Noname manuscript No.
(will be inserted by the editor)

Co-Divergence and Tree Topology

T. Calamoneri · A. Monti · B. Sinaimeri

Received: date / Accepted: date

Abstract In reconstructing the common evolutionary history of hosts and parasites, the current method of choice is the phylogenetic tree reconciliation. In this model, we are given a host tree H, a parasite tree P, and a function σ mapping the leaves of P to the leaves of H and the goal is to find, under some biologically motivated constraints, a reconciliation, that is a function from the vertices of P to the vertices of H that respects σ and allows the identification of biological events such as co-speciation, duplication and host switch.

The *maximum co-divergence* problem consists in finding the maximum number of co-speciations in a reconciliation. This problem is NP-hard for arbitrary phylogenetic trees and no approximation algorithm is known.

In this paper we consider the influence of tree topology on the maximum co-divergence problem. In particular we focus on a particular tree structure, namely caterpillar, and show that –in this case– the heuristics that are mostly used in the literature provide solutions that can be arbitrarily far from the optimal value. Then, we prove that finding the max co-divergence is equivalent to compute the maximum length of a subsequence with certain properties of a given permutation. This equivalence leads to two consequences: (i) it shows that we can compute efficiently in polynomial time the optimal time-feasible reconciliation and (ii) it can be used to understand how much the tree topology influences the value of the maximum number of co-speciations.

Research carried on while the third author was visiting the Computer Science Department of Sapienza University of Rome. This work has been partially supported by Sapienza University of Rome, project "Combinatorial structures and algorithms for problems in co-phylogeny".

T. Calamoneri, A. Monti
Computer Science Department - Sapienza University of Rome - Rome, Italy
E-mail: {calamo,monti}@di.uniroma1.it

B. Sinaimeri
INRIA Grenoble Rhône-Alpes, Université Lyon 1; CNRS, UMR5558, LBBE, Villeurbanne, France.
E-mail: blerina.sinaimeri@inria.fr

Keywords Co-phylogeny · Reconciliations · Caterpillars

1 Introduction

Symbiotic interactions concern almost every organism in the biosphere and have a great impact in essential fields of human life like health and agriculture. Due to the fact that symbiotic relationships represent a close association that may be continuous over time, the species involved may affect each other's evolution (this is known as co-evolution). Indeed, the closeness of the interaction can lead to co-speciation, that is the joint speciation of the involved species. Studying co-speciation is important for many reasons: it can shed light on the analysis of the rates of evolution between host and parasites, in determining how old the association between the host and the parasite is, and also in helping the design of better ways to combat pathogenic organisms (see for example [12]).

The task of obtaining evidence of co-speciation, presenting exciting opportunities for the study of evolution, poses many theoretical and methodological challenges. One of the principal ways to infer co-speciation is through *co-phylogeny* which allows the reconstruction of the co-evolutionary history of ecologically linked groups of organisms. Nowadays the most used method in co-phylogeny studies is phylogenetic tree reconciliation [1,3–5,9,11,6,15]. In this model, we are given a host tree H, a parasite tree P, and a function σ mapping the leaves of P to the leaves of H (σ reflects the current knowledge on which existing parasites inhabit which hosts). The goal is to find a function γ from the vertices of the parasite tree to the vertices of the host tree, that extends σ and associates to each internal vertex of the parasite tree one of the following types of biological events: (a) *co-speciation*, indicating that the parasite speciates in correspondece to a speciation of the host, (b) *duplication*, indicating that the parasite speciates independently of the host and (c) *host switch*, indicating that after a speciation of the parasite one of the new species switches to a new host that is not related to the previous one (see for example [3,4,9,11]).

A high number of co-speciations of the species involved in the symbiotic relationship is usually considered as an indicator of possible co-evolution. Thus, one notion of optimality –that was introduced by Page in [11] and is usually referred as *maximum co-divergence*– requires to maximize the number of co-speciations [11,16]. If timing information on the host tree is unknown (for real datasets this information is rarely availble and reliable) this problem is NP-hard [10,16]. The difficulty stems from the presence of host switches. Indeed, a host switch introduces a temporal constraint in the order of the speciations in the host tree. This because for a host switch to happen the donor host and the receiver host must have co-existed in time. Hence, a sequence of host switches may lead to an incompatible sequence of speciation events in the host tree. A reconciliation for which there exists an order of the speciation events in the host tree that respects the partial order induced by the topology of the tree as well as the constraints introduced by the host switches is called *time feasible*. In the following we refer to the problem of finding a time feasible reconciliation with maximum co-divergence as TF-MCD.

Considered the NP-hardness of TF-MCD, two natural relaxations are mostly considered in the literature: (i) the host switches are forbidden in the solution, and in this case the problem can be solved optimally using the LCA (the least common ancestor) mapping, and (ii) the host switches are allowed but the solutions are not required to be time feasible, in this case the problem can still be solved in polynomial time using dynamic programming (DP) but may be biologically infeasible [1, 7, 16].

For general instances, both of these relaxations provide reconciliations whose value can be arbitrary far from the value of an optimal time feasible solution. A natural question is whether for particular topologies of phylogenetic trees these algorithms can provide optimal time feasible solutions. This could potentially be used to design efficient heuristics or approximation algorithms that, by locally identifying these structures as subtrees, make optimal local choices. In this context, we consider TF-MCD in the very special case in which both the host and the parasite trees are caterpillars of the same size and function σ is a bijection.

After recalling the basic notations and definitions (Section 2) and proving some properties of time feasible reconciliations on caterpillars (Section 3), we show that even in this case the two algorithms (LCA and DP based) provide solutions that can be arbitrarily far from the optimal time feasible solution (Section 4). Then, in Section 5, we prove that finding the max co-divergence is equivalent to compute the maximum length of a subsequence with certain properties of a given permutation. This equivalence leads to two consequences: (i) it shows that we can compute efficiently in polynomial time the optimal time-feasible reconciliation and (ii) it can be used to understand how much the tree topology influences the value of the maximum number of co-speciations (that, as already pointed out, is considered as an indication of co-evolution). In Subsection 5.1. we show that, choosing the bijection σ between the n leaves of the two caterpillars uniformly at random, the value of the maximum number of co-speciations is $\Theta(\sqrt{n})$ w.h.p., so implying that –when this number is close to this value– no biological information can be deduced. In Section 6 we explore some structural properties of the set of optimal solutions for our particular instances. Many open problems arise from our work, and we list some of them in Section 7.

2 Basic notations and definitions

In this section we formalize many concepts already informally introduced in Section 1.

A *rooted phylogenetic tree* is a leaf-labelled tree that models the evolution of a set of taxa from their most recent common ancestor (placed at the root). The internal vertices of the tree correspond to the speciation events. A direction is intrinsically assumed in the tree, that corresponds to the direction of evolutionary time. Henceforth, by a phylogenetic tree T, we mean a rooted tree with labelled leaves and where every vertex has in-degree 1 (except for the root that has in-degree 0) and out-degree 2 (except for the leaves that have out-degree 0). For such a tree T, the set of vertices is denoted by $V(T)$, the set of arcs by $A(T)$, and the set of leaves by $Leaves(T)$.

For a vertex $v \in V(T)$:

- if v is different from the root, we call its *parent* as $par(v)$;
- we denote by T_v the subtree of T rooted in v and the vertices in T_v are the *descendants* of v;
- the set of *ancestors* of v, denoted by $Anc(v)$, is the set of vertices in the unique path from the root of T to v. We denote by $Anc^{-}(v) = Anc(v) \setminus \{v\}$.

Two vertices u and v are said to be *incomparable* if neither $u \in Anc(v)$ nor $v \in Anc(u)$.

The reconciliation function
Let H, P be the phylogenetic trees for the host and parasite species respectively. We define σ as a function from the leaves of P to the leaves of H that represents the association between currently living parasite and host species.

Definition 1 [16] Given H, P and σ, a function $\gamma : V(P) \to V(H)$ is a *reconciliation* if:

1. for any $p \in Leaves(P)$, $\gamma(p) = \sigma(p)$ (γ extends σ);
2. for any internal vertex $p \in V(P) \setminus Leaves(P)$ with children p_1 and p_2:
   (a) $\gamma(p_i) \notin Anc^{-}(\gamma(p))$ for $i = 1, 2$ (a child cannot be mapped to an ancestor of the father);
   (b) $\mathtt{LCA}(\gamma(p), \gamma(p_1)) = \gamma(p)$ or $\mathtt{LCA}(\gamma(p), \gamma(p_2))) = \gamma(p)$ (one of the two children is mapped to the subtree rooted at its father);

moreover, γ highlights a subset Ξ_γ of $A(P)$ and partitions the set $V(P)$ into three sets Θ_γ, Δ_γ and Σ_γ as follows:

3. given an arc $(u, v) \in A(P)$, $(u, v) \in \Xi_\gamma \Leftrightarrow \mathtt{LCA}(\gamma(u), \gamma(v)) \notin \{\gamma(u), \gamma(v)\}$ ($\gamma(u)$ and $\gamma(v)$ are incomparable and arc (u, v) is a host switch);
4. for any $p \in V(P) \setminus Leaves(P)$ with children p_1 and p_2:
   (a) $p \in \Theta_\gamma \Leftrightarrow (p, p_1) \in \Xi_\gamma$ or $(p, p_2) \in \Xi_\gamma$ (p is associated to a host switch event);
   (b) $p \in \Delta_\gamma \Leftrightarrow \mathtt{LCA}(\gamma(p_1), \gamma(p_2)) \in \{\gamma(p_1), \gamma(p_2)\}$ (the children are mapped to comparable vertices and p is associated to a duplication event);
   (c) $p \in \Sigma_\gamma \Leftrightarrow \mathtt{LCA}(\gamma(p_1), \gamma(p_2)) = \gamma(p)$ and $\gamma(p_1)$ and $\gamma(p_2)$ are incomparable (p is associated to a co-speciation event).

Time-feasibility of a reconciliation
Time feasibility is an important biological constraint that we require on the reconciliation. We recall here the definition of a time feasible reconciliation presented in [5,15].

Definition 2 Given a reconciliation γ, construct directed graph G_γ as follows:

- the vertex set of G_γ coincides with $V(H)$,
- the arc set of G_γ contains the arcs of $A(H)$ and in addition:
  - for any host switch $(u,v) \in \Xi_\gamma$ the following arcs:
    (1a) $(par(\gamma(u)), \gamma(v))$; (1b) $(par(\gamma(v)), \gamma(u))$;
  - for all the couples of host switches $(u,v),(u',v') \in \Xi_\gamma$ for which $u \in Anc(u')$ the following arcs:
    (2a) $(par(\gamma(u)), \gamma(u'))$; (2b) $(par(\gamma(u)), \gamma(v'))$;
    (2c) $(par(\gamma(v)), \gamma(u'))$; (2d) $(par(\gamma(v)), \gamma(v'))$.

γ is *time feasible* if and only if G_γ does not contain any directed cycle.

Max co-divergence problem (TF-MCD)

Given any reconciliation γ, we define its *value*, $val(\gamma)$, as the number of co-speciations it contains, *i.e.* $|\Sigma_\gamma|$. In the literature, this value is usually called as *co-divergence*. We focus on the following problem:

Definition 3 The *max co-divergence problem (TF-MCD)* is characterized by the following instance and question.
Instance: $I = (H, P, \sigma)$ containing two phylogenetic trees H, P and a function $\sigma : Leaves(P) \to Leaves(H)$;
Question: Find a time feasible reconciliation γ for which $val(\gamma)$ is maximum.

Our setting

In this paper we consider the case when *the host tree H and the parasite tree P are both caterpillars on n leaves and σ is a bijection.*

For the forthcoming definitions refer to Figure 1. We recall that a *caterpillar* is a tree in which all the vertices are within distance 1 of a central path, called *spine*. Every rooted phylogenetic tree that is also a caterpillar can be uniquely identified by the sequence of the labels of its leaves ordered from the furthest to the nearest to the root (for what concerns the only two leaves at the same distance from the root, by convention, we put first the one with lexicographic smaller label). Without loss of generality we will assume that P will be identified by the sequence $0, 1, 2, \ldots, n-1$ and H by $l_0, l_1, \ldots, l_{n-1}$, with $l_0 < l_1$.

We will assume that the (internal) vertices on the spine of P are called starting from the furthest from the root as $x_1, x_2, \ldots, x_{n-1}$. Similarly, we call y_1, …, y_{n-1} the vertices on the spine of H.

Note that x_1 has two children that are the leaves 0 and 1 and y_1 has two children that are the leaves l_0 and l_1; for any other $2 \le i \le n-1$, the children of vertex x_i are leaf i and internal vertex x_{i-1}, while the children of y_i are leaf l_i and internal vertex y_{i-1}.

Since in our setting H and P have a fixed structure and σ is a bijection, an instance $I = (H, P, \sigma)$ can be identified simply by sequence $S = \sigma^{-1}(l_0), \ldots, \sigma^{-1}(l_{n-1})$ of the leaves of the host, that is a permutation on $\{0, 1, 2, \ldots, n-1\}$. So, in the rest of this paper, when we speak about S we implicitly assume to have given the two caterpillars H and P, and bijection σ.

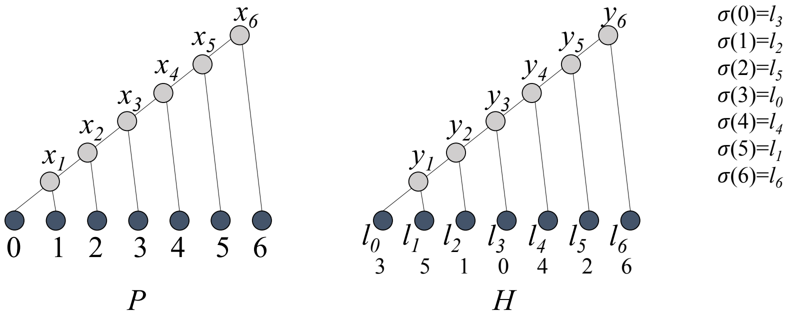

Fig. 1 An example of an instance where host and parasite trees are the caterpillars on 7 leaves together with the mapping σ. Note that $S = 3, 5, 1, 0, 4, 2, 6$.

3 Properties of time feasible reconciliations for caterpillars

Here we prove some properties of time feasible reconciliations whose instance is a pair of caterpillars connected by a bijection σ. We will use these properties in the following of the paper.

Theorem 1 *Given an instance S, let γ be one of its time feasible reconciliations; it holds that $\gamma(x_i) \notin Anc^-(\gamma(x_j))$ for any $1 \leq i < j \leq n-1$.*

Proof Suppose on the contrary that the claim does not hold, so there exist two indices $i < j$ such that $\gamma(x_i) \in Anc^-(\gamma(x_j))$ and $j - i$ is minimum. This assumption has some immediate consequences:

1. $j - i > 1$, indeed, in view of Definition 1.2(a) (a child cannot be mapped to an ancestor of its parent), x_j cannot be the parent of x_i;
2. $Anc^-(\gamma(x_i)) \subset Anc^-(\gamma(x_j))$;
3. $\gamma(x_{i+1}) \notin Anc^-(\gamma(x_j))$ and $\gamma(x_i) \notin Anc^-(\gamma(x_{j-1}))$ because $j - i$ is minimum.

From item 1. it follows that $x_{i+1} \neq x_j$ and $x_{j-1} \neq x_i$ (although x_{i+1} and x_{j-1} are not necessarily distinct). We will prove that, in view of our assumption, in γ there necessarily exist two host switches (x_{i+1}, x_i) and (x_j, x_{j-1}) making γ time infeasible, so reaching a contradiction.

Presence of host switch (x_{i+1}, x_i)**.** In view of Definition 1.2(a) we have that $\gamma(x_i) \notin Anc^-(\gamma(x_{i+1}))$. On the other hand, from items 2. and 3. we have $\gamma(x_{i+1}) \notin Anc^-(\gamma(x_j)) \supset Anc^-(\gamma(x_i))$.

It follows that x_{i+1} is mapped by γ neither in subtree of $H_{\gamma(x_i)}$, nor in an ancestor of $\gamma(x_i)$. Hence, it is mapped in some leaf outside $H_{\gamma(x_i)}$ and (x_{i+1}, x_i) is a host switch.

Presence of host switch (x_j, x_{j-1})**.** In view of item 3. $\gamma(x_i) \notin Anc^-(\gamma(x_{j-1}))$. Moreover, from Definition 1.2(a) and item 2. we have $\gamma(x_{j-1}) \notin Anc^-(\gamma(x_j)) \supset Anc^-(\gamma(x_i))$.

Thus $\gamma(x_{j-1})$ is a leaf outside $H_{\gamma(x_i)}$. Since $\gamma(x_i) \in Anc^-(\gamma(x_j))$, $\gamma(x_j) \in H_{\gamma(x_i)}$, so x_j and x_{j-1} are incomparable, implying that (x_j, x_{j-1}) is a host switch.

Since $x_j \in Anc^-(x_i)$, in graph G_γ will be added the four arcs related to the pair of host switches (x_{i+1}, x_i) and (x_j, x_{j-1}) with $u = x_j, v = x_{j-1}$ and $u' = x_{i+1}, v' = x_i$, and in particular arc (2b), that is $(par(\gamma(x_j)), \gamma(x_i))$. By hypothesis $\gamma(x_i) \in Anc^-(\gamma(x_j))$, thus in G_γ there is already a path from $\gamma(x_i)$ to $par(\gamma(x_j))$; hence the addition of arc $(par(\gamma(x_j)), \gamma(x_i))$ creates a cycle contradicting the hypothesis that γ is time feasible. □

The next corollary follows straightforwardly from Theorem 1.

Corollary 1 *Given an instance S, let γ be one of its time feasible reconciliations. For any two internal vertices of P, x_i and x_j, with $i < j$ for which $\gamma(x_i), \gamma(x_j) \notin Leaves(H)$, it holds that $\gamma(x_j) \in Anc(\gamma(x_i))$.*

In other words, for any time feasible reconciliation γ, the order of the mapping of the internal vertices is kept on the spine of the caterpillar H.

The next theorem shows a simple sufficient condition for γ to be time feasible.

Theorem 2 *Given an instance S, let γ be one of its reconciliations. Then γ is time feasible if for every two distinct internal vertices a, b of P, with $b \in Anc^-(a)$, both the following conditions hold:*

(i) If $\gamma(a) \notin Leaves(H)$ then $\gamma(b) \notin Leaves(H)$
(ii) $\gamma(a) \notin Anc^-(\gamma(b))$

Proof Let γ be a reconciliation that satisfies the conditions (i) and (ii) and suppose by contradiction that γ is not time feasible. By Definition 2, this implies that G_γ has a cycle C. Note that G_γ is formed by the directed caterpillar H and some additional arcs. By construction, in G_γ there are no arcs departing from a leaf of H, as every arc departs from $par(w)$ for some w. Thus, no leaf belongs to cycle C (where every vertex must have outdegree at least one). Hence, calling y_i the vertex in C with smallest index, there exists y_j, with $i \leq j$ such that arc $e = (y_i, y_j)$ is in C. This arc should have necessarily been added because of either : (1) a single host switch (u, v) or (2) a pair of two host switches (u, v) and (u', v') such that $u \in Anc^-(u')$ (moreover, it easily follow that $u \in Anc^-(v') \cap Anc^-(v)$). We show now that (1) and (2) are both not possible by a case by case analysis depicted in Figure 2:

(1a) If e is of type (1a) of Definition 2, then $y_i = par(\gamma(u))$ and $y_j = \gamma(v)$; hence, $\gamma(v) \in Anc^-(\gamma(u))$ (see Figure 2(1a));
(1b) if e is of type (1b) of Definition 2, then $y_i = par(\gamma(v))$ and $y_j = \gamma(u)$; it follows that $\gamma(u) \in Anc^-(\gamma(v))$ (see Figure 2(1b)).

In both cases we get a contradiction as (u, v) is a host switch and, by item (3) of Definition 1, $\gamma(u)$ and $\gamma(v)$ must be incomparable in H.

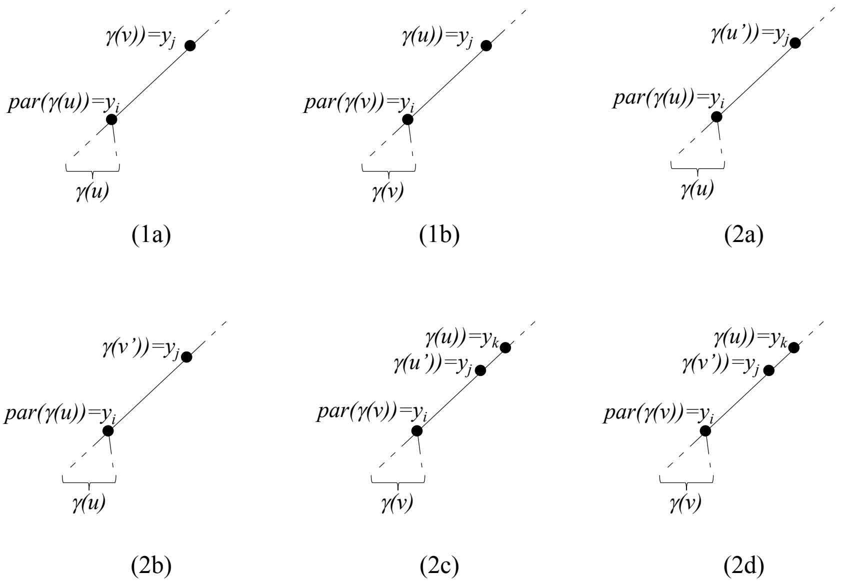

Fig. 2 The cases in the proof of Theorem 2.

(2a) If e is of type (2a) of Definition 2, then $y_i = par(\gamma(u))$ and $y_j = \gamma(u')$; hence $\gamma(u') \in Anc^-(\gamma(u))$ (see Figure 2(2a));

(2b) if e is of type (2b) of Definition 2, then $y_i = par(\gamma(u))$ and $y_j = \gamma(v')$; so $\gamma(v') \in Anc^-(\gamma(u))$ (see Figure 2(2b)).

Since we know that $u \in Anc^-(u') \cap Anc^-(v')$, in both cases we contradict condition (ii) of the statement.

(2c) If e is of type (2c) of Definition 2, $y_i = par(\gamma(v))$ and $y_j = \gamma(u')$; from condition (i) of the statement, $\gamma(u)$ is an internal vertex y_k in H; from condition (ii), $\gamma(u') \notin Anc^-(\gamma(u))$, hence $k > j$ (see Figure 2(2c));

(2d) if e is of type (2d) of Definition 2, $y_i = par(\gamma(v))$ and $y_j = \gamma(v')$; from condition (i) $\gamma(u)$ is an internal vertex y_k in H; from condition (ii), $\gamma(v) \notin Anc^-(\gamma(u))$, hence $k > j$ (see Figure 2(2d)).

In both cases it follows that $\gamma(u) \in Anc^-(\gamma(v))$, that is u and v are not incomparable and hence (u, v) cannot be a host switch, a contradiction. □

It is worth to note that Theorem 2 defines only sufficient conditions for a reconciliation to be time feasible. Indeed, while condition (ii) of Theorem 2 is necessary for the time feasibility of a reconciliation (see Theorem 1), condition (i) is not, as shown by the example in Figure 3.

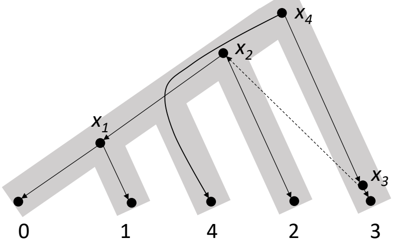

Fig. 3 An example of time feasible reconciliation γ showing that condition (i) of Theorem 2 is not necessary: here $S = 0, 1, 4, 2, 3$; $a = x_1$ and $b = x_3$, hence $b \in Anc^-(a)$, $\gamma(a) \notin Leaves(H)$ but $\gamma(b) \in Leaves(H)$.

For every reconciliation γ, here we prove that for every internal vertex x_i of P, at least one of the leaves in its subtree is mapped in a leaf of the subtree rooted at $\gamma(x_i)$ in H.

Lemma 1 *Given an instance S, let γ be one of its time feasible reconciliations; for every internal vertex x_i there exists a $j \in Leaves(P_{x_i})$ for which $\sigma(j) \in Leaves(H_{\gamma(x_i)})$.*

Proof The proof is by contradiction, so we assume that there exists $\bar{\imath}$ such that all the leaves of P $0, 1, \dots, \bar{\imath}$ are mapped outside $H_{\gamma(x_{\bar{\imath}})}$. Then let l be the smallest integer for which $\gamma(x_l)$ belongs to $H_{\gamma(x_{\bar{\imath}})}$. (Observe that such an l is at most $\bar{\imath}$ because at least $\gamma(x_{\bar{\imath}}) \in H_{\gamma(x_{\bar{\imath}})}$.)

If $l = 1$ by our assumption $\sigma(0), \sigma(1) \notin H_{\gamma(x_{\bar{\imath}})}$ which contradicts Definition 1.2(b) (stating that at least one of the two children is mapped to the subtree rooted at the father).

If $l > 1$ then $x_{l-1} \notin H_{\gamma(x_{\bar{\imath}})}$ as l is the smallest integer and again by our assumption $\sigma(l) \notin H_{\gamma(x_{\bar{\imath}})}$ which again contradicts Definition 1.2(b). □

Note that the previous result holds not only for caterpillars but for any two host and parasite trees.

We now prove that every reconciliation maps co-speciations and duplications to internal vertices of H.

Lemma 2 *Given an instance S, let γ be one of its time feasible reconciliations. For every internal vertex x_i, if $\gamma(x_i) \in Leaves(H)$ then $x_i \in \Theta_\gamma$.*

Proof Let x_i be such that $\gamma(x_i) = l_k$ and suppose on the contrary that $x_i \notin \Theta_\gamma$. Note that clearly $x_i \notin \Sigma_\gamma$, hence $x_i \in \Delta_\gamma$ and then both its children are mapped in l_k. Note that $i > 1$ as the children of x_1 cannot be both mapped in l_k (being σ a bijection). Then we should have $\gamma(x_{i-1}) = \sigma(i) = l_k$. Note that none of the leaves of $P_{x_{i-1}}$ is mapped in $Leaves(H_{\gamma(x_{i-1})}) = Leaves(H_{l_k}) = \{l_k\}$. Then from Lemma 1 we reach a contradiction. □

4 LCA and DP based algorithms on caterpillar phylogenetic trees

In this section we show that, even in the special case when the two trees are caterpillars, the LCA and DP based algorithms can produce solutions arbitrarily far from the value of an optimal time feasible solution. To this purpose, given an instance S we can define three values:

- $val(\gamma_{LCA})$, the number of co-speciations in the unique time feasible reconciliation γ_{LCA} obtained by mapping each internal vertex of P to the least common ancestor of the mapping of its children in H;
- $val(\gamma_{OPT})$, the number of co-speciations in any time feasible reconciliation γ_{OPT} that has the maximum number of co-speciations;
- $val(\gamma_{DP})$, the number of co-speciations in a possibly time infeasible reconciliation γ_{DP} that has the maximum number of co-speciations. Note that this can be obtained by a dynamic programming algorithm (see for example [1,5,16]).

For these three values the following holds:

Fact 3 *For every arbitrary instance $I = (H, P, \sigma)$:*

$$val(\gamma_{LCA}) \leq val(\gamma_{OPT}) \leq val(\gamma_{DP}).$$

Fact 3 provides upper and lower bounds for the maximum number of co-speciations in an optimal time feasible reconciliation. Even in our very simplified setting, these lower and upper bounds can be arbitrarily far as the gap may depend on the size of the trees as shown by the following examples.

Example 1. We define an instance S for which $val(\gamma_{LCA})$ is arbitrarily far from $val(\gamma_{OPT})$.

S is defined by the following bijection σ:

$$\sigma(i) = \begin{cases} l_{n-1} & i = 0 \\ l_0 & i = n-1 \\ l_i & \text{otherwise.} \end{cases}$$

Sequence S is obtained from the identity permutation by exchanging the values in the first and last position. As an example, see H_1 in Figure 4 when $n = 7$.

Observe that γ_{LCA} maps all x_i, $1 \leq i \leq n-1$, in the root of H_1. From Definition 1, $\Sigma_{\gamma_{LCA}} = \{x_1\}$ and $\Delta_{\gamma_{LCA}} = \{x_2, \dots, x_{n-1}\}$. Hence, $val(\gamma_{LCA}) = 1$.

Now, observe that γ defined as $\gamma(x_i) = y_i$ for $1 \leq i \leq n-1$ is a reconciliation with $\Sigma_\gamma = \{x_2, \dots x_{n-2}\}$, $\Theta_\gamma = \{x_1\}$ and $\Delta_\gamma = \{x_{n-1}\}$. Hence, $val(\gamma) = n-3$, and γ is a time feasible reconciliation in view of Theorem 2. Then:

$$1 = val(\gamma_{LCA}) < val(\gamma) = n-3 \leq val(\gamma_{OPT}).$$

Example 2. We define an instance S for which $val(\gamma_{DP})$ is arbitrarily far from $val(\gamma_{OPT})$.

To this purpose for n odd, let $S = 0, \frac{n+1}{2}, 1, \frac{n+1}{2}+1, 2, \frac{n+1}{2}+2, \ldots, n-1, \frac{n+1}{2}-1$ be defined by the following bijection σ:

$$\sigma(i) = \begin{cases} l_{\frac{n+i}{2}} & \text{if } i \text{ is odd} \\ l_{\frac{i}{2}} & \text{otherwise} \end{cases}$$

Sequence S is obtained by putting the first $\frac{n+1}{2}$ integers in the even positions and the remaining $\frac{n-1}{2}$ integers in the odd positions. As an example, see H_2 in Figure 4 when $n = 7$. For this example, γ_{LCA} is the following reconciliation:

$$\gamma_{LCA}(x_i) = \begin{cases} y_{2i} & \text{if } 1 \leq i \leq \frac{n-1}{2} \\ y_{\frac{n-1}{2}} & \text{otherwise} \end{cases}$$

that produces $\Sigma_{\gamma_{LCA}} = \{x_1, x_2, \ldots x_{\frac{n-1}{2}}\}$ and $\Delta_{\gamma_{LCA}} = \{x_{\frac{n+1}{2}} \ldots, x_{n-1}\}$. Hence, $val(\gamma_{LCA}) = \frac{n-1}{2}$. By applying Theorem 4 (stated in the next section) we have $val(\gamma_{LCA}) = val(\gamma_{OPT})$.

Consider now the following reconciliation γ:

$$\gamma(x_i) = \begin{cases} y_{2i} & \text{if } 1 \leq i < \frac{n-1}{2} \\ \sigma(n-1) & \text{if } i = \frac{n-1}{2} \\ y_{2i-n} & \text{otherwise} \end{cases}$$

that produces $\Sigma_\gamma = \{x_1, \ldots x_{\frac{n-1}{2}-1}, x_{\frac{n-1}{2}+1}, \ldots x_{n-1}\}$, $\Theta_\gamma = \{x_{\frac{n-1}{2}}, x_{\frac{n+1}{2}}\}$, giving $val(\gamma) = n-3$. Clearly, as the DP algorithm takes the maximum over all reconciliations (time infeasible and feasible) we conclude:

$$\frac{n-1}{2} = val(\gamma_{LCA}) = val(\gamma_{OPT}) < n-3 = val(\gamma) \leq val(\gamma_{DP}).$$

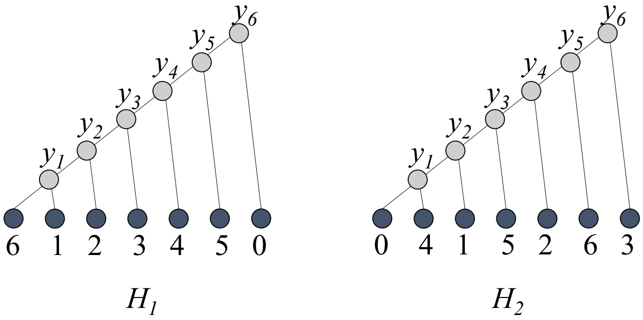

Fig. 4 The host trees H_1 and H_2 illustrating the two examples where the LCA and the DP based algorithms produce reconciliations whose values are far from the optimum.

5 TF-MCD is polynomially solvable for identical caterpillars and bijection σ

In this section we show that, although the DP and LCA based algorithms do not provide good solutions in the case of caterpillars, it is however possible to provide the optimal time feasible solution in polynomial time.

As already mentioned in the introduction, if the timing information on the host tree (*i.e.* the order in which speciation events occurred in the host phylogeny) is not available, TF-MCD is NP-hard [10,16]. If timing information on the host tree is known, the problem has been tackled in [14,7,1]. However, the optimal solutions produced by the algorithm in [1] can still be time infeasible. This is because the constraint used in [1] to ensure the time feasibility works only locally but two locally time consistent host switch events can be globally inconsistent (see [7]). In [14,7] the authors propose an algorithm of time complexity $O(nm)$ (for n and m size of the host and parasite tree, respectively) which globally guarantees the time feasibility, but their model is slightly different from the one we used (for example the duplication events are not defined the same way and host switch events are defined on edges).

Based on this, although the structure of caterpillar defines a total order of the speciation events in the host tree, the polynomiality results in [14,7,1] do not directly apply to our model. Here we show that such an algorithm can be still found. Moreover it is efficient as it has time complexity $O(n \log n)$.

Definition 4 Given a sequence $z = z_1, \ldots, z_t$, $t \geq 2$ of integers, a subsequence $z_{i_1}, z_{i_2} \ldots, z_{i_k}$, of z is *nearly increasing* if either (a) $i_k = 2$, or (b) $z_{i_2} < \ldots < z_{i_k}$ and $z_{i_1} < z_{i_3}$. We denote by $lnis(z)$ the length of a longest nearly increasing subsequence of z.

In other words, a subsequence is nearly increasing if either it is increasing or it becomes increasing by exchanging the positions of the first two elements. Observe that the possibility to re-order the first two elements of the sequence mimics the fact that there is no fixed order in the unique pair of leaf siblings of the caterpillar.

We now proceed to prove the main result of this section.

Lemma 3 *Given an instance S, let γ be one of its time feasible reconciliations; then $lnis(S) \geq val(\gamma) + 1$.*

Proof Let it be $val(\gamma) = r$. If $r = 0$ or $r = 1$ then the claim is obvious because $lnis(S) \geq 2$. So, let it be $r \geq 2$. We will show that the r co-speciations of γ will define a nearly increasing subsequence of S of length $r + 1$. Let $\Sigma_\gamma = \{x_{i_1}, \ldots, x_{i_r}\}$, with $i_1 < i_2 \ldots < i_r$. From Lemma 2 for all $1 \leq j \leq r$, $\gamma(x_{i_j}) = y_{k_j}$. From Corollary 1 we have $k_1 < k_2 \ldots < k_r$. The sequence $k_1, \ldots, k_r$ identifies a subsequence of length r of S, $S' = \sigma^{-1}(l_{k_1}), \ldots, \sigma^{-1}(l_{k_r})$ in H.

We show now that S' is an increasing subsequence, that is for all $1 \leq j < r$, $\sigma^{-1}(l_{k_j}) < \sigma^{-1}(l_{k_{j+1}})$. To this purpose is enough to show that

$$\text{for } 1 \leq j < r \text{ it holds } i_j < \sigma^{-1}(l_{k_{j+1}}) \leq i_{j+1}. \tag{1}$$

Since $x_{i_{j+1}}$ is associated to a co-speciation event, exactly one of its two children must be mapped on $l_{k_{j+1}}$. If the child mapped on $l_{k_{j+1}}$ is leaf i_{j+1} then we are done since $i_j < \sigma^{-1}(l_{k_{j+1}}) = i_{j+1}$. Otherwise the child mapped on $l_{k_{j+1}}$ is internal vertex $x_{i_{j+1}-1}$. Let t be the largest integer such that $x_{i_{j+1}-1}, \ldots, x_{i_{j+1}-t}$ are all mapped in $l_{k_{j+1}}$. This means that $\sigma^{-1}(l_{k_{j+1}}) = i_{j+1} - t$. Since $i_{j+1} - t \leq i_{j+1}$ we have that $\sigma^{-1}(l_{k_{j+1}}) \leq i_{j+1}$.

On the other hand, x_{i_j} is a co-speciation, so it is mapped on the spine of H and so it cannot be mapped in $l_{k_{j+1}}$, and thus $i_{j+1} - t > i_j$. Hence, $i_j < i_{j+1} - t = \sigma^{-1}(l_{k_{j+1}})$.

It remains to show that we can extend S' to the left to create a nearly increasing subsequence of length $r+1$. We show that exists a l_j with $0 \leq j < k_1$ such that $\sigma^{-1}(l_j) < \sigma^{-1}(l_{k_2})$.

If $i_1 = 1$ then, as x_1 is a co-speciation, both its children $0, 1$ are mapped in the subtree $H_{\gamma(x_1)}$ and we have two cases:

(a) $\sigma^{-1}(l_{k_1}) = 0$: then 1 will be mapped in some l_j with $j < k_1$ and we extend S' by adding as a first element 1 (note that $1 < \sigma^{-1}(l_{k_2})$;

(b) $\sigma^{-1}(l_{k_1}) = 1$: then 0 will be mapped in some l_j with $j < k_1$ and we extend S' by adding as a first element 0 obtaining an $(r+1)$ long increasing subsequence.

Finally, suppose that $i_1 > 1$, then there exists x_{i_1-1} and Definition 1.4(c) guarantees $\gamma(x_{i_1-1}) \in H_{\gamma(x_{i_1})}$. There are two cases:

(a) $\gamma(x_{i_1-1}) = l_{k_1}$: then, as x_{i_1} is a co-speciation, i_1 is mapped in some l_j with $j < k_1$. From inequality (1) $i_1 < \sigma^{-1}(l_{k_2})$ and so we extend to the left S' by adding i_1;

(b) $\gamma(x_{i_1-1}) \neq l_{k_1}$: then, by Lemma 1, there exists a leaf l_j with $j < k_1$ such that $\sigma^{-1}(l_j) \leq i_1 - 1$. We extend S' to the left by adding $\sigma^{-1}(l_j)$. Note that the obtained sequence is increasing as $\sigma^{-1}(l_j) < \sigma^{-1}(l_{k_1})$. □

In the following lemma we show it is possible to construct a time feasible reconciliation in correspondence of a nearly increasing subsequence. The main idea is the following: if S' is a nearly increasing subsequence of S, for any leaf l_{i_j} of H corresponding to the i_j-th element of S', except the first one, we map in y_{i_j} (that is the corresponding internal node of l_{i_j}) the internal node of P whose leaf is labeled with the same label as l_{i_j} (formally this node is $x_{\sigma^{-1}(l_{i_j})}$). These internal nodes of P will be the co-speciations of γ. Every other internal node of P, x_k, that has not been mapped yet, is mapped together with x_m with m being the maximum value that is smaller than k and such that x_m is already mapped. In this way, we still have to map the first group of internal nodes of P, $x_1, \ldots, x_t$ with $t = \sigma^{-1}(l_{i_2}) - 1$. The mapping of these latter nodes will be different according to whether S' is increasing or not. However, in both cases, the mapping will guarantee that the first node we mapped $x_{\sigma^{-1}(l_{i_2})}$ will be a co-speciation.

Lemma 4 *Given an instance S, there exists a time feasible reconciliation γ such that $val(\gamma) \geq lnis(S) - 1$.*

Proof Let S' be a nearly increasing subsequence of S. Preliminarily observe that $|S'| \geq 2$; if $|S'| = 2$ then the last element in S is either 0 or 1 (indeed, if there were a value $a \neq 0, 1$, then either $0, 1, a$ or $1, 0, a$ would be a nearly increasing subsequence of S of length 3); the reconciliation mapping all internal vertices of P on the root of H is time feasible and $val(\gamma) = 1$ as $\Sigma_\gamma = \{x_1\}$.

Let now $|S'| > 2$ and $S' = \sigma^{-1}(l_{i_1}), \dots, \sigma^{-1}(l_{i_{r+1}})$ be a nearly increasing subsequence of S, of length $r+1$. We consider two cases: (a) $\sigma^{-1}(l_{i_1}) < \sigma^{-1}(l_{i_2})$ or (b) $\sigma^{-1}(l_{i_2}) < \sigma^{-1}(l_{i_1}) < \sigma^{-1}(l_{i_3})$.

(a) $\sigma^{-1}(l_{i_1}) < \sigma^{-1}(l_{i_2})$ – refer to Fig. 5.a for an example.

We define the following reconciliation γ for all x_i with $1 \leq i \leq n-1$:

$$\gamma(x_i) = \begin{cases} \sigma(0) & 1 \leq i < \sigma^{-1}(l_{i_1}) \\ l_{i_1} & \sigma^{-1}(l_{i_1}) \leq i < \sigma^{-1}(l_{i_2}) \\ y_{l_{i_k}} & \sigma^{-1}(l_{i_k}) \leq i < \sigma^{-1}(l_{i_{k+1}}), \text{ for all } 2 \leq k \leq r \\ y_{l_{i_{r+1}}} & i \geq \sigma^{-1}(l_{i_{r+1}}). \end{cases}$$

Observe that in view of the definition of γ, for every $2 \leq k \leq r+1$, x_{i_k} is mapped to an internal vertex $\gamma(x_{i_k})$. Moreover, its child i_k is mapped in the subtree of $H_{\gamma(x_{i_k})}$ consisting of only one leaf while the other child, $x_{i_{k-1}}$, is mapped in the other subtree.
Hence S' defines r co-speciations being $\Sigma_\gamma \supseteq \{x_{i_2}, x_{i_3}, \dots x_{i_{r+1}}\}$ and so $val(\gamma) \geq r$. It remains to show that γ is a time feasible reconciliation. To see this, observe that both the conditions of Theorem 2 hold for γ.

(b) $\sigma^{-1}(l_{i_2}) < \sigma^{-1}(l_{i_1}) < \sigma^{-1}(l_{i_3})$ – refer to Fig. 5.b for an example.

We define the following reconciliation γ for all x_i with $1 \leq i \leq n-1$:

$$\gamma(x_i) = \begin{cases} \sigma(0) & 1 \leq i < \sigma^{-1}(l_{i_2}) \\ l_{i_2} & \sigma^{-1}(l_{i_2}) \leq i < \sigma^{-1}(l_{i_1}) \\ y_{l_{i_k}} & \sigma^{-1}(l_{i_k}) \leq i < \sigma^{-1}(l_{i_{k+1}}), \text{ for all } 2 \leq k \leq r \\ y_{l_{i_{r+1}}} & i \geq \sigma^{-1}(l_{i_{r+1}}). \end{cases}$$

Using arguments similar to the previous case, we observe that γ is time feasible and S' defines r co-speciations, $\Sigma_\gamma \supseteq \{x_{i_2}, x_{i_2}, \dots x_{i_{r+1}}\}$. Hence, $val(\gamma) \geq r$.

□

Given an instance S, Lemma 3 holds for every time feasible γ and in particular for γ_{OPT}, so we get $lnis(S) \geq val(\gamma_{OPT}) + 1$; on the other hand, $val(\gamma_{OPT}) \geq val(\gamma)$, so from Lemma 4 it comes out $val(\gamma_{OPT}) - 1 \geq lnis(S)$. Observe that $lnis(S)$ can be computed similarly to the length of the longest increasing subsequence of S. Indeed, the latter one can be computed in $O(n \log n)$ time [13,2] where the main idea is to scan the sequence from left to right, maintaining at any given time an efficient representation of all the possible

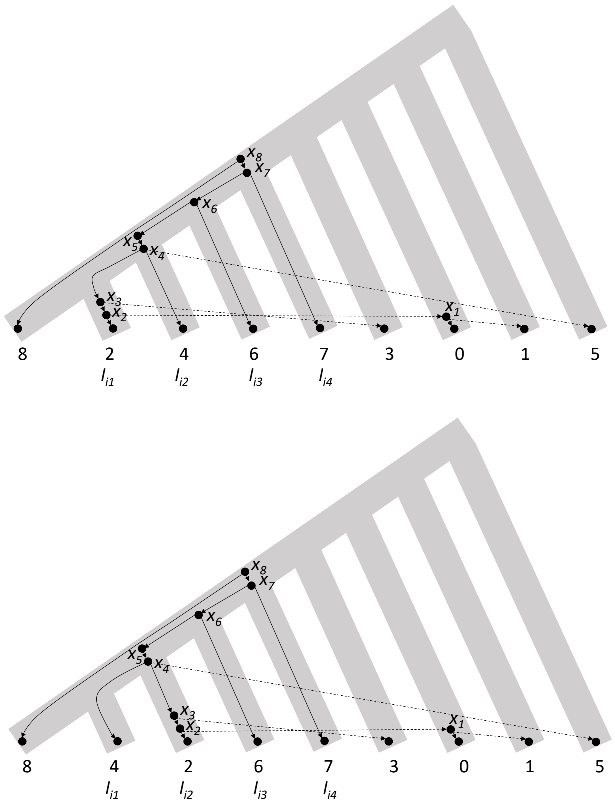

Fig. 5 Two instances where: (a) $S = 8, 2, 4, 6, 7, 3, 0, 1, 5$ and hence $\sigma^{-1}(l_{i_2}) < \sigma^{-1}(l_{i_1})$. (b) $S = 8, 4, 2, 6, 7, 3, 0, 1, 5$ and hence $\sigma^{-1}(l_{i_1}) < \sigma^{-1}(l_{i_2})$.

increasing subsequences that can be formed with the elements seen so far and then applying binary search to extend them. It is easy to see that $lnis(S)$ can be computed in the same way by keeping also the sequences that can be increasing by exchanging the first two positions. This adds a constant multiplicative factor to the comparisons made by the algorithm and then $lnis(S)$ can be computed in $O(n \log n)$. Hence, it follows:

Theorem 4 *Given an instance S, it holds:*

$$val(\gamma_{OPT}) + 1 = lnis(S).$$

and it is possible to find an optimal time feasible reconciliation in polynomial time.

5.1 Behavior of $val(\gamma_{OPT})$ for randomly chosen instances

Given a sequence S, let $lis(S)$ be the length of the longest increasing subsequence; it is obvious that $lis(S) \leq lnis(S) \leq lis(S) + 1$. We exploit some known results for $lis(S)$ to draw some inferences on $lnis(S)$. To do this, we call $\pi(n)$ a permutation chosen uniformly at random on the first n positive integers and define the *expected value of the longest increasing subsequence length over all permutations of order* n as $l_n = \frac{1}{n!}\sum_{\pi(n)} lis(\pi(n))$. It is well known that, for all $n \geq 1$, $l_n \geq \sqrt{n}$ (see for *e.g.* [13]).

The following result shows that in fact the behavior of $lis(\pi(n))$ for a typical permutation $\pi(n)$ is asymptotycally the same as that of its average value. More formally:

Theorem 5 *(Hammersley's convergence theorem [8]) The limit* $\lim_{n\to\infty}\frac{l_n}{\sqrt{n}}$ *exists. Furthermore, for every permutation* $\pi(n)$ *chosen uniformly at random, and for every* $\epsilon > 0$, $\Pr\{|\frac{lis(\pi(n))}{\sqrt{n}} - \Lambda| > \epsilon\} \to 0$ *as* $n \to \infty$*, where* Λ *is a constant.*

It is clear that this result holds also for $lnis(S)$. It follows that, for an instance S, in order to deduce some possible biological correlation between the two caterpillars, we should obtain a similarity that is greater than what is expected by chance, in other words we should have $lnis(S) = \omega(\sqrt{n})$.

6 Properties of reconciliations for caterpillars

If we do not restrict our attention to optimal reconciliations but consider the whole set of time feasible reconciliations, then it is always possible to have both a reconciliation without host switches (*e.g.* by the LCA mapping) and a reconciliation with no duplications (*e.g.* by mapping every internal vertex of the parasite to a leaf of the host). In other words, forbidding either host switches or duplications let us loose the optimality keeping the time feasibility. This is true even when we consider only our special instances. Indeed, Example 1 in Section 4 shows that an optimal time feasible solution without host switches may not exist; the same holds for duplications *e.g.* for instance $S = 3, 1, 8, 2, 7, 5, 0, 4, 6$: by checking exhaustively (e.g. using a tool as Eucalypt [5]) each of its optimal time feasible reconciliations, we realize it contains at least one duplication. Nevertheless, if we drop the time feasibility requirement, we can eliminate duplications keeping the optimality. The next theorem provides a transformation that, given an optimal time feasible reconciliation, outputs a reconciliation without duplications with the same *val* but not necessarily time feasible.

Theorem 6 *Given an instance* S*, there exists always a (not necessarily time feasible) reconciliation* δ *with* $val(\delta) = val(\gamma_{OPT})$ *and* $\Delta_\delta = \emptyset$.

Proof Let $\gamma_{OPT} = \gamma$ be the optimal time feasible reconciliation defined in the proof of Lemma 4. If γ has no duplications then we are done. Otherwise, from Lemma 2, the duplications can be only in the internal vertices of H. Hence, let $y_{d_1}, \ldots, y_{d_f}$ for some $1 \leq f \leq lnis(S)$, the vertices in which at least one duplication is mapped. By construction, for each d_j, let $x_{d_j}, x_{d_j+1}, \ldots, x_{d_j+t_j}$, with $t_j \geq 1$, be the vertices mapped by γ in y_{d_j}. Observe $x_{d_j} \in \Sigma_\gamma$ and all the others are in $\Delta_\gamma \cup \Theta_\gamma$. Let t'_j be the largest integer for which $x_{d_j+t'_j}$ is a duplication. For each $j = 1, \ldots, f$ we construct reconciliation δ from γ by setting $\delta(x_{d_j}) = \ldots = \delta(x_{d_j+t'_j-1}) = l_{d_j}$ (see Figure 6). On all the other vertices, δ coincides with γ. Notice that:

(i) x_{d_j} was a co-speciation in γ and is a host switch in δ;
(ii) $x_{d_{j+1}}, \ldots, x_{d_j+t'_j-1}$ were either duplications or host switches in γ and are all host switches in δ;
(iii) $x_{d_j+t'_j}$ was a duplication in γ and becomes a co-speciation in δ;
(iv) $x_{d_j+t'_j+1}, \ldots, x_{d_j+t_j}$ remain host switches when passing from γ to δ.

Clearly $val(\delta) = val(\gamma)$. As shown in Figure 6, δ is not necessarily time feasible. □

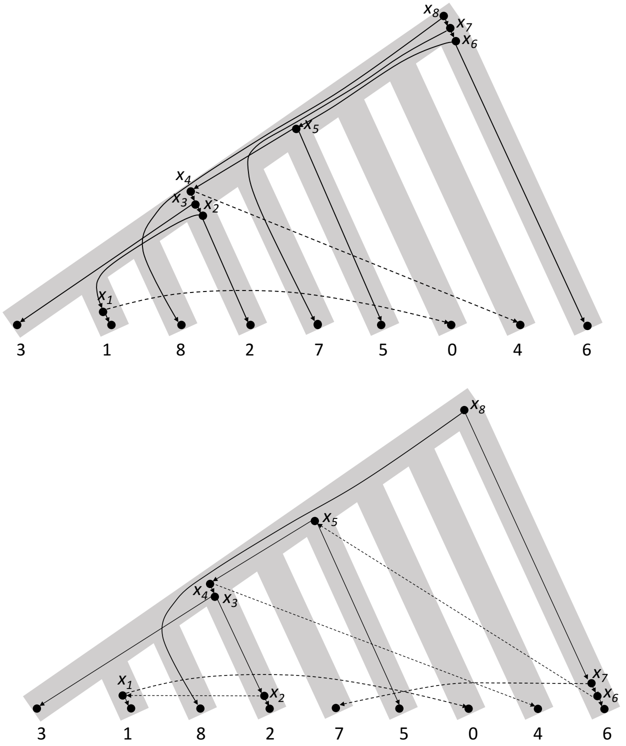

Fig. 6 $S = 3, 1, 8, 2, 7, 5, 0, 4, 6$; (left) optimal time feasible reconciliation γ as defined in the proof of Lemma 4; (right) reconciliation δ as defined in the proof of Theorem 6; δ is time infeasible because of the pair of host switches $(u, v) = (x_7, 7)$ and $(u', v') = (x_6, x_5)$; the addition of arc $(par(\gamma(7)), \gamma(x_5))$ of kind (2d) creates a cycle in G_δ.

7 Conclusions

In this paper we have considered the problem of finding time feasible reconciliations with the maximum number of co-speciations in the very special case in which both the host and the parasite trees are caterpillars of the same size and function σ is a bijection.

Many open questions and possible research lines raise from this work:

- A natural step consists in relaxing our constraints. Namely, a first question is to study the maximum co-divergence problem for two caterpillars when σ is not a bijection anymore. This research could also lead to a tight relation between the number of co-speciations and the maximum length of a common subsequence with more general constraints than the nearly increasing one.

  Moreover, it is certainly interesting to investigate other tree topologies for H and P such as for example complete binary trees. This approach could lead to design approximation algorithms that locally solve TF-MCD on substructures of the input phylogenies that can be reconciled optimally in polynomial time.

- Recall that using the transformation described in Section 6 it is possible to transform an optimal reconciliation to a reconciliation of the same value that does not contain any duplication, but without necessarily ensure the time feasibility. An interesting question may be to reach an optimal time feasible reconciliation that has the minimum number of duplications. A first idea is to procede in a greedy fashion, removing one by one the duplications of an optimal time feasible reconciliation while keeping time feasibility, but this simple procedure seems not to always lead to the optimum.

References

1. Bansal, M.S., Alm, E., Kellis, M.: Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics **28**(12), i283–i291 (2012)
2. Bespamyatnikh, S., Segal, M.: Enumerating longest increasing subsequences and patience sorting. Information Processing Letters **76**(1), 7 – 11 (2000)
3. Charleston, M.A.: Jungles: a new solution to the host/parasite phylogeny reconciliation problem. Mathematical Biosciences **149**(2), 191–223 (1998)
4. Charleston, M.A.: Recent results in cophylogeny mapping. Advances in Parasitology **54**, 303–330 (2003)
5. Donati, B., Baudet, C., Sinaimeri, B., Crescenzi, P., Sagot, M.: Eucalypt: efficient tree reconciliation enumerator. Algorithms for Molecular Biology **10**(1), 3 (2015). URL http://www.almob.org/content/10/1/3
6. Doyon, J.P., Ranwez, V., Daubin, V., Berry, V.: Models, algorithms and programs for phylogeny reconciliation. Brief. Bioinform. **12**(5), 392–400 (2011)
7. Doyon, J.P., Scornavacca, C., Gorbunov, K.Y., Szöllősi, G.J., Ranwez, V., Berry, V.: An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In: E. Tannier (ed.) Proceedings of the 8th annual RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG 2010), *Lecture Notes in*

Bioinformatics, vol. 6398, pp. 93–108. Spring-Verlag Berlin Heidelberg, Ottawa, Canada (2011)

8. Hammersley, J.M.: A few seedlings of research. Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability (Univ. California, Berkeley, Calif., 1970/1971) pp. 345–394 (1972)
9. Merkle, D., Middendorf, M.: Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. Theory in Biosciences **123**, 277–299 (2005)
10. Ovadia, Y., Fielder, D., Conow, C., Libeskind-Hadas, R.: The cophylogeny reconstruction problem is NP-complete. Journal of Computational Biology **18**(1), 59–65 (2011)
11. Page, R.D.M.: Parallel phylogenies: reconstructing the history of host-parasite assemblages. Cladistics **10**(2), 155–173 (1994)
12. Page, R.D.M.: Tangled Trees: Phylogeny, Cospeciation and Coevolution. The University of Chicago Press, Chicago (2003)
13. Romik, D.: The Surprising Mathematics of Longest Increasing Subsequences. Institute of Mathematical Statistics Textbooks. Cambridge University Press (2015)
14. Ronquist, F.: Reconstructing the history of host-parasite associations using generalised parsimony. Cladistics **11**(1), 73–89 (1995)
15. Stolzer, M.L., 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), i409–i415 (2012)
16. Tofigh, A., Hallett, M., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. Journal of IEEE/ACM Transactions on Computational Biology and Bioinformatics **8**(2), 517–535 (2011)