cost.

Lemma 7. *Consider an optimal binary resolution G^B of G^N along with its MPR with S, represented by DTL-scenario α. If G^B and its MPR α are not canonical, then it is possible to efficiently compute a canonical optimal resolution $G^{B'}$ and a canonical MPR, α' of $G^{B'}$ with S.*

Proof. Since G^B and its MPR α are not canonical, it follows from Lemma 6(1) that *rt*(G^B) must map to $S(s_i)$, for some $i \in \{1, \ldots, m\}$. We will show how to create an alternative binary resolution $G^{B'}$ of G^N and an MPR α' of $G^{B'}$ with S, with the same reconciliation cost such that *rt*$(G^{B'})$ maps to the *start* node. Since *rt*(G^B) does not map to the *start* node, the *start* node must have an *MRAR*. We perform a subtree-prune-and-regraft operation on G^B as follows: We prune the subtree $G^B(\textit{MRAR}(\textit{start}))$ and regraft it above the root of the remainder of G^B, thereby creating a new root node in the resulting tree. Thus, the resulting tree, $G^{B'}$, has a root node whose children are the roots of the subtrees $G^B(\textit{MRAR}(\textit{start}))$ and $G^B(\textit{rt}(G^B)))\setminus G^B(\textit{MRAR}(\textit{start}))$. The DTL-scenario α' for $G^{B'}$ and S is identical to that for G^B and S, except that, the edge from *rt*$(G^{B'})$ to $G^B(\textit{rt}(G^B)))\setminus G^B(\textit{MRAR}(\textit{start}))$ is designated as a transfer edge, and *rt*$(G^{B'})$ is assigned the same mapping as that for *MRAR*(*start*) in G^B. The resulting DTL-scenario remains valid and has the same reconciliation cost as the original since we simply remove the transfer edge $(\textit{pa}(\textit{MRAR}(\textit{start})), \textit{MRAR}(\textit{start}))$ in G^B and replace it with another. Observe that *rt*$(G^{B'})$ must now map to the *start* node resulting in a canonical binary resolution and its canonical MPR. Also observe that this construction has time complexity linear in the size of G^B. □

Lemma 8. *Given any canonical optimal binary resolution G^B of G^N and a canonical MPR of G^B with S, each node in $V(G^B)$ that maps to a node of $S(s_j)$, for any $1 \leq j \leq m$, must have an MRAR node.*

Proof. For contradiction, suppose there exists an $S(s_j)$, where $1 \leq j \leq m$, such that at least one of the nodes of G^B that maps to $S(s_j)$ doesn't have an *MRAR*. Since G^B and its given MPR are canonical, *rt*(G^B) must map to the *start* node. Consider all those nodes of G^B that map to $S(s_j)$ but do not have any ancestors that map to $S(s_j)$. From Lemma 6(1), it follows that all such nodes must be recipients of transfer events. Since all other nodes of G^B that map to $S(s_j)$ must descend from one such node in G^B, the lemma follows.

□

Lemma 9. *Consider any subtree $S(s_j)$, for $1 \leq j \leq m$, of the species tree, and consider its three leaf nodes with labels of the form $x_{\cdot,\cdot,1}$. There are exactly three blue nodes in the gene tree that must map to these three leaf nodes of $S(s_j)$. Let these three blue nodes be denoted by a, b, and c. Given any canonical optimal binary resolution G^B of G^N and a canonical MPR of G^B with S, if there are no nodes g_i, g_i', or g_i'', for any $i \in \{1, \ldots, n\}$, that map to a node of $S(s_j)$, then the MRAR's for a, b, and c must all be distinct and must map to nodes of $S(s_j)$.*

Proof. By Lemma 3.5 we know that each blue node has an *MRAR* in G^B. Note that each of these *MRAR* nodes must map to a node of the subtree $S(s_j)$ to which its blue node maps. We therefore separate our analysis into three cases: (i) There is only one node that is an *MRAR* for a, b, and c, (ii) there are two nodes that are *MRAR*'s for a, b, and c, and (iii) there are three nodes that are *MRAR*'s for a, b, and c. If case (iii) holds, then we are done, since each of those three *MRAR*'s must map to a node of $S(s_j)$. We will consider each of these three cases:

Case 1. If a, b, and c have the same *MRAR*, say x, then $x \geq lca_{G^B}(a, b, c)$. Let y denote $lca_{G^B}(a, b, c)$. Since there are no *MRAR*'s within the subtree $G^B(y)$, each node of $G^B(y)$ must map to $S(s_j)$, and y and x must both map to s_j. Without loss of generality, let z denote $lca_{G^B}(a, b)$ such that $z < y$. Observe that, since none of the g_i's, g_i''s

or g_i'''s map to $S(s_j)$, a, b, and c are the only leaves of $G^B(y)$ that map to leaves in $S(s_j)$. This implies that all of the internal nodes along the paths from y to a, b, and c, except for nodes y and z must be transfer nodes. The observation also implies that subtree $G^B(y)$ must induce at least 9 losses in $S(s_j)$. Furthermore, each node along the path from x to y must itself be a transfer node for the same reason.

We will now show how to create an alternative DTL-scenario α' with smaller reconciliation cost than α, leading to a contradiction. We update the mappings of all internal nodes along the path from a to x (including x) to be the mapping of a, all nodes along the path from b to z (not including z) to be the mapping of b, and the mapping of all internal nodes along the path from c to y (not including y) to be the mapping of c. The resulting DTL-scenario remains valid, and only introduces two additional transfer nodes, y and z, and no additional losses. This is because all existing transfer nodes on the paths remain valid transfer nodes, and changing the mapping of the *MRAR* node does not lead to any increase in the number of losses (only the recipient node of the transfer event changes). Since this update decreases the number of losses by 9, the new DTL-scenario α' must have a reconciliation cost that is lower than the original α by $9 \times P_{loss} - 2 \times P_{\Theta} = 1$. A contradiction.

Case 2. If there are two nodes that are *MRAR*'s for a, b, and c, then two of the blue nodes, say a and b must have the same *MRAR*. Let x denote the *MRAR* of a and b, y denote $lca_{G^B}(a, b)$, and x' denote *MRAR*(c). Then, $x \geq y$, and each node along the paths from a to y and b to y must map to $S(s_j)$. Note that the subtree $G^B(y)$ must invoke at least 6 losses in $S(s_j)$. We will show that, in spite of the relative arrangement of a, b, c, y, x, and x', all internal nodes along the paths from a to y (not including y), b to y (not including y), c to x' (including x'), and y to x (including x, unless $x = y$) must be transfer nodes.

Consider the path a to y. Suppose there is an internal node, say z, where $z \neq y$, along this paths that is not a transfer node. Then z must be a speciation or duplication node. Let z' denote the child of z that is not on the a to y path. Since z maps to $S(s_j)$, so must z', and z' must therefore have at least one leaf descendant that maps to $S(s_j)$. The node c is the only possible candidate for this leaf descendant. Thus, the path from z' to c cannot contain any transfer edges. This implies that $x' \geq z$, which is a contradiction, since *MRAR*$(a) = x$ and *MRAR*$(c) \neq x$. A completely analogous argument also establishes that each node except y along the path from y to x must be a transfer node. Finally, consider the path c to x'. As before, suppose there is an internal node, say z, along this paths that is not a transfer node. Then z must be a speciation or duplication node. Let z' denote the child of z that is not on the c to x' path. Since z maps to $S(s_j)$, so must z', and z' must therefore have at least one leaf descendant that maps to $S(s_j)$. a and b are the only two possible candidates for this leaf descendant. Note, however, that any path from z' to a or b must go through the node x (since *MRAR*$(a) =$ *MRAR*$(b) = x$ and *MRAR*$(c) = x'$). Thus, the path from z' to a or b travels through a transfer edge, implying that z' cannot have either a or b as descendants, a contradiction. This proves that all internal nodes along the paths from a to y (not including y), b to y (not including y), c to x' (including x'), and y to x (including x, unless $x = y$) must be transfer nodes.

We will now show how to create an alternative DTL-scenario α' with smaller reconciliation cost than α, leading to a contradiction. We update the mappings of all internal nodes along the path from a to x (including y) to be the mapping of a, all nodes along the path from b to y (not including y) to be the mapping of b, and all nodes along the path from c to x' (including x') to be the mapping c. The resulting DTL-scenario remains valid, and only introduces one additional transfer node, y, and no additional losses. This is because all existing transfer nodes on the paths remain valid transfer nodes, and changing the mapping of the two *MRAR* nodes does not lead to any increase in the number of losses (only the recipient node for the transfer event changes). Since this update decreases the number of losses by at least 6, the new DTL-scenario α' must have a reconciliation cost that is lower than the original by at least α by $6 \times P_{loss} - 1 \times P_{\Theta} = 2$. A contradiction. □

The next lemma places a lower bound on the reconciliation cost of any optimal binary resolution G^B of G^N.

Lemma 10. *For any canonical optimal binary resolution G^B of G^N and a canonical MPR of G^B with S, if the nodes g_i and g_i' and g_i'', for each $i \in \{1, \ldots, n\}$, map to exactly k distinct subtrees $S(s_j)$, for $1 \leq j \leq m$, then the reconciliation cost of G^B with S is at least $k + 48m - 12n$.*

Proof. From Lemma 6(1) we know that each of the subtrees g_i and g_i' and g_i'' has $c_i - 1$ transfer nodes. This contributes a total of $3 \times (3m - n)$ transfer edges. Similarly, from Lemma 3.5, we know that all nodes, labeled $x_{i,\cdot,\cdot}$, for any $i \in \{1, \ldots, n\}$ that map to subtrees $S(s_j)$ other than the k chosen ones, must have a distinct *MRAR* This contributes another $(3m - 3k)$ transfer edges. Also, from Lemma 6(1), it follows that all of the nodes of G^B that map to the k chosen $S(s_j)$'s, must have at least one *MRAR*, giving a total of k additional transfers. The total reconciliation cost due to these transfers is $4 * 3(3m - n) + 4(3m - 3k) + 4k$, which is $48m - 12n - 8k$. To complete the proof it suffices to show that the remainder of the reconciliation cost is at least $9k$.

Specifically, we will show that, for each of the k chosen subtrees $S(s_j)$, the nodes of G^B that map to $S(s_j)$ contribute an average additional cost of at least 9 through either losses, duplications, or uncounted transfers. Note that the nodes $g'(i)$ and $g''(i)$ may each prevent a single loss

event. We will initially ignore the presence of the $g'(i)$'s and g''_i's when counting losses for any given $S(s_j)$, but we will reduce the total number of losses obtained from our analysis by $2n$ later.

We first consider those $S(s_j)$ that have a mapping from one or more g'_i or g''_i, but not from $g(i)$, and calculate the minimum additional cost induced. Let $S(s_j)$ be a subtree that has mappings from one or more g'_i or g''_i, but not from $g(i)$. We distinguish 3 cases, depending on whether there are one, two, three distinct *MRAR*'s for the three blue nodes, denoted a, b, and c.

Case 1: If a, b, and c share the same *MRAR*, then this *MRAR* node must map to s_j and must induce 9 losses along the paths from the *MRAR* to a, b, and c (since there are no g_i's and we ignore $g'(i)$'s and g''_i's when counting losses).

Case 2: If a, b, and c have two distinct *MRAR*'s then two of the blue nodes, say a and b must share an *MRAR*, denoted x. The paths from x to a and b must thus induce 6 losses (since there are no g_i's and we ignore $g'(i)$'s and g''_i's when counting losses). Also, since we have only counted one *MRAR* (transfer event) per S_j in the analysis above, there is one additional *MRAR* in this case, giving an additional cost of 4 for its transfer event. The total additional cost in this case is thus 10, which is greater than 9.

Case 3: If a, b, and c have three distinct *MRAR*'s then we consider two further cases: In the first case, suppose that one of the g'_i's or g''_i's that map to $S(s_j)$ have an *MRAR* that is different than the three *MRAR*'s for a, b, and c. This means that there are at least 4 distinct *MRAR*'s that map to $S(s_j)$, only one of which has been counted before. This yields an additional cost of 12 for the remaining three transfers, and we again have a cost of at least 9. In the second case, there are only three *MRAR*'s for a, b, c, and the g'_i's and g''_i's. There must thus be shared *MRAR*, denoted x for one of the blue nodes, say a, and a g'_i or g''_i. The path from x to a must induce at least one loss (since there are no g_i's). Thus, in thuis case we have two uncounted *MRAR*'s (transfers) and at least one additional loss, yielding an additional cost of at least 9.

Thus, the nodes of G^B that map to an $S(s_j)$ that has a mapping from one or more g'_i or g''_i, but not from $g(i)$, contribute at least an additional cost of 9.

We now consider all other $S(s_j)$, i.e. all $S(s_j)$'s that have mappings from one or more g_i's. Observe that for each g_i that maps to S_j, the nodes of G^B mapping to $S(s_j)$ must either induce an additional duplication event or an additional transfer event. This contributes a cost of at least 2 for each g_i, thus contributing at least $2n$ overall. Let $S(s_j)$ be a subtree that has mappings from at least one $g(i)$. The computation of contributed loss costs due to $S(s_j)$ is analogous to that shown above (cases 1, 2, and 3, with only minor variation) and again shows that the nodes of G^B that map to an $S(s_j)$ that has a mapping from at least one $g(i)$, contribute at least an additional cost of 9.

The total additional cost over all the k $S(s_j)$'s is thus at least $9k$, plus at least $2n$ for the duplications or additional transfers caused by the g_i's, and minus at most $2n$ for the losses prevented by the g'_is and g''_i's, i.e., $9k$. This completes the proof. □

The following lemma establishes the reverse direction of Claim 1.

Lemma 11. *If there exists an optimal binary resolution of G^N such that its MPR with S has reconciliation cost at most $k + 48m - 12n$, then there exists a solution of size at most k for the M3SC instance ϕ.*

Proof. Consider an optimal binary resolution G^B such that its MPR with S has reconciliation cost at most $k + 48m - 12n$. We will assume that both G^B and its MPR are canonical. (If not, we can use the efficient constructive procedure from the proof of Lemma 3.5 to create a canonical resolution and a canonical MPR with the same reconciliation cost.) We can obtain a solution for the M3SC instance as follows: Choose the set A_j to be in the set cover, for $j \in \{1 \ldots, m\}$, if and only if the subtree $S(s_j)$ has a mapping from at least one of the g_i's, g'_i's, or g''_i's, for $i \in \{1 \ldots, n\}$.

We first show that this yields a valid set cover. From Lemma 6(2) it follows that g_i, g'_i, or g''_i, for any given $i \in \{1, \ldots, n\}$, can only map to a subtree $S(s_j)$, for $j \in \{1, \ldots m\}$ that contains leaves with labels of the form $x_{i,\cdot,\cdot}$, i.e., at least one leaf in the subtree $G^B(g_i)$, $G^B(g'_i)$, or $G^B(g''_i)$ must map to that $S(s_j)$. The subtree $S(s_j)$ contains leaves with labels of the form $x_{i,\cdot,\cdot}$ if and only if the set A_j in the M3SC instance ϕ contains element u_i. Finally, since g_i, g'_i, and g''_i, for each $i \in \{1, \ldots, n\}$ must map to an $S(s_j)$, for some $j \in \{1, \ldots, m\}$, it follows that the chosen A_j's would cover all the elements $u_1, u_2, \ldots, u_n\}$.

We now show that the size of the resulting solution for the M3SC instance ϕ has size at most k. Suppose, for contradiction, that the size is k', where $k' > k$. This means that there must be k' subtrees $S(s_j)$, where $j \in \{1, \ldots, m\}$, that receive mappings from at least one of the g_i's, g'_i's, or g''_i's, for $i \in \{1 \ldots, n\}$. However, from Lemma 10, we know that the MPR of G^B with S must then have a cost of at least $k' + 48m - 12n$, which is strictly greater than $k + 48m - 12n$. A contradiction. Thus, there must be at most k subtrees $S(s_j)$, where $j \in \{1, \ldots, m\}$, that receive mappings from at least one of the g_i's, g'_i's, or g''_i's, for $i \in \{1 \ldots, n\}$, completing the proof. □

4 Extension to dated DTL reconciliation

An alternative model of DTL reconciliation has been proposed when the internal nodes of the species tree can be fully ordered in time [10]. We refer to this model as the

Dated-DTL reconciliation model. Dated-DTL reconciliation makes use of the total order on the species nodes to ensure that the reconstructed optimal reconciliation is time-consistent. A key feature of this model is that it subdivides the species tree into different *time slices* [10] and then restricts transfer events to only occur within the same time slice.

We show how to assign divergence times to each node of the species tree. Observe that all subtrees $S(s_i)$, for each $i \in \{1 \dots m\}$, have identical structure. All nodes at the same level in each $S(s_i)$ are assigned the same divergence time across all the subtrees. The *start* node is assigned to be at the same level as the other leaves of S. The rest of the nodes in S may be assigned arbitrary divergence times respecting the topology of S. Under this divergence time assignment, it can be shown that there exists an optimal resolution of the gene tree for which an MPR exists that only invokes transfer events that respect the timing constraints of the dated species tree as required by the dated-DTL reconciliation model. This implies that, for our gadget, any optimal resolution of the gene tree under the undated DTL reconciliation model has the same minimum reconciliation cost as the dated-DTL reconciliation model.

Theorem 2. *The OGTR problem under the dated-DTL reconciliation model is NP-hard.*

Proof. Consider the DTL-scenario α described in Section 3.4 to prove the forward direction of the proof. Note that all transfer events invoked by α occur within the same time-slice of the dated species tree described above, as required by the dated-DTL reconciliation model. Thus, even for the dated case, any MPR has cost at most $k + 48m - 12n$. Moreover, since the reconciliation cost under dated-DTL reconciliation cannot be smaller than that under DTL reconciliation, Claim 1 must also apply under dated-DTL reconciliation. This completes the proof. □

5 Conclusion

In this work, we have shown that the OGTR problem, i.e., the problem of reconciling non-binary gene trees with binary species trees under the DTL reconciliation model, is NP-hard. Our reduction applies to both the undated and dated formulations of DTL-reconciliation and, furthermore, shows that the problem is NP-hard even for a biologically meaningful event cost assignment of 1, 2, and 4 for losses, duplications, and transfers, respectively. The uncertainty about its complexity has prevented the development of algorithms for the OGTR problem. This work will spur the development of effective exact, approximate, and heuristic algorithms for this problem, making it possible to apply the powerful DTL reconciliation framework to non-binary gene trees.

Funding: This work was supported in part by startup funds from the University of Connecticut to MSB.

References

[1] Koonin, E.V.: Orthologs, paralogs, and evolutionary genomics. Annual Review of Genetics **39**(1) (2005) 309–338

[2] Vilella, A.J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R., Birney, E.: Ensemblcompara genetrees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Research **19**(2) (2009) 327–335

[3] Chen, K., Durand, D., Farach-Colton, M.: Notung: dating gene duplications using gene family trees. In: RECOMB. (2000) 96–106

[4] David, L.A., Alm, E.J.: Rapid evolutionary innovation during an archaean genetic expansion. Nature **469** (2011) 93–96

[5] Durand, D., Halldórsson, B.V., Vernot, B.: A hybrid micro-macroevolutionary approach to gene tree reconstruction. J. Comput. Biol. **13**(2) (2006) 320–335

[6] Burleigh, J.G., Bansal, M.S., Eulenstein, O., Hartmann, S., Wehe, A., Vision, T.J.: Genome-scale phylogenetics: Inferring the plant tree of life from 18,896 gene trees. Syst. Biol. **60**(2) (2011) 117–125

[7] Scornavacca, C., Jacox, E., Szllosi, G.J.: Joint amalgamation of most parsimonious reconciled gene trees. Bioinformatics (in press)

[8] Bansal, M.S., Wu, Y.C., Alm, E.J., Kellis, M.: Improved gene tree error correction in the presence of horizontal gene transfer. Bioinformatics **31**(8) (2015)

[9] Gorbunov, K.Y., Liubetskii, V.A.: Reconstructing genes evolution along a species tree. Molekuliarnaia Biologiia **43**(5) (2009) 946–958

[10] Doyon, J.P., Scornavacca, C., Gorbunov, K.Y., Szöllosi, G.J., Ranwez, V., Berry, V.: An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In Tannier, E., ed.: RECOMB-CG. Volume 6398 of Lecture Notes in Computer Science., Springer (2010) 93–108

[11] Tofigh, A., Hallett, M.T., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM Trans. Comput. Biology Bioinform. **8**(2) (2011) 517–535

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

[13] Stolzer, M., Lai, H., Xu, M., Sathaye, D., Vernot, B., Durand, D.: Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics **28**(18) (2012) 409–415

[14] Bansal, M.S., Alm, E.J., Kellis, M.: Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss. Journal of Computational Biology **20**(10) (2013) 738–754

[15] Scornavacca, C., Paprotny, W., Berry, V., Ranwez, V.: Representing a set of reconciliations in a compact way. Journal of Bioinformatics and Computational Biology **11**(02) (2013) 1250025

[16] Libeskind-Hadas, R., Wu, Y.C., Bansal, M.S., Kellis, M.: Pareto-optimal phylogenetic tree reconciliation. Bioinformatics **30**(12) (2014) i87–i95

[17] Ovadia, Y., Fielder, D., Conow, C., Libeskind-Hadas, R.: The cophylogeny reconstruction problem is NP-complete. J. Comput. Biol. **18**(1) (2011) 59–65

[18] Libeskind-Hadas, R., Charleston, M.: On the computational complexity of the reticulate cophylogeny reconstruction problem. J. Comput. Biol. **16** (2009) 105–117

[19] Daubin, V., Gouy, M., Perriere, G.: A phylogenomic approach to bacterial phylogeny: Evidence of a core of genes sharing a common history. Genome Research **12**(7) (2002) 1080–1090

[20] Frédéric Delsuc, H.B..H.P.: Phylogenomics and the reconstruction of the tree of life. Nature Reviews Genetics **6** (2005) 361–375

[21] Chang, W., Eulenstein, O.: Reconciling gene trees with apparent polytomies. In Chen, D.Z., Lee, D.T., eds.: Computing and Combinatorics, 12th Annual International Conference, COCOON 2006, Taipei, Taiwan, August 15-18, 2006, Proceedings. Volume

4112 of Lecture Notes in Computer Science., Springer (2006) 235–244
[22] Lafond, M., Swenson, K., El-Mabrouk, N.: An optimal reconciliation algorithm for gene trees with polytomies. In Raphael, B., Tang, J., eds.: Algorithms in Bioinformatics. Volume 7534 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 106–122
[23] Zheng, Y., Zhang, L.: Reconciliation with non-binary gene trees revisited. In Sharan, R., ed.: Research in Computational Molecular Biology. Volume 8394 of Lecture Notes in Computer Science. Springer International Publishing (2014) 418–432
[24] Kordi, M., Bansal, M.S.: On the complexity of Duplication-Transfer-Loss reconciliation with non-binary gene trees. In Harrison, R., Li, Y., Mandoiu, I., eds.: Bioinformatics Research and Applications. Volume 9096 of LNCS. (2015) 187–198
[25] Karp, R.M.: Reducibility among combinatorial problems. In Miller, R.E., Thatcher, J.W., eds.: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York. The IBM Research Symposia Series, Plenum Press, New York (1972) 85–103

Misagh Kordi received the BS degree in computer science and engineering from Kharazmi University, Iran, in July 2010, and the MS degree in computer science and engineering from the University of Tehran, Iran, in July 2013. He is currently working towards the PhD degree in the Department of Computer Science and Engineering at the University of Connecticut, USA. His research interests include computational biology and phylogenetics, graph theory, complexity theory, approximation algorithms, and algorithms in general.

Mukul S. Bansal is currently an assistant professor with the Department of Computer Science and Engineering at the University of Connecticut, USA. His research interests are in computational biology and bioinformatics, with an emphasis on computational molecular evolution. He is especially interested in computational problems related to understanding the evolution of genes, genomes, and species. He received the PhD degree in computer science from Iowa State University in 2009. He was an Edmond J. Safra postdoctoral fellow at the School of Computer Science at Tel Aviv University in Israel until December 2010, and a postdoctoral associate at the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology until August 2013.