



The Most Parsimonious Reconciliation Problem in the Presence of Incomplete Lineage Sorting and Hybridization Is NP-Hard

Matthew LeMay ✉

Department of Mathematics, Harvey Mudd College, Claremont, CA, USA

Yi-Chieh Wu¹ ✉ 

Department of Computer Science, Harvey Mudd College, Claremont, CA, USA

Ran Libeskind-Hadas ✉ 

Department of Computer Science, Harvey Mudd College, Claremont, CA, USA

Abstract

The maximum parsimony phylogenetic reconciliation problem seeks to explain incongruity between a gene phylogeny and a species phylogeny with respect to a set of evolutionary events. While the reconciliation problem is well-studied for species and gene trees subject to events such as duplication, transfer, loss, and deep coalescence, recent work has examined species phylogenies that incorporate hybridization and are thus represented by networks rather than trees. In this paper, we show that the problem of computing a maximum parsimony reconciliation for a gene tree and species network is NP-hard even when only considering deep coalescence. This result suggests that future work on maximum parsimony reconciliation for species networks should explore approximation algorithms and heuristics.

2012 ACM Subject Classification Applied computing → Computational biology

Keywords and phrases phylogenetics, reconciliation, deep coalescence, hybridization, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.WABI.2021.1

Related Version *Previous Version:* <https://www.biorxiv.org/content/10.1101/2021.03.14.435321v1>

Funding *Matthew LeMay:* Supported by the National Science Foundation under Grant No. IIS-1751399.

Yi-Chieh Wu: Supported by the National Science Foundation under Grant No. IIS-1751399.

Ran Libeskind-Hadas: Supported by the National Science Foundation under Grant No. IIS-1905885.

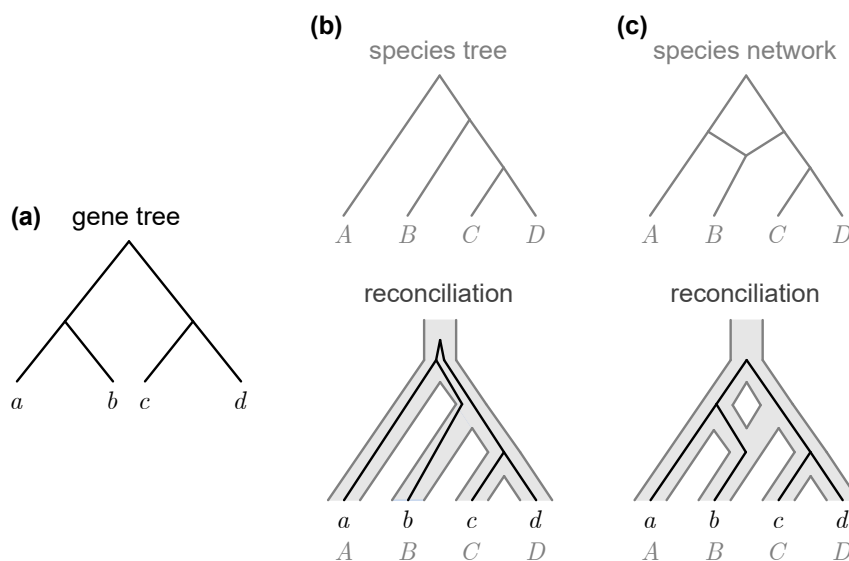
Acknowledgements The authors thank Adam Walker and the anonymous reviewers for valuable comments that helped improve the paper.

1 Introduction

Genes evolve via several evolutionary processes operating at various evolutionary timescales. Nucleotides can mutate, and domains can recombine. Genes can be generated, lost, or replaced through gene duplication, gene loss, horizontal gene transfer, and gene conversion. Populations can diverge or combine through speciation and hybridization. In addition to these events, within a population, polymorphisms can persist across speciation events, leading to a phenomenon known as *incomplete lineage sorting (ILS)* [15, 18]. Thus, the history of a set of genes may differ from the history of the species in which they evolved [12].

¹ Corresponding author





■ **Figure 1 Gene trees, species trees, and species networks.** (a) A gene tree. (b) A species tree and reconciliation. Under the multispecies coalescent model, the gene tree evolves within the species tree, and incongruence between the trees is due to ILS. (c) A species network and reconciliation. The same gene tree evolves within the species network, and no ILS is necessary.

In phylogenetics, *reconciliations* attempt to explain these differences by mapping gene histories within species histories to infer the evolutionary events that shaped that gene family (Figure 1a,b). The simplest and most common approach seeks a most parsimonious reconciliation (MPR) [7, 12, 14], in which each type of event in the model has an associated non-negative cost and the objective is to find a reconciliation of minimum total cost.

The time complexity of the MPR problem depends on the events being modeled and the set of constraints being considered. For example, the lowest common ancestor mapping, which can be computed in polynomial time [14, 26], solves the MPR problem when considering only duplications [8], only duplications and losses [8], and only deep coalescence [22]. When considering duplications, transfers, and losses, the MPR problem can be solved in polynomial time or is NP-hard depending on whether the species tree is undated, partially dated, or fully dated, and on whether the reconciliation is constrained to be time-consistent [13, 20]. Similarly, depending on details of the underlying model, the MPR problem can be solved in polynomial time when considering duplications, transfers, losses, and coalescence [4, 17], or is NP-hard when considering duplications, losses, and coalescence [2]. The MPR problem is also NP-hard when simultaneously modeling the evolution of domains, genes, and species [9].

Though the species history is often represented by a tree, hybridization is increasingly recognized as an important evolutionary process, requiring the use of species networks (Figure 1c). In eukaryotic species, hybridization encompasses two different processes: hybrid speciation, in which there is no underlying tree, and introgression, in which there is an underlying tree [5, 6]. Horizontal gene transfer in prokaryotic species can be considered a special case of introgression [16] and results in reticulate evolutionary histories as well.

Several authors have recently considered reconciliations with species networks. For example, several methods exist for the related problem of inferring a species network that minimizes deep coalescence [23, 24, 25]. However, the authors did not analyze the time complexity of their algorithms. Furthermore, their approaches require searching over the space of species network topologies, in contrast to the problem considered here, in which the species network is assumed to be known. Other authors have focused on the problem

of inferring MPRs between gene trees and species networks, showing that the problem can be solved in polynomial time when minimizing the duplication-transfer-loss cost [11] or the duplication-loss cost [19]. There is little previous work on the problem of inferring MPRs between a gene tree and species network in which incongruence is due to deep coalescence, and the question of whether this problem can be solved in polynomial time remained open.

In this paper, we show that the problem of inferring an MPR between a gene tree and species network in the presence of incomplete lineage sorting is NP-hard. Our results suggest that future work on this problem should focus on developing heuristics or approximation algorithms.

2 Definitions

We use the terms *node* and *vertex* interchangeably. A *rooted binary phylogenetic network* refers to a rooted directed acyclic graph with a single root with in-degree 0 and out-degree 2; additional internal nodes with either in-degree 1 and out-degree 2, called *branch nodes*, or in-degree 2 and out-degree 1, called *hybridization nodes*; and one or more leaves with in-degree 1 and out-degree 0. Edges leading to hybridization nodes are called *hybridization edges*. Given a network N , let $V(N)$ denote its node set and $E(N)$ denote its edge set. Let $L(N) \subset V(N)$ denote its leaf set, $I(N) = V(N) \setminus L(N)$ denote its set of internal nodes, and $r(N) \in I(N)$ denote its root node. For a node $v \in V(N)$, let $c(v)$ denote its set of children (the empty set if v is a leaf), let $p(v)$ denote its set of parents (the empty set if v is the root node), and, if v has a single parent, $e(v)$ denotes the edge $(p(v), v)$. The size of N , denoted by $|N|$, is equal to $|V(N)| + |E(N)|$.

Let \leq_N ($<_N$) be the partial order on $V(N)$ such that $v \leq_N u$ ($v <_N u$) if and only if there exists a path in N from u to v ($v \neq u$); v is said to be *lower or equal to* (lower than) u , and v a (strict) *descendant* of u , and u a (strict) *ancestor* of v .

Given two nodes u and v of N such that $v \leq_N u$, a path from u to v in N is a sequence of contiguous edges from u to v in N . Note that if $u = v$, the path from u to v is empty. As there can be multiple paths between pairs of vertices in a network, let $paths_N(u, v)$ denote the set of all paths from u to v . Let $paths(N)$ denote the set of all paths in network N .

A binary phylogenetic tree is a binary phylogenetic network with no hybridization nodes; that is, a directed binary tree. In the remainder of this paper, we refer to rooted binary phylogenetic networks and rooted phylogenetic trees simply as *networks* and *trees*, respectively.

A *species network* S represents the evolutionary history of a set of species and a *gene tree* G represents the evolutionary history of a set of genes sampled from these species. A *leaf mapping* $Le: L(G) \rightarrow L(S)$ associates each leaf in the gene tree with a corresponding species from which the gene was sampled. The mapping need not be one-to-one nor onto. Note that gene phylogenies are assumed to be trees whereas species phylogenies may, in general, be networks.

In this paper, we assume that both the gene tree and species network are undated. Thus, the only temporal constraints on the nodes are those induced by ancestor-descendant relationships.

2.1 Reconciliations

A *reconciliation* for a given gene tree, species network, and leaf mapping comprises a pair of mappings: The *vertex mapping* $R_v: V(G) \rightarrow V(S)$ associates each node of G with a node of S . For each non-root node g of G , the *path mapping* $R_p: V(G) \rightarrow paths(S)$ associates a path in S from $R_v(p(g))$ to $R_v(g)$. The vertex mapping must be consistent with the given leaf

mapping and must satisfy temporal constraints; namely if a gene node g is mapped to species node s and a child g' of g is mapped to species node s' , then s must be an ancestor of s' . The path mapping is required because S is a network, and thus there may be multiple paths between ancestors and descendants in the network. The formal definition of a reconciliation is given in Definition 1.

► **Definition 1** (Reconciliation). *Given a gene tree G , a species network S , and a leaf mapping Le , a reconciliation² R for (G, S, Le) is a pair of mappings (R_v, R_p) where $R_v : V(G) \rightarrow V(S)$ is a vertex mapping and $R_p : V(G) \rightarrow \text{paths}(S)$ is a path mapping subject to the following constraints:*

1. *If $g \in L(G)$, then $R_v(g) = Le(g)$.*
2. *If $g \in I(G)$, then for each $g' \in c(g)$, $R_v(g') \leq_S R_v(g)$.*
3. *If $g \neq r(G)$, then $R_p(g) \in \text{paths}_S(R_v(p(g)), R_v(g))$. Otherwise, $R_p(g) = \emptyset$.*

Constraint 1 asserts that R_v extends the leaf mapping Le . Constraint 2 asserts that R_v satisfies the temporal constraints implied by S . Constraint 3 asserts that the vertex mapping and path mapping are consistent. We note that some formulations of the reconciliation problem include an additional constraint asserting that no two paths in the path mapping use two different hybridization edges leading to the same hybridization node. While we do not explicitly enforce this constraint in Definition 1, the NP-hardness proof in the next section satisfies this additional constraint nonetheless.

In a multispecies coalescent process, evolution in the species network is viewed backward in time, from the leaves toward the root. Then, given a reconciliation R , we can count the number of gene lineages “passing through” each edge e of the species network. Specifically, given edge $e \in E(S)$,

$$\mathbf{L}_R(e) = |\{g \in V(G) : e \in R_p(g)\}|,$$

and the number of “extra lineages” is defined to be

$$\mathbf{XL}_R(e) = \max(0, \mathbf{L}_R(e) - 1).$$

Note, for example, that if two gene paths pass through a species edge, there is one extra lineage on that edge.

Finally, the *deep coalescence cost* of a reconciliation is the sum of extra lineages across all edges of the species network:

$$\mathbf{DC}_R = \sum_{e \in E(S)} \mathbf{XL}_R(e).$$

This value is the *reconciliation cost* in this model.

Finally, we formalize these optimization and decision problems:

► **Problem 2** (Most Parsimonious Reconciliation (MPR)). *Given a gene tree G , a species network S , and a leaf mapping Le , find a reconciliation R for (G, S, Le) such that the deep coalescence cost \mathbf{DC}_R is minimized.*

► **Problem 3** (Most Parsimonious Reconciliation Decision Problem (MPRD)). *Given a gene tree G , a species network S , a leaf mapping Le , and an integer k , is there a reconciliation R for (G, S, Le) such that $\mathbf{DC}_R \leq k$?*

² When explaining topological incongruence through only deep coalescence, a reconciliation is sometimes called a *coalescent history* [5].

3 NP-hardness

► **Theorem 4.** *MPRD is NP-hard.*

In the proof that follows, it will be convenient to consider the gene tree as the collection of all paths P from its root to its leaves. For a given leaf mapping $Le : L(G) \rightarrow L(S)$, a *lineage mapping* with respect to Le is a mapping M from each path $p_\ell \in P$ whose endpoint is leaf ℓ to a path in S from some fixed node $v \in V(S)$ to $Le(\ell)$. Each reconciliation has a corresponding lineage mapping, M . Specifically, let $R = (R_v, R_p)$ be a reconciliation and let $r(G), g_1, \dots, g_k, g_k = \ell$, denote the nodes on the unique path from $r(G)$ to leaf ℓ . Then, M associates this path in G with path $R_p(g_1)R_p(g_2) \dots R_p(g_k)$ in S . Note that multiple different reconciliations may induce the same lineage mapping since there are, in general, different mappings of nodes of G to nodes of S than induce the same set of paths. For simplicity, when referring to lineage mappings we use the notation $M(\ell)$ in lieu of $M(p_\ell)$.

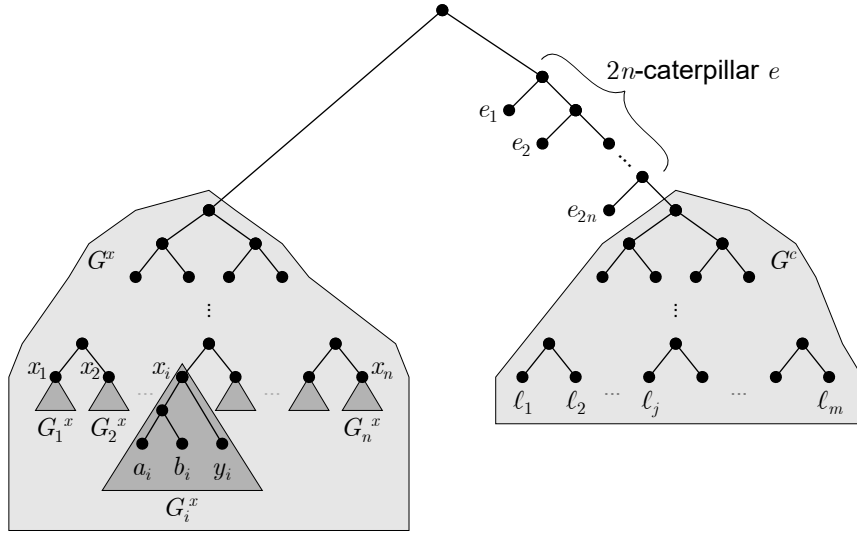
Finally, we use the notation (A, B) to represent a binary tree with a root and children A and B , either of which may be leaves or trees themselves.

Proof. Our proof is by a reduction from 3SAT. In particular, we consider the most general version of 3SAT in which the literals in a clause need not be unique and clauses need not be unique. Consider an instance of 3SAT with n variables and m clauses and, without loss of generality, assume that n and m are both powers of 2. (If not, the 3SAT instance can be padded with dummy variables and clauses to construct an equivalent instance in polynomial time by simply introducing new variables as needed and repeating clauses as needed.)

Construction. The gene tree G is constructed as follows: A *gene variable gadget* for a variable x_i comprises a tree G_i^x with three leaves, labelled a_i, b_i, y_i , with the topology $((a_i, b_i), y_i)$. The root of this gadget is labelled x_i . These n variable gadgets are connected via a perfect binary tree G^x . A *gene clause gadget* for a clause c_j consists simply of a single leaf ℓ_j . These m leaves are connected via a perfect binary tree G^c . A *k-caterpillar* of length k is a binary tree constructed from a path of length k ($k + 1$ vertices and k edges) where each of the first k vertices on that path has two children: one is the next vertex on the path, and another is a leaf. In total a k -caterpillar has $k + 1$ leaves. The root of the gene tree G has two children: one is the root of the tree G^x and the other is the root of a $2n$ -caterpillar called e . One of the two deepest leaves of caterpillar e is the root of G^c and the remaining leaves are labeled e_1, \dots, e_{2n} in order of depth from the root. The structure of the gene tree is depicted in Figure 2.

The species network S is constructed as follows: A *species variable gadget* for variable x_i consists of a subtree S_i^x with four children labeled T_i, A_i, B_i , and F_i , with topology $((T_i, A_i), (F_i, B_i))$. A_i and B_i are leaves. The root of this gadget is labelled X_i . T_i and F_i are the first vertices of paths which correspond to setting x_i to true or false, respectively. We henceforth refer to these paths as the *variable setting paths* for T_i and F_i , respectively. The remaining vertices on these variable settings paths are described in the next paragraph; ultimately these paths join at a hybridization node which has a single leaf child Y_i . The roots of these n variable gadgets are joined via a perfect binary tree S^x .

A *species clause gadget* S_j^c for clause C_j is constructed as follows. Let z_1, z_2 , and z_3 denote the three literals in that clause. If literal z_1 is the unnegated variable x_i then a vertex $U_{1,j}$ and its child $V_{1,j}$ are introduced on the variable setting path for F_i in the species variable gadget for x_i . Conversely, if literal z_1 is the negation of x_i , then vertex $U_{1,j}$ and its child $V_{1,j}$ are introduced on the variable setting path for T_i . The analogous process is



■ **Figure 2** The gene tree in the NP-hardness construction. The shaded subtree G^x on the left contains a variable gadget G_i^x for each variable x_i in the 3SAT instance, and the shaded subtree G^c on the right contains a single leaf ℓ_j for each clause j in the 3SAT instance.

used to introduce a pair of vertices $U_{2,j}$ and $V_{2,j}$ for the variable setting path for literal z_2 and a pair of vertices $U_{3,j}$ and $V_{3,j}$ for the variable setting path for literal z_3 . If the clause contains repeated literals, we will add the vertices to paths in numerical order based on their subscripts. For example, if $z_1 = z_2$, then we will introduce the vertex $U_{2,j}$ immediately after the vertex $V_{1,j}$ on the corresponding variable setting path, so that $V_{1,j}$ will be one of the parents of $U_{2,j}$. The root of the species clause gadget for clause C_j is a vertex U_j with $U_{1,j}$ as one child and U_j' as the second child whose children are $U_{2,j}$ and $U_{3,j}$. Note that $U_{1,j}$, $U_{2,j}$, and $U_{3,j}$ are hybridization nodes since they each have two parents. Node $V_{1,j}$ has a second child V_j , $V_{2,j}$ and $V_{3,j}$ have a child V_j' (a hybridization node), V_j' is another parent of V_j (a hybridization node) which, in turn, has a single child L_j , a leaf of the species network. The roots of the m species clause gadgets are connected with a perfect binary tree S^c .

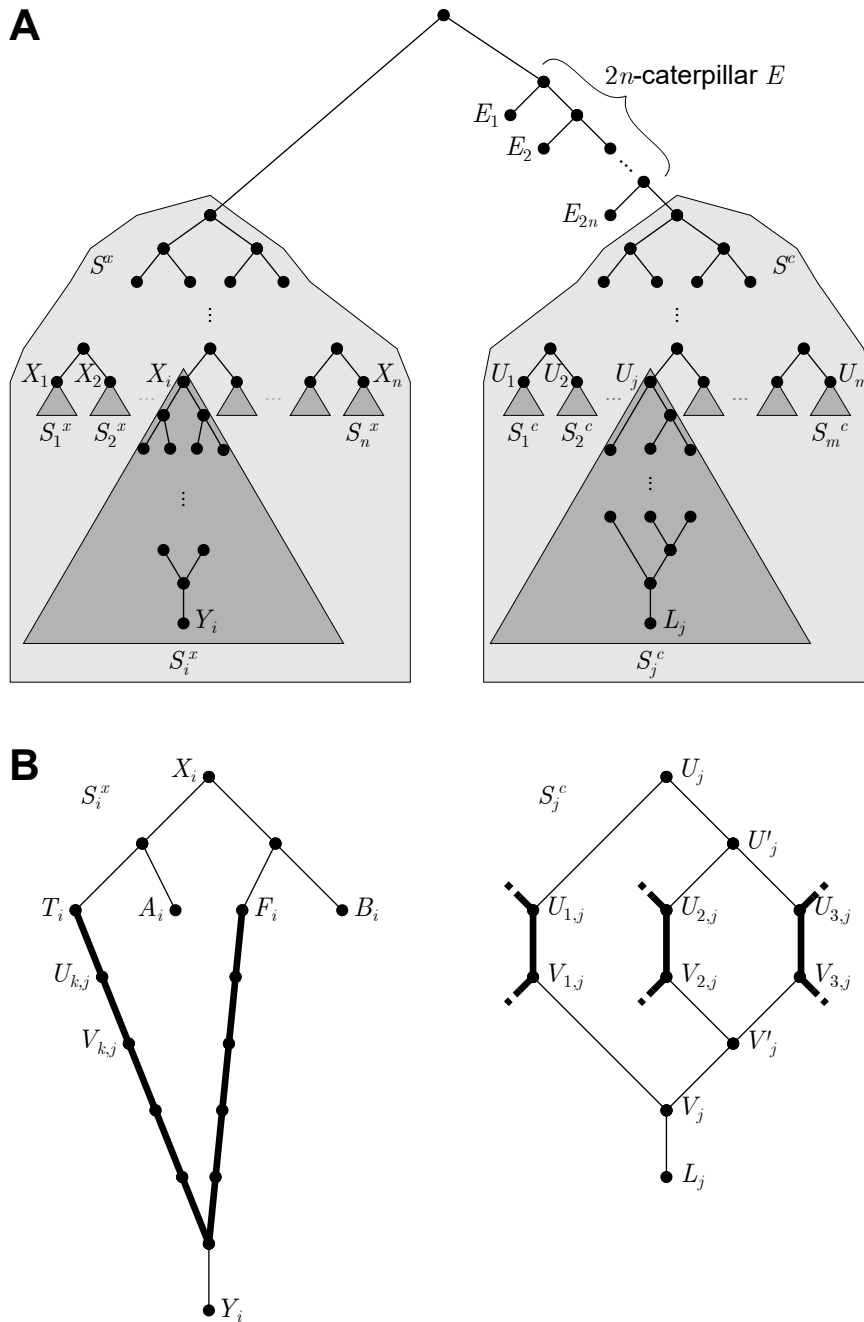
Finally, the root of the species network has two children: one is the root of the species variable tree S^x and the other is the root of a $2n$ -caterpillar called E . One of the two deepest leaves of caterpillar E is the root of the clause gadget tree S^c and the remaining leaves are labeled E_1, \dots, E_{2n} in increasing depth from the root. A representation of the species network is shown in Figure 3.

The leaf mapping Le is as follows: For the leaves in the variable gadgets, $Le(a_i) = A_i$, $Le(b_i) = B_i$, and $Le(y_i) = Y_i$, for $1 \leq i \leq n$. For the leaves of the $2n$ -caterpillar, $Le(e_i) = E_i$, for $1 \leq i \leq 2n$. For the leaves in the clause gadgets, $Le(\ell_i) = L_i$, for $1 \leq i \leq m$.

Finally, the value of k in the decision problem is set to be n , the number of variables in the 3SAT instance. It is easily seen that this construction can be performed in time polynomial in the size of the 3SAT instance.

Correctness. We prove that the constructed MPRD instance has a reconciliation with deep coalescence cost no more than n if and only if there is a satisfying assignment of the variables in the given 3SAT instance.

We begin with several observations. First, in any reconciliation, $r(G)$ must be mapped to $r(S)$ because the lowest common ancestor of the leaves in the variable gadgets and clause gadgets in G is $r(G)$ while the lowest common ancestor in S of their images under the



■ **Figure 3** The species network in the NP-hardness reduction. (A) The shaded subtree S^x on the left contains a variable gadget S_i^x for each variable x_i in the 3SAT instance, and the subtree S^c on the right contains a clause gadget S_j^c for each clause j in the 3SAT instance. (B) The variable gadget on the left is shown in detail. Vertices T_i and F_i are the first vertices on the variable setting paths (indicated in bold) for variable x_i . The clause gadget on the right is shown in detail for clause j . The bold edges are from variable setting paths for the three variables in clause j . Note that the edge $U_{k,j}, V_{k,j}$ indicated on the left is the k^{th} bold edge in the clause gadget for clause j if and only if variable x_i is the k^{th} variable in clause j . In this example, $U_{k,j}, V_{k,j}$ appears on a true variable setting path, indicating that variable x_i appears negated in clause j .

leaf mapping is $r(S)$. The species network S has unique paths from its root to the leaves A_1, \dots, A_n , B_1, \dots, B_n , and E_1, \dots, E_{2n} . Therefore, all lineage mappings have the same unique paths $M(\ell)$ for all leaves ℓ among a_1, \dots, a_n , b_1, \dots, b_n , and e_1, \dots, e_{2n} . The only leaves $\ell \in L(G)$ for which $M(\ell)$ has more than one possible path are y_1, \dots, y_n and ℓ_1, \dots, ℓ_m .

Note that in order for a reconciliation to have cost no more than n , the induced lineage mapping M must satisfy the property that $M(y_i)$ contains the node X_i , the root of the gadget S_i^x . To see this, suppose by way of contradiction that there is a variable leaf y_i such that the path $M(y_i)$ (a path from $r(S)$ to Y_i) does not contain the vertex X_i . The only paths from $r(S)$ to Y_i that do not contain X_i are through clause gadgets, and therefore $M(y_i)$ must pass through the E caterpillar. Since there is a unique path from $r(S)$ to A_i in S and that path does not pass through the E caterpillar, $M(a_i)$ cannot contain nodes from the E caterpillar. Therefore, $M(y_i)$ must diverge from $M(a_i)$ at $r(S)$, and therefore also diverges from $M(e_{2n})$ at $r(S)$ since e_{2n} is more distantly related to y_i than a_i is to y_i . But then each of the $2n$ internal nodes of the caterpillar E has at least two lineages, one from $M(y_i)$ and one from $M(e_{2n})$, contributing a cost of at least $2n > n$, contradicting the assumed cost bound.

There are, therefore, only two possibilities for $M(y_i)$ – it either includes the variable setting path for T_i or the variable setting path for F_i in the variable gadget for x_i . Both of these options contribute at least one to the total cost since $M(a_i)$ and $M(b_i)$ must diverge at (or above) X_i and, since y_i is more distantly related to a_i and b_i than a_i and b_i are to one another, $M(y_i)$ must diverge from $M(a_i)$ and $M(b_i)$ at (or above) X_i . Thus, $M(y_i)$ must contribute an extra lineage on an edge shared with $M(a_i)$ or $M(b_i)$. Since there are n variables, this contributes a cost of n , so these are necessarily the only extra lineages.

For any clause j , $M(\ell_j)$ must contain at least one of the edges $(U_{k,j}, V_{k,j})$ for $k \in \{1, 2, 3\}$. This is a consequence of the fact that without these three edges, there is no path in S from the root to L_j . Therefore $M(\ell_j)$ shares an edge with at least one species variable setting path corresponding to the negation of a literal in clause j .

Now suppose there is a satisfying assignment of the variables in the 3SAT instance. Then construct a lineage mapping M with respect to Le as follows: $M(y_i)$ contains the variable setting path T_i if the variable x_i is set to true, and the variable setting path F_i if the variable x_i is set to false. For a clause j , let z_k , $k \in \{1, 2, 3\}$, denote one of three literals in that clause that evaluates to true with respect to the given satisfying assignment. If z_k is an unnegated variable x_i , then, by construction, the F_i variable setting path contains the edge $(U_{k,j}, V_{k,j})$ in the clause gadget S_j^c . We then construct $M(\ell_j)$ so that it follows the unique path from $r(S)$ to U_j , and then passes through the clause gadget via that edge. If z_k is a negated variable $\neg x_i$, then, by construction, the T_i variable setting path contains the edge $(U_{k,j}, V_{k,j})$ in the clause gadget S_j^c and $M(\ell_j)$ is chosen to pass through the clause gadget via that edge. A reconciliation inducing this lineage mapping is trivial since each vertex in the gene tree has a corresponding vertex in the species network. The only cost incurred by this reconciliation is one for each variable gadget as noted above. The total cost is therefore n and thus this is a “yes” instance of MPRD.

Conversely, suppose there is some reconciliation with cost at most n and let M be the corresponding lineage mapping. Then, we induce a setting of each variable x_i based on whether $M(y_i)$ contains the T_i or F_i variable setting path. As noted previously, this induces a cost of n and thus the remaining paths cannot contribute any additional cost. Therefore, for each clause C_j , $M(\ell_j)$ must pass through an otherwise unused edge $(U_{k,j}, V_{k,j})$, $k \in \{1, 2, 3\}$, implying that, by construction, the k^{th} literal in clause C_j has a setting that satisfies that clause. Therefore, the 3SAT instance is satisfied. ◀

Finally, we note that for simplicity, the reduction above did not seek to ensure that the species network has a temporal representation, meaning that there is a consistent timing of events in that network. It is always possible to add additional nodes to the species network to satisfy the temporal representation property [1] and it is easily verified that our reduction holds after adding these nodes.

4 Discussion

In this work, we have shown that the problem of inferring an MPR between a gene tree and species network in the presence of incomplete lineage sorting is NP-hard. These results suggest several important directions for future research. First, approximation algorithms and exact fixed-parameter tractable algorithms should be explored for the MPR problem. Second, the problem may be solved effectively in many instances using satisfiability solvers or integer linear programming, as has been done for phylogenetic reconciliation in other event models [3, 10, 21]. Third, heuristics can be explored and tested experimentally.

References

- 1 Mihaela Baroni, Charles Semple, and Mike Steel. Hybrids in real time. *Syst Biol*, 55(1):46–56, 2006. doi:10.1080/10635150500431197.
- 2 Daniel Bork, Ricson Cheng, Jincheng Wang, Jean Sung, and Ran Libeskind-Hadas. On the computational complexity of the maximum parsimony reconciliation problem in the duplication-loss-coalescence model. *Algorithm Mol Biol*, 12(6), 2017. doi:10.1186/s13015-017-0098-8.
- 3 Morgan Carothers, Joseph Gardi, Gianluca Gross, Tatsuki Kuze, Nuo Liu, Fiona Plunkett, Julia Qian, and Yi-Chieh Wu. An integer linear programming solution for the most parsimonious reconciliation problem under the duplication-loss-coalescence model. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3388440.3412474.
- 4 Yao-ban Chan, Vincent Ranwez, and Céline Scornavacca. Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations. *J Theor Biol*, 432:1–13, 2017. doi:10.1016/j.jtbi.2017.08.008.
- 5 R. A. Leo Elworth, Huw A. Ogilvie, Jiafan Zhu, and Luay Nakhleh. Advances in computational methods for phylogenetic networks in the presence of hybridization. In Tandy Warnow, editor, *Bioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret*, pages 317–360. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-10837-3_13.
- 6 Ryan A. Folk, Pamela S. Soltis, Douglas E. Soltis, and Robert Guralnick. New prospects in the detection and comparative analysis of hybridization in the tree of life. *Am J Bot*, 105(3):364–375, 2018. doi:10.1002/ajb2.1018.
- 7 Morris Goodman, John Czelusniak, G. William Moore, A.E. Romero-Herrera, and Genji Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*, 28(2):132–163, 1979. doi:10.1093/sysbio/28.2.132.
- 8 Paweł Górecki and Jerzy Tiuryn. Dls-trees: A model of evolutionary scenarios. *Theoret Comput Sci*, 359(1–3):378–399, 2006. doi:10.1016/j.tcs.2006.05.019.
- 9 L. Li and M. S. Bansal. An integrated reconciliation framework for domain, gene, and species level evolution. *IEEE/ACM Trans Comput Biol Bioinform*, 16(1):63–76, 2019. doi:10.1109/TCBB.2018.2846253.
- 10 Lei Li and Mukul S. Bansal. An integer linear programming solution for the domain-gene-species reconciliation problem. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, page 386–397, Washington, DC, USA, 2018. Association for Computing Machinery. doi:10.1145/3233547.3233603.

- 11 R Libeskind-Hadas and M Charleston. On the computational complexity of the reticulate cophylogeny reconstruction problem. *J Comput Biol*, 16:105–117, 2009. doi:10.1089/cmb.2008.0084.
- 12 Wayne P. Maddison. Gene trees in species trees. *Syst Biol*, 46(3):523–536, 1997. doi:10.1093/sysbio/46.3.523.
- 13 Y. Ovadia, D. Fielder, C. Conow, and R. Libeskind-Hadas. The cophylogeny reconstruction problem is NP-complete. *J Comput Biol*, 18(1):59–65, 2011. doi:10.1089/cmb.2009.0240.
- 14 Roderic D.M. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst Biol*, 43(1):58–77, 1994. doi:10.1093/sysbio/43.1.58.
- 15 P. Pamilo and M. Nei. Relationships between gene trees and species trees. *Mol Biol Evol*, 5(5):568–583, September 1988. doi:10.1093/oxfordjournals.molbev.a040517.
- 16 Roswitha Schmickl, Sarah Marburger, Sian Bray, and Levi Yant. Hybrids and horizontal transfer: introgression allows adaptive allele discovery. *J Exp Bot*, 68(20):5453–5470, 2017. doi:10.1093/jxb/erx297.
- 17 Maureen Stolzer, Han Lai, Minli Xu, Deepa Sathaye, Benjamin Vernot, and Dannie Durand. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, 28(18):409–415, 2012. doi:10.1093/bioinformatics/bts386.
- 18 Fumio Tajima. Evolutionary relationship of DNA sequences in finite populations. *Genetics*, 105(2):437–460, 1983. doi:10.1093/genetics/105.2.437.
- 19 Thu-Hien To and Celine Scornavacca. Efficient algorithms for reconciling gene trees and species networks via duplication and loss events. *BMC Genomics*, 16(10):S6, October 2015. doi:10.1186/1471-2164-16-S10-S6.
- 20 Ali Tofgh, Michael Hallett, and Jens Lagergren. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans Comput Biol Bioinform*, 8(2):517–535, March 2011. doi:10.1109/TCBB.2010.14.
- 21 Nicolas Wieseke, Tom Hartmann, Matthias Bernt, and Martin Middendorf. Cophylogenetic reconciliation with ILP. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 12(6):1227–1235, November 2015. doi:10.1109/TCBB.2015.2430336.
- 22 Taoyang Wu and Louxin Zhang. Structural properties of the reconciliation space and their applications in enumerating nearly-optimal reconciliations between a gene tree and a species tree. *BMC Bioinf*, 12(Suppl 9):S7–, 2011. doi:10.1186/1471-2105-12-S9-S7.
- 23 Yun Yu, R. Matthew Barnett, and Luay Nakhleh. Parsimonious inference of hybridization in the presence of incomplete lineage sorting. *Syst Biol*, 62(5):738–751, 2013. doi:10.1093/sysbio/syt037.
- 24 Yun Yu, Nikola Ristic, and Luay Nakhleh. Fast algorithms and heuristics for phylogenomics under ILS and hybridization. *BMC Bioinformatics*, 14(15):S6, October 2013. doi:10.1186/1471-2105-14-S15-S6.
- 25 Yun Yu, Cuong Than, James H. Degnan, and Luay Nakhleh. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. *Syst Biol*, 60(2):138–149, 2011. doi:10.1093/sysbio/syq084.
- 26 Christian M. Zmasek and Sean R. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–828, September 2001. doi:10.1093/bioinformatics/17.9.821.