Integer Linear Programming Formulation for the Unified Duplication-Loss-Coalescence Model

Javad Ansarifar[1], Alexey Markin[1], Paweł Górecki[2], and Oliver Eulenstein[1]

[1] Department of Computer Science, Iowa State University, USA
javad@iastate.edu|amarkin@iastate.edu|oeulenst@iastate.edu
[2] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
gorecki@mimuw.edu.pl

Abstract. The classical Duplication-Loss-Coalescence parsimony model (DLC-model) is a powerful tool when studying the complex evolutionary scenarios of simultaneous duplication-loss and deep coalescence events in evolutionary histories of gene families. However, inferring such scenarios is an intrinsically difficult problem and, therefore, prohibitive for larger gene families typically occurring in practice. To overcome this stringent limitation, we make the first step by describing a non-trivial and flexible Integer Linear Programming (ILP) formulation for inferring DLC evolutionary scenarios. To make the DLC-model more practical, we then introduce two sensibly constrained versions of the model and describe two respectively modified versions of our ILP formulation reflecting these constraints. Using a simulation study, we showcase that our constrained ILP formulation computes evolutionary scenarios that are substantially larger than the scenarios computable under our original ILP formulation and DLCPar. Further, scenarios computed under our constrained DLC-model are overall remarkably accurate when compared to corresponding scenarios under the original DLC-model.

Keywords: Phylogenetics, Duplications, Losses, Coalescence, Reconciliation

1 Introduction

Reconstructing evolutionary histories of gene families, or gene trees, is of central importance for the understanding of gene and protein function. Gene trees make comparative and investigative studies possible that illuminate relationships between the structure and function among orthologous groups of genes, and are an indispensable tool for assessing the functional diversity and specificness of biological interlinkage for genes within the same family [1, 9, 11, 15, 16].

Crucial for understanding evolutionary histories of gene families (gene trees) is contemplating them against a respective species phylogeny; i.e., the evolutionary history of species that host(ed) the genes under consideration. This approach is known as *gene tree reconciliation*, and it can directly reveal the most valuable points of interest, such as (i) *gene duplication* events, (ii) *gene loss* events, and

(iii) *deep coalescence* or *incomplete lineage sorting* events (which appear as a result of a genetic polymorphism surviving speciation).

Traditional tree reconciliation approaches, while computationally efficient, are rather limited in practice, as they either only account for duplication and loss events, or, on the other hand, only for deep coalescence events [7, 12, 19]. Beyond the traditional approaches, recently, a robust unified *duplication-loss-coalescence (DLC)* approach has been developed that simultaneously accounts for duplications, losses, and deep coalescence events. In particular, Rasmussen and Kellis [17] originally developed a rigorous statistical model referred to as *DLCoal*. Then a computationally more feasible parsimony framework, which we refer to here as *DLC-model* was developed by Wu et al. [20]. That is, DLC-model is a discrete version of the DLCoal model, and it was shown to be very effective in practice in terms of identification of ortholog/paralog relations and accurate inference of the duplication and loss events. Wu et al. additionally presented an optimized strategy for enumerating possible reconciliation scenarios and a dynamic programming solution to find the optimum reconciliation cost; this algorithm is known as *DLCPar*.

While it has been demonstrated that DLC-model is computationally more feasible when compared to DLCoal, the exact DLCPar algorithm is still only applicable to reconciliation problems involving less than 200 genes. Limiting evolutionary studies to such a small number of genes is highly restrictive in practice, where frequently gene families with thousands of genes and hundreds of host species appear [10]. Further, the DLCPar algorithm is not scalable due to its exponential runtime [3]. Naturally, there is a demand for novel models that are (i) efficiently computable and (ii) comparable to DLCPar in terms of its accuracy.

In this work, we present a non-trivial and flexible *integer linear programming (ILP)* formulation of the DLC-model optimization problem. Then we formulate two novel and constrained DLC-models, and use our ILP formulation to validate these constrained models. That is, our models have smaller solution space and, therefore, are more efficiently computable than the original DLC-model. The validation is performed via a comprehensive simulation study with realistic parameters derived from a real-world dataset. The simulations demonstrate that both our models are applicable to larger datasets than DLCPar. Moreover, one of the models, despite the constraints, almost always provides the same reconciliation cost as the unconstrained algorithms.

Related work. In recent years, there has been an increased interest in phylogenetic methods involving simultaneous modeling of duplication, loss, and deep coalescence events [6, 18]. For example, recently, an approach for *co-estimation* of the gene trees and the respective species tree based on the DLCoal model was presented [5]. Further, Chen et al. [2] presented a parsimony framework for the reconciliation of a gene tree with a species tree by simultaneously modeling DLC events as well as horizontal gene transfer events. While promising, their approach remains computationally challenging.

Note that to the best of our knowledge, no models were proposed that would be more efficiently computable than DLC-model but be comparable with it in terms of effectiveness.

Our contribution. We developed a flexible ILP formulation that solves the DLCPar optimization problem. During the development of this formulation, we observed formal issues with the original definition of the DLC-model in [20]. Consequently, in this work, we also present corrected and improved model definitions, which are equivalent to the Wu et al. model. For example, we corrected problems with the definition of a partial order on gene tree nodes, which could otherwise lead to incorrect scoring of deep coalescence events (see Section 2 for the full updated model definitions).

Further, the ILP formulation enabled us to test the viability of a *constrained DLC-model*, which we present in this work. In particular, we observed that the advanced time complexity of DLCPar originates from allowing the duplications to appear at any edge of the gene tree, even if there is no direct "evidence" for such occurrences. While this flexibility allows accounting for all feasible DLC scenarios, we show that constraining the duplication locations to those with direct evidence of duplications will enable one to dramatically improve the efficiency of computing optimum reconciliations (without losing the accuracy).

To study the performance of the ILP formulation and test our constrained models, we designed a coherent simulation study with parameters derived from the 16 fungi dataset [17], which became a standard for multi-locus simulations [4, 14, 20]. We compared the runtimes of the unconstrained ILP (DLCPar-ILP), the constrained ILPs, and the DLCPar algorithm by Wu et al. While we observed that DLCPar was generally faster than DLCPar-ILP there were multiple instances where DLCPar-ILP was able to compute optimum reconciliations, whereas DLCPar failed. Out of 30 instances, when DLCPar failed, DLCPar-ILP was able to provide an optimum in 17 cases. Therefore, we suggest using those two methods as complements of each other. Further, an advantage of using ILPs, is that one can terminate an ILP solver early, but still get a good approximation of the optimum reconciliation cost due to the intricate optimization algorithms used by ILP solvers.

Finally, the constrained ILP models proved to be efficient even on larger datasets with more than 200 genes, where DLCPar and DLCPar-ILP failed. Moreover, we observed that one of our constrained models was accurate in 98.17% of instances.

2 Model Formulation

We use definitions and terminology similar to [20], but modify them for improved clarity and correctness.

A (phylogenetic) tree $T = (V(T), E(T))$ is a rooted binary tree, where $V(T)$ and $E(T)$ denote the set of nodes and the set of directed edges (u, v), respectively. Leaves of a phylogenetic tree are labeled by species names. By $L(T)$ we denote the set of leaves (labels) and by $I(T)$ the set of internal nodes of T, i.e., $V(T) \setminus L(T)$.

Let $r(T)$ denote the root node. By $\dot{V}(T)$ we denote the set $V(T) \setminus \{r(T)\}$. For a node v, $c(v)$ is the set of children of v (note that $c(v)$ is empty if v is a leaf), $p(v)$ is the parent of v, and $e(v)$ denotes the branch $(p(v), v)$. Let $T(v)$ be the (maximal) subtree of T rooted at v. Further, by $\mathsf{clu}(v)$ we denote the species labels below v.

Let $\leq_T$ be the partial order on $V(T)$, such that $u \leq_T v$ if and only if u is on the path between $r(T)$ and v, inclusively. For a non-empty set of nodes $b \subseteq V(T)$, let $lca_T(b)$ be the least common ancestor of b in T.

A species tree S represents the relationships among a group of species, while *a gene tree* G depicts the evolutionary history of a set of genes samples from these species. To represent the correspondence between these biological entities, we define a leaf mapping $\mathsf{Le}\colon L(G) \to L(S)$ that labels each leaf of a gene tree with the species, i.e., a leaf from S, from which the gene was sampled. The LCA mapping, $\mathcal{M}$, from gene tree nodes to species tree notes is defined as follows: if g is a leaf node, then $\mathcal{M}(g) := \mathsf{Le}(g)$; if g has two children g' and g'' then $\mathcal{M}(g) := \mathsf{lca}(\mathcal{M}(g'), \mathcal{M}(g''))$.

Definition 2.1. (DLC scenario) Given a gene tree G, a species tree S, and a leaf mapping $Le\colon L(G) \to L(S)$, the *DLC (reconciliation) scenario* for G, S, and Le is a tuple $\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$, such that

- $\mathcal{M}\colon V(G) \to V(S)$ denotes a **species map** that maps each node of gene tree to a species node. In this work, species maps are fixed to the LCA mapping.
- $\mathbb{L}$ denotes the **locus set**.
- $\mathcal{L}\colon V(G) \to \mathbb{L}$ is a *surjective* locus map that maps each node of gene tree to a locus,
- For a species node s, let parent_loci(s) be the set of loci that yield a new locus in s defined as $\{\mathcal{L}(p(g))\colon g \in \dot{V}(G),\ \mathcal{M}(g) = s \text{ and } \mathcal{L}(g) \neq \mathcal{L}(p(g))\}$. Then, $\mathcal{O}$ is a **partial order** on $V(G)$, such that, for every s and every $l \in$ parent_loci(s), $\mathcal{O}$ is a total order on the set of nodes $O(s, l) := \{g\colon g \in \dot{V}(G),\ \mathcal{M}(g) = s \text{ and } \mathcal{L}(p(g)) = l\}$.

Subject to the constraints.

1. For every locus l, the subgraph of the gene tree induced by $\mathcal{L}^{-1}(\{l\})$ is a tree. Moreover, every leaf of such a tree that is also a leaf in G must be uniquely labeled by species.
2. For every $s \in V(S)$, $l \in$ parent_loci(s), $g, g' \in O(s, l)$ if $g \leq_G g'$, then $g \leq_{\mathcal{O}} g'$.
3. A node g is called *bottom* if no child of g maps to $\mathcal{M}(g)$. We say that a node g is *top* (in $\mathcal{M}(g)$) if g is bottom in $\mathcal{M}(p(g))$. Then, $x >_{\mathcal{O}} y >_{\mathcal{O}} z$ for every bottom node $x \in O(s, l)$, every non-bottom node $y \in O(s, l)$, and every top node z in s.

The first constraint assures that all gene nodes with the same locus form a connected component; i.e., each locus is created only once. The second constraint incorporates the gene tree's topology in partial order $\mathcal{O}$. Finally, the third constraint guarantees that bottom and top nodes are properly ordered by $\mathcal{O}$.

Inserting Implied Speciation Nodes. For proper embedding a gene tree into a species tree, we require additional degree-two nodes inserted into the gene tree.

Given a gene tree, we define the transformation called *insertion of an implied speciation* as follows. The operation subdivides an edge $(g, g') \in G$ with a new node h, called an *implied speciation*, and sets $\mathcal{M}(h) = p(\mathcal{M}(g'))$ if (i) either $p(\mathcal{M}(g')) > \mathcal{M}(g)$, or (ii) $p(\mathcal{M}(g')) = \mathcal{M}(g)$ and g is not a bottom node of $\mathcal{M}(g)$. Note that h becomes a bottom node after the insertion.

Then, we transform G by a maximal sequence of implied speciation insertions. It is not difficult to see that the resulting gene tree with implied speciation nodes is well defined and unique.

Counting evolutionary events. Note that, we first define the species map $\mathcal{M}$, then we transform gene tree by inserting the implied speciation nodes. Next, we define the locus map and partial order $\mathcal{O}$ on the transformed gene tree. Finally, having the DLC scenario, we can define the evolutionary events induced by the scenario.

We start with several definitions. Let s be a node from the species tree. By $\bot(s)$ and $\top(s)$ we denote the sets of all bottom and all top nodes of s, respectively. By $nodes(s)$ we denote the set of gene nodes mapping to s (i.e., $\mathcal{M}^{-1}(\{s\})$. The *internal* nodes of s are defined as $int(s) = nodes(s) \setminus \bot(s)$.

For G, S, Le and $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$, we have the following evolutionary events at $s \in V(S)$.

- **Duplication:** A non-root gene tree node g is called a *duplication* (at $\mathcal{M}(g)$) if $\mathcal{L}(g) \neq \mathcal{L}(p(g))$. Additionally, we call g the *locus root*. We then say that a duplication happened on edge $(p(g), g)$.
- **Loss:** A locus l is *lost* at s if l is present in s or at the top of s but l is not present at the bottom of s. Formally, l is lost if $l \in \mathcal{L}(\top(s) \cup nodes(s))$ and $l \notin \mathcal{L}(\bot(s))$.
- **ILS at speciation:** Let $C(s, l)$ be the set of all gene lineages (g, g') such that g is a top node at s, whose loci is l, and g' is mapped to s. Then, locus l induces $\max\{|C(s, l)| - 1, 0\}$ (deep) coalescence events at speciation s.
- **ILS at duplication:** For each duplication d, whose parent loci is l, a gene lineage in species s at locus l is *contemporaneous* with d if the lineage starts before and ends after the duplication node d. Let $K(d)$ denote the set of all edges contemporaneous with d. Formally, $K(d) = \{g \colon g \in O(s, l)$ and $g >_{\mathcal{O}} d >_{\mathcal{O}} p(g)\}$. Then, the duplication d induces $\max\{|K(d)| - 1, 0\}$ (deep) coalescence events.

Problem 1 (DLCParsimony). Given G, S, Le, and real numbers c_D, c_L. and c_{DC}, the reconciliation cost for a DLC scenario $\alpha = (\mathcal{M}, \mathcal{L}, \mathcal{O})$ is

$$R_\alpha := \sum_{s \in V(S)} c_D \cdot nD_\alpha(s) + c_L \cdot nL_\alpha(s) + c_{DC} \cdot (nCS_\alpha(s) + nCD_\alpha(s)),$$

where $nD_\alpha(s)$, is the total number of duplication nodes at s, $nL_\alpha(s)$ is the total number of lost loci at s, and $nCS_\alpha(s)$ is the total number of coalescence events at speciation s, and $nCD_\alpha(s)$ is the total number coalescence events at duplications mapped to s in the scenario α.

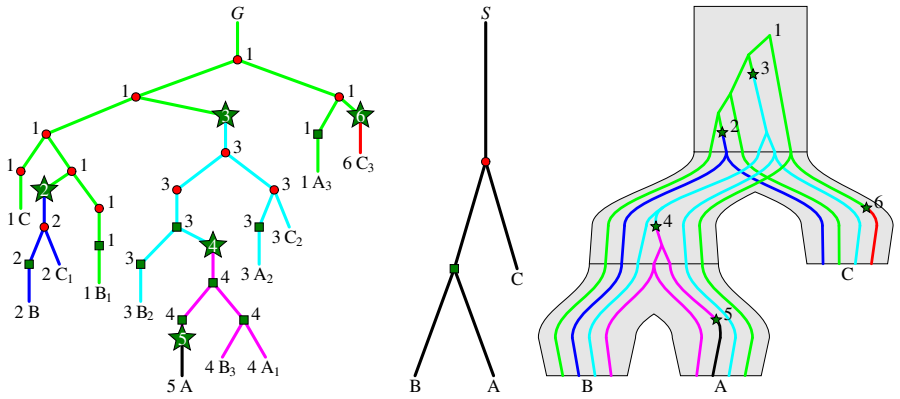

Fig. 1: An example of a DLC scenario with six loci 1 through 6. Stars indicate the duplication events. Left: a gene tree with nodes annotated by locus numbers. Middle: species tree. Right: embedding of the gene tree into the species tree.

3 ILP formulation for DLCParsimony

We now present an Integer Linear Programming (ILP) formulation for solving the DLCParsimony problem. From now on, we refer to this formulation as *DLCPar-ILP*. First, we define global parameters that can be used to constraint the formulation (see constrained models in the next section).

Model Parameters.

D_g Binary parameter for each $g \in I(G)$. It is 1, if a duplication event is *allowed* in one of the children of g. In this section $D_g = 1$ for all g, since we do not want to constrain our model.

Next we define the notation that will be used throughout the formulation.

Model Notation.

$\mathcal{I}(s)$ Possible order values (indices) of gene nodes within a total ordering of gene nodes induced by $\mathcal{O}$ and restricted to species node s. That is, $\mathcal{I}(s) = \{1, ..., |int(s)|\}$

$\mathcal{N}$ The maximum possible number of loci; i.e., maximum possible number of duplications plus one. In particular, $\mathcal{N} = 1 + \sum_{g \in I(G)} D_g$. Further, we denote the set $\{1, \ldots, \mathcal{N}\}$ by $[\mathcal{N}]$.

F_g Indicates the locus index of node g and is defined as $F_g := \sum_{g' \in I(G), g' \leq_{ord} g} D_{g'}$, where $\leq_{ord}$ is some total order on $I(G)$. F_g guarantees that duplication at node g yields a new and distinguished locus F_g in the locus tree.

Now we declare the core variables needed for the ILP formulation.

Decision Variables.

x_{uv} A binary variable for edge $(u, v) \in E(G)$. Equals to 1 if v is a duplication; otherwise 0.
y_{gl} Binary variable. 1 if node $g \in V(G)$ is assigned to locus l; otherwise 0.
e_{ls} Binary variable. 1 if locus l is lost at species node/branch s; otherwise 0.
c_{ls} The number of deep coalescence events at a speciation s induced by the locus l.
d_{gl} If g is a duplication and $l = \mathcal{L}(p(g))$, then it denotes the number of corresponding deep coalescence events induced by locus l. Otherwise, $d_{gl} = 0$.
z_{go}: Binary variable. 1, if node $g \in V(G)$ is assigned to order $o \in \mathcal{I}(\mathcal{M}(g))$.
w_{gol}: Binary variable. 1, if node $g \in V(G)$ is assigned to order o and locus l.
m_{gol}: Binary variable. 1, if node g is assigned to order o and locus l and one of children of g is a locus root (i.e., a duplication event happened immediately below g).

Finally, we describe the objective function and the model constraints using the above variables. In particular, the objective function at equation 1 minimizes the DLC score. The first term in objective function calculates the total number of duplication events, whereas the second term computes the number of loss events and coalescence events at speciations. The coalescence events at duplications are computed by the last term in the objective function.

Model constraints.

$$\min \zeta = \sum_{e \in E(G)} x_e + \sum_{s \in V(S)} \sum_{l \in [\mathcal{N}]} (e_{ls} + c_{ls}) + \sum_{s \in V(S)} \sum_{l \in [\mathcal{N}]} \sum_{g \in int(s)} d_{gl} \quad (1)$$

$$\text{s.t.} \quad \sum_{e=(g,g') \in E(G)} x_e \leq D_g \qquad g \in V(G) \quad (2)$$

$$\sum_{g \in \perp(s)} y_{gl} \leq 1 \qquad \forall s \in L(S), l \in [\mathcal{N}] \quad (3)$$

$$\sum_{l \in [\mathcal{N}]} y_{gl} = 1 \qquad \forall g \in V(G) \quad (4)$$

$$y_{r(G),1} = 1 \quad (5)$$

$$F_g x_e \leq \sum_{l \in [\mathcal{N}]} l y_{g'l} \leq F_g x_e + \mathcal{N}(1 - x_e) \qquad \forall e = (g, g') \in E(G) \quad (6)$$

$$-\mathcal{N} x_{gg'} \leq y_{g'l} - y_{gl} \leq \mathcal{N} x_{gg'} \qquad \forall (g, g') \in E(G), l \in [\mathcal{N}] \quad (7)$$

$$\sum_{g \in \top(s)} y_{gl} - |V(G)|(e_{ls} + \sum_{g \in \perp(s)} y_{gl}) \leq 0 \qquad \forall l \in [\mathcal{N}], s \in V(S) \quad (8)$$

$$\sum_{g \in \top(s), (g,g') \in E(G), g' \in nodes(s)} y_{gl} - 1 \leq c_{ls} \qquad \forall l \in [\mathcal{N}], s \in V(S) \quad (9)$$

$$\sum_{o \in \mathcal{I}(s)} z_{go} = 1 \qquad \forall s \in V(S), g \in int(s) \quad (10)$$

$$\sum_{g \in int(s)} z_{go} = 1 \qquad \forall s \in V(S), o \in \mathcal{I}(s) \quad (11)$$

$$\sum_{o' \in \mathcal{I}(s), o' \leq o} z_{g'o'} \leq 1 - z_{go} \qquad \forall s \in V(S), g, g' \in int(s), (g, g') \in E(G), o \in \mathcal{I}(s) \quad (12)$$

$$2w_{gol} \leq y_{gl} + z_{go} \leq 1 + w_{gol} \qquad \forall s \in V(S), l \in [\mathcal{N}], g \in int(s), o \in \mathcal{I}(s) \quad (13)$$

$$\sum_{g \in \top(s), (g,g') \in E(G), g' \in nodes(s)} y_{g'l} - 1 \leq n_{ls} \qquad \forall l \in [\mathcal{N}], s \in V(S) \quad (14)$$

$$n_{ls} + \sum_{g' \in int(s) \setminus \{g\}} \sum_{o' < o} (w_{g'o'l} - m_{g'o'l}) \quad \forall l \in [\mathcal{N}], s \in V(S),$$
$$\leq d_{gl} + |\top(s)|(1 - m_{gol}) \quad o \in \mathcal{I}(s), g \in int(s) \tag{15}$$
$$2m_{gol} \leq w_{gol} + \sum_{e=(g,g') \in E(G)} x_e \leq 1 + m_{gol} \quad \forall s \in V(S), l \in [\mathcal{N}],$$
$$g \in int(s), o \in \mathcal{I}(s) \tag{16}$$
$$d_{gl}, e_{ls}, c_{ls}.n_{ls} \geq 0 \tag{17}$$
$$m_{gol}, w_{gol}, x_e, y_{gl}, z_{go} \in \{0, 1\} \tag{18}$$

In a most parsimonious reconciliation scenario for each internal gene node g only one of its children can be a new locus root [20]. This condition is enforced by inequality 2. Inequality 3 enforces that extant gene nodes mapping to the same extant species must be assigned to different loci. Further, each gene node must be assigned to one locus and it is enforced by Constraint 4. Constraint 5 assigns the original locus (locus 1) to the root of the gene tree. Constraint 6 forces the child gene and its parent to map to different loci if there exists a duplication event between them. Constraint 7 guarantees that if there is no duplication event at gene edge (g, g'), then the locus of g and g' must be the same.

Constraint 8 enforces the correct calculation of loss events. In particular, it ensures that e_{ls} for locus l and species s is 1 if there exists a gene node from $\top(s)$ with locus l, while there is no gene node in $\bot(s)$ with the same locus. Constraint 9 ensures the correct assignment of c_{ls} variables (i.e., the number of coalescence events at speciations). Constraints 10 and 11 jointly assign the partial orders to interior nodes at each species branch. Based on these constraints each order must be assigned to one interior node and each interior node must be assigned to one position in the order. Constraint 12 corresponds to the constraint 2 in Definition 2.1. Constraint 13 ensures proper assignment of the w_{gol} variables. Constraints 14 and 15 should be considered together (note that n_{ls} is an additional variable that joins those two equations; it is required to properly count extra gene lineages at duplications). Those constraints together ensure proper counting of the deep coalescence events at a duplication that happens in one of the children of node g for locus l at species node s. Constraint 16 assures the correct assignment of m_{gol} variables.

3.1 Designing efficiently computable formulations

While the original DLCPar model is very flexible in terms of edges, where duplications can appear, this flexibility contributes substantially to the computational complexity of DLCPar (see the Scalability study for more details). Therefore, in this section, we consider a strategy of restraining the duplication placement only to those edges, where there is *evidence* that a duplication has occurred.

In particular, we call a node $g \in V(G)$ with children g' and g'' an *apparent duplication parent* if $\mathsf{clu}(g') \cap \mathsf{clu}(g'')$ is not empty. That is, there exist extant species, which both child lineages of g sort out to.

We then constraint the DLCPar model in the way that only children of apparent duplication parents can be locus roots. In fact, there are two options

for how this constraint can be implemented, which we call ILP-C1 and ILP-C2 that are formalized below.

ILP-C1. Observe that D_g variables defined in the previous section allow us to constrain the locations of gene duplication events easily. That is, we define the ILP-C1 formulation by properly setting the D_g variables: $D_g = 1$ if and only if g is an apparent duplication parent.

ILP-C2. Since apparent duplication parents provide strong evidence of duplications, we define, in addition, a tighter model (ILP-C2). In this model, we require that one of the children of each apparent duplication parent *must* be a duplication. Note that, while this is a strong condition, it allows us to simplify the ILP formulation and reduce the number of variables. That is, we anticipate that ILP-C2 formulation performs fastest in practice.

More precisely, in this model, we "know", where duplications must appear (at least we know the parents of duplications). Therefore, Inequality 2 in DLCPar-ILP should become an equality (which tightens the solution space); further, the m_{gol} variables become redundant, so they can be removed.

3.2 Size of ILP formulations

We analyze the size of our ILP formulations in terms of their number of variables and constraints. Let n denote the number of nodes in the gene tree and let m denote the number of nodes in the species tree. Further, let k denote the maximum possible number of loci in the gene tree. Note that $k < n$ and k in the ILP-C1 and ILP-C2 models can be expected to be significantly smaller than in the DLCPar-ILP model due to the modified D_g variables.

Then in the DLCPar-ILP and ILP-C1 models, the upper bound on the number of variables is

$$2km + (2k + 1)(n + n^2) = O(k(m + n^2)),$$

and the number of constraints is

$$(3k + 1)n^2 + (k + 2m + 3)n + 4mk + 1 = O(kn^2 + m(n + k)).$$

Finally, the ILP-C2 model has

$$2km + (2k + 1)n + (k + 1)n^2$$

variables, and

$$(k + 1)n^2 + (k^2 + 2m + 3)n + 4mk + 1$$

constraints. Observe, that the ILP-C2 model has fewer variables than the other two models (while asymptotically the same).

3.3 Searching for multiple optimal solutions

The proposed formulations can be extended to detect multiple optimal solutions through an iterative algorithm. At each iteration of that algorithm, our models identify one more alternative optimal solution (if such a solution exists). In particular, for a fixed model, at the first iteration, we solve the original model and save the optimal variables x^*, y^*, and z^* as a part of an optimal solution. To identify a different optimal solution with the same objective value, we add a new constraint such that the ILP model does not repeat identifying previously detected optimal solutions. This constraint is defined as

$$\sum_{e \in E(G)} (x_e - 1)x_e^* + \sum_{g \in V(G)} \sum_{l \in [\mathcal{N}]} (y_{gl} - 1)y_{gl}^* + \sum_{g \in V(G)} \sum_{l \in [\mathcal{N}]} (z_{go} - 1)z_{go}^* \leq -1.$$

We repeat this process as long as the optimal DLC score is the same as the previous iterations.

4 Simulation study

We present a broad simulation study that (i) compares the computational efficiency and scalability of the developed ILP models with DLCPar and (ii) validates the accuracy of the constrained ILP formulations. Note that we carry out our studies under varied simulation parameters controlling the rate of duplication/loss events as well as the rate of ILS.

Experimental setup. The process for converting an instance of the DLCParsimony problem to an ILP formulation was implemented in Python 3. Then ILP instances were solved with the Gurobi optimizer version 9.0 [8]. As for DLCPar [20], we used the exact version of the software without heuristic options for a fair comparison. Further, we set the DLCParsimony cost parameters as $c_D = c_L = c_{DC} = 1$. We performed the experiments on a standard workstation with 1.2 GHz (3.6 GHz maximum) CPU.

Simulated data. We used the standard *SimPhy* simulator [13] to generate the DLCParsimony instances. SimPhy works by first simulating a birth-death species tree and then applying the 2-step DLCoal process by Rasmussen et al. [17] to simulate the multi-locus gene trees. We use the standard simulation parameters derived from the real-world 16 fungi dataset [4, 14, 17]. In particular, we follow the parameter settings by Molloy and Warnow [14].

To conduct a comprehensive analysis and properly evaluate the proposed constrained DLCPar model, we perform our experiments under various realistic levels of the gene duplication and loss (GDL) and incomplete lineage sorting (ILS). More precisely, we use three different GDL levels: 1e-10 duplication&loss events per year (low GDL rate), 2e-10 (moderate GDL rate), and 5e-10 (high GDL rate). Further, we use two different ILS levels by controlling the tree-wide effective population size; i.e., we use the effective population sizes of 1e7 and 5e7 (that correspond to low and moderate ILS levels respectively, according to [14]).

Combination	Population size	GDL	Number of Instances			
			DLCPar-ILP	ILP-C1	ILP-C2	DLCPar
1	1e7	1e-10	2	0	0	0
2	1e7	2e-10	9	0	0	1
3	1e7	5e-10	10	1	1	8
4	5e7	1e-10	8	0	0	2
5	5e7	2e-10	9	0	0	3
6	5e7	5e-10	16	2	1	16

Table 1: Number of Instances with running time above 600 seconds out of 100 instances for each combination

Finally, we simulated DLCParsimony instances with the number of species varying from 5 to 50. That is, overall, we had $3 \times 2 \times 10 = 60$ different parameter settings for DLCParsimony instances. Then to ensure consistency, for each of the 60 parameter combinations, we generated 10 independent DLCParsimony instances. Then we executed DLCPar, DLCPar-ILP, and two constrained ILP models (referred to here as *ILP-C1* and *ILP-C2*) on each of the 600 generated problem instances. Due to a large number of instances and the advanced complexity of the models, we constrained each execution time to 10 minutes.

4.1 Results and Discussion

Run-time comparison. Table 1 shows the breakdown for each algorithm, on how many instances did it fail to complete within 10 minutes. Further, Figure 2 demonstrates the scalability ILP-C1 and ILP-C2 algorithms using examples of high-GDL and low-ILS levels. Note that we omitted DLCPar and DLCPar-ILP from the figure as there were multiple instances where those algorithms did not complete (introducing noise).

As expected, we observed that the constrained ILP formulations generally performed faster than both DLCPar and DLCPar-ILP, particularly for instances with more than 50 genes. Overall, ILP-C1 and ILP-C2 were not able to complete within 10 minutes only on 3 and 2 instances out of 600, respectively. The smallest instance size, where all algorithms failed, contained 50 species and 202 genes. Note, however, that ILP-C1 and ILP-C2 were able to complete on other instances with up to 272 genes (which was the largest number of genes in our study).

Further, we generally observed that DLCPar-ILP failed to complete on more instances than DLCPar (54/600 compared to 30/600), and DLCPar was faster than DLCPar-ILP on average. However, we observed that there were 17 instances, where DLCPar-ILP was able to complete, while DLCPar failed. At the same time, there were 38 instances, where DLCPar was able to complete, while DLCPar-ILP failed. That is, there is no clear domination of one method over the other, and the two methods can be used as complements of each other.

Validating constrained models. Given that for the vast majority of instances DLCPar-ILP or DLCPar have completed, we were able to validate the assump-

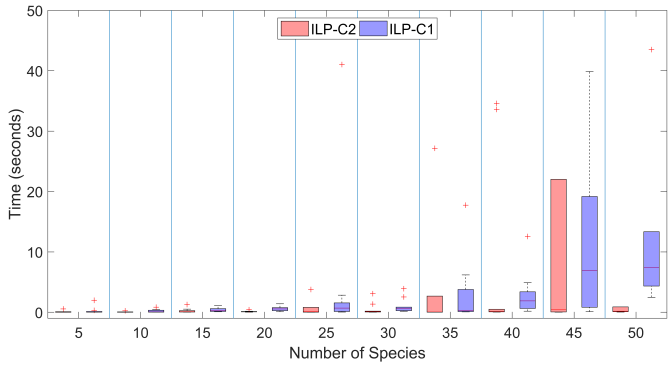

Fig. 2: Computational time comparison for ILP-C1 and ILP-C2 on the example of high-GDL and low-ILS instances.

tions of the constrained models. That is, we compare the optimum DLC reconciliation score from the constrained models against the overall optimum DLC score (in the unconstrained case). See Table 2 for the results breakdown.

Interestingly, we observed that in 98.17% of instances (where we know the optimum unconstrained cost) ILP-C1 provided exactly the same reconciliation cost as the original DLC-model. Moreover, in the 10 instances, where ILP-C1 provided a slightly higher cost, the difference in costs was *at most* 2. On the other hand, ILP-C2, which showed to be faster on average than ILP-C1, provided over-estimated reconciliation costs more often. It was exactly correct in 89.9% cases, and the difference in costs in the other 55 cases was at most 8.

That is, overall, ILP-C1 proved to be both very effective and efficient in practice, almost always providing the globally optimum reconciliation cost. Therefore, we suggest the use of this constrained model in practice.

ILP-C2 proved to be faster than ILP-C1 on average, but it gives worse accuracy due to the strength of the constraints. Indeed, ILP-C2 can be very effective in domains with low levels of ILS, since it over-estimated costs significantly less frequently when population size was smaller (see Table 2).

Combination	Population size	GDL	Number of Instances	
			ILP-C1	ILP-C2
1	1e7	1e-10	0/98	0/98
2	1e7	2e-10	1/91	1/91
3	1e7	5e-10	2/90	14/90
4	5e7	1e-10	0/92	6/92
5	5e7	2e-10	2/91	14/91
6	5e7	5e-10	5/84	20/84

Table 2: Number of Instances, where ILP-C1 and ILP-C2 score was larger than the DLCPar-ILP/DLCPar score.

Acknowledgements

We would like to thank the three anonymous reviewers for their valuable suggestions and comments, and Mukul Bansal for his support in building the foundations of the constrained DLC-model and helpful discussions. AM and OE are supported by the National Science Foundation Grant No. 1617626. PG is supported by the National Science Center grant 2017/27/B/ST6/02720.

References

1. Arvestad, L., Berglund, A.C., Lagergren, J., Sennblad, B.: Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In: Proceedings of RECOMB'04. pp. 326–335 (2004)
2. Chan, Y.b., Ranwez, V., Scornavacca, C.: Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations. Journal of theoretical biology **432**, 1–13 (2017)
3. Du, H., Ong, Y.S., Knittel, M., Mawhorter, R., Liu, N., Gross, G., Tojo, R., Libeskind-Hadas, R., Wu, Y.C.: Multiple optimal reconciliations under the duplication-loss-coalescence model. IEEE/ACM transactions on computational biology and bioinformatics (2019)
4. Du, P., Hahn, M.W., Nakhleh, L.: Species tree inference under the multispecies coalescent on data with paralogs is accurate. bioRxiv p. 498378 (2019)
5. Du, P., Nakhleh, L.: Species tree and reconciliation estimation under a duplication-loss-coalescence model. In: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. pp. 376–385 (2018)
6. Du, P., Ogilvie, H.A., Nakhleh, L.: Unifying gene duplication, loss, and coalescence on phylogenetic networks. In: Cai, Z., Skums, P., Li, M. (eds.) Bioinformatics Research and Applications. pp. 40–51. Springer, Cham (2019)
7. Górecki, P., Tiuryn, J.: DLS-trees: A model of evolutionary scenarios. Theor. Comput. Sci. **359**(1-3), 378–399 (2006)
8. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020), http://www.gurobi.com

9. Koonin, E.V.: Orthologs, paralogs, and evolutionary genomics. Annu. Rev. Genet. **39**, 309–338 (2005)
10. Li, H., Coghlan, A., Ruan, J., Coin, L.J., Heriche, J.K., Osmotherly, L., Li, R., Liu, T., Zhang, Z., Bolund, L., et al.: Treefam: a curated database of phylogenetic trees of animal gene families. Nucleic acids research **34**(suppl_1), D572–D580 (2006)
11. Lynch, M., Conery, J.S.: The evolutionary fate and consequences of duplicate genes. science **290**(5494), 1151–1155 (2000)
12. Maddison, W.P.: Gene trees in species trees. Systematic biology **46**(3), 523–536 (1997)
13. Mallo, D., de Oliveira Martins, L., Posada, D.: Simphy: phylogenomic simulation of gene, locus, and species trees. Systematic biology **65**(2), 334–344 (2015)
14. Molloy, E.K., Warnow, T.: FastMulRFS: Statistically consistent polynomial time species tree estimation under gene duplication. BioRxiv (2019)
15. Ohno, S.: Evolution by gene duplication. Springer-Verlag, Berlin (1970)
16. Page, R.D.: Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. Systematic Biology **43**(1), 58–77 (1994)
17. Rasmussen, M.D., Kellis, M.: Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome research **22**(4), 755–765 (2012)
18. Szöllősi, G.J., Tannier, E., Daubin, V., Boussau, B.: The Inference of Gene Trees with Species Trees. Systematic Biology **64**(1), e42–e62 (07 2014)
19. Wu, T., Zhang, L.: Structural properties of the reconciliation space and their applications in enumerating nearly-optimal reconciliations between a gene tree and a species tree. BMC Bioinformatics **12**(S-9), S7 (2011)
20. Wu, Y.C., Rasmussen, M.D., Bansal, M.S., Kellis, M.: Most parsimonious reconciliation in the presence of gene duplication, loss, and deep coalescence using labeled coalescent trees. Genome research **24**(3), 475–486 (2014)