1-1-2009

On the Computational Complexity of the Reticulate Cophylogeny Reconstruction Problem

Ran Libeskind-Hadas
Harvey Mudd College

Michael A. Charleston
University of Sydney

Recommended Citation

R. Libeskind-Hadas and M. Charleston, "On the Computational Complexity of the Reticulate Cophylogeny Reconstruction Problem," Journal of Computational Biology, Vol. 16, No. 1, January 2009, pp. 105-117. DOI: 10.1089/cmb.2008.0084

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 1, 2009

Pp. 105–117
DOI: 10.1089/cmb.2008.0084

On the Computational Complexity of the Reticulate Cophylogeny Reconstruction Problem

RAN LIBESKIND-HADAS[1] and MICHAEL A. CHARLESTON[2]

ABSTRACT

The cophylogeny reconstruction problem is that of finding minimal cost explanations of differences between evolutionary histories of ecologically linked groups of biological organisms. We present a proof that shows that the general problem of reconciling evolutionary histories is NP-complete and provide a sharp boundary where this intractability begins. We also show that a related problem, that of finding Pareto optimal solutions, is NP-hard. As a byproduct of our results, we give a framework by which meta-heuristics can be applied to find good solutions to this problem.

Key words: coevolution, cophylogeny, computational complexity, NP-completeness.

1. INTRODUCTION

IT IS COMMONLY RECOGNIZED that many biological questions must be answered within an evolutionary framework, particularly those comparing taxonomically related organisms. Logically, this extends to studies of the relationships among groups of ecologically linked organisms, such as parasites and their hosts, or genes and the species that house them. The study of coevolution thus requires a theory of *cophylogenetics*, just as any study involving a comparison of species requires the field of phylogenetics. Cophylogenetics relies on estimating relationships among species that are no longer present, by making inferences based on "known" phylogenetic histories of groups of organisms. That is, we are presented with the histories of two ecologically linked taxonomic units (taxa), such as parasites and hosts, and a set of known associations between them, and must reconstruct the ancestral associations. The known associations are perforce from the present: there is generally no fossil evidence to support particular associations of parasites and hosts.

We focus on one of the major methods called *cophylogeny mapping*. With this approach, we consider two histories represented as phylogenetic networks, and the associations between their tips. We attempt to map one network into the other in order to construct a set of coevolutionary events that best explains the current observations.

Naturally this approach has its supporters and detractors but we note that it is generally agreed to be highly intuitive in that the solutions presented are readily interpreted. Moreover, the method is general

[1]Department of Computer Science, Harvey Mudd College, Claremont, California.
[2]School of Information Technology, Sydney Bioinformatics, and Centre for Mathematical Biology, University of Sydney, Sydney, Australia.

in that it can accommodate a fairly complex model of coevolution, with multiple event costs and sound statistical testing of the level of inferred congruence between the two histories. The main issue with cophylogeny mapping is not that it relies on estimates of either history (which is usually the case though there are ways to deal with the inherent uncertainty in that process), but that it is computationally very intensive. In typical studies with more than a dozen or so taxa, existing mapping methods for finding optimal solutions rapidly become unfeasible, particularly when the phylogenies are only slightly, or not at all, congruent.

In the standard cophylogeny reconstruction problem, we are given a host tree H, a parasite tree P, a function φ mapping the leaves (extant taxa) of P to the leaves of H, and costs associated with each of four biologically plausible operations: codivergence, duplication, host switching, and loss (Fig. 1). The objective is to find a least cost association of the trees that can be constructed with the four permitted operations. Note that it is possible to perform such analyses with a more general costing scheme by using the concept of Pareto optimality. As it is not generally possible to assign costs to the events leading to similarity and difference between the two phylogenies, the four event costs can be left unassigned and subject only to the constraint that codivergence is less costly than the other events.

Algorithms for the cophylogeny reconstruction problem in tree phylogenies have been studied by a number of researchers (Charleston, 1998; Merkle and Middendorf, 2005; Page, 1994; Ronquist, 1995, 1998). However, not all phylogenies are trees; there are many cases in which hybridization results in new species, resulting in reticulate phylogenies (Brockelman and Gittins, 1984; Harrison and Rand, 1989; Marshall and Brockelman, 1986; Xu, 2000). Thus, there is practical interest in generalizing the cophylogeny reconstruction problem to *phylogenetic networks*: directed acyclic graphs in which there is a single source of in-degree zero and any number of sinks of out-degree zero (Charleston, 2002). Vertices of in-degree greater than one correspond to hybridization events and nodes of out-degree greater than one correspond to divergence events.

As far as we are aware, there are no other methods for comparing evolutionary trees with hybridization events that capture the asymmetry we observe in the cophylogeny problem. While Bordewich and Semple found (Bordewich and Semple, 2007) that minimum hybridization is NP-Hard, their problem is in terms of different possible trees for the same leaf set; similarly, Than et al.'s PhyloNet package (Than, Ruths, and Nakhleh, 2008) compares networks on the same taxa, rather than on ecologically linked systems such as hosts and parasites.

The reconstruction problems for both phylogenetic trees and networks is generally believed to be computationally intractable. However, in over a decade of study of these problems, no proofs of their intractability have been reported. To the best of our knowledge, this paper makes the first contribution in this direction.

Specifically, we show that the cophylogeny reconstruction problem is NP-complete when the host phylogeny is reticulate, even if the parasite phylogeny is a tree. Moreover, we provide a sharp boundary between where this problem is polynomial-time solvable and where it becomes NP-complete. As explained

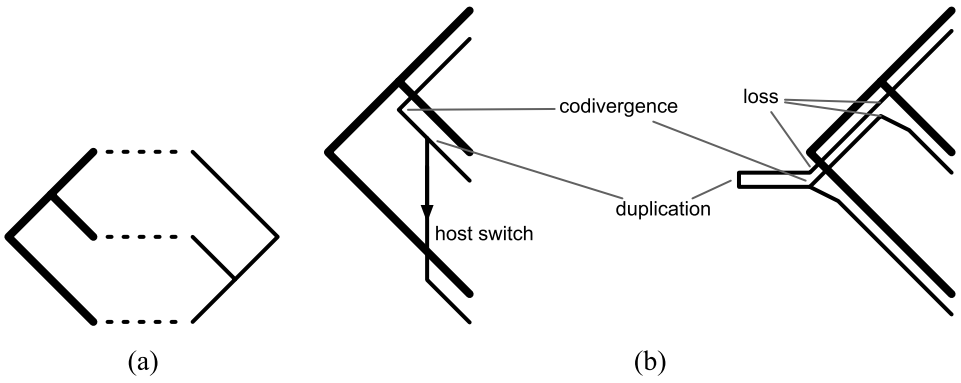

FIG. 1. **(a)** A simple tanglegram with host tree H (heavy lines, left) and parasite tree P (right), and the associations φ between their tips (dashed); **(b)** two possible reconstructions that explain the relationship between H and P.

at the end of Section 4, our NP-completeness result implies that the Pareto case is also NP-hard. Finally, we describe a meta-heuristic-based approach to finding good, but not necessarily optimal, solutions and provide a portable and easily modifiable implementation as a research tool.

2. PREVIOUS WORK

Most earlier work on the cophylogeny reconstruction problem assumes that the host and parasite phylogenies are trees. When host switches are not permitted in the trees, the problem can trivially be solved optimally in linear time (Ma et al., 2000; Zhang, 1997). However, the presence of host switching operations evidently makes the problem much harder. Therefore, previous work has consisted of heuristics (Merkle and Middendorf, 2005; Ronquist, n.d.), which are fast but do not guarantee optimal solutions, and exact algorithms, which take exponential time in the worst case but provide optimal solutions (Charleston, 1998; Page, 1994; Ronquist, n.d.). The problem of finding optimal solutions for trees has been widely regarded as computationally intractable, but no proof of intractability has been discovered to date.

Assuming that the order of the divergence events in the host tree is known, Ronquist (1995, 1998) sketches several algorithms for finding optimal solutions in polynomial time but observes that these approaches can take exponential time if no ordering is known *a priori*. Charleston (1998) advances a graph theoretic approach using the notion of jungles to solve the problem optimally, but the algorithm also requires exponential time in general. This approach is used in the TreeMap software tool (Charleston and Page, n.d.). Merkle and Middendorf (2005) propose heuristics that work well in practice but do not guarantee optimal solutions and these heuristics are used in the Tarzan software tool (Merkle and Middendorf, n.d.). It should be noted that one of the subproblems that arises in the Tarzan approach is NP-complete, and it is this subproblem that necessitates heuristics. However, the fact that this particular approach involves a NP-complete subproblem does not imply that the cophylogeny problem itself is NP-complete, but rather that this particular approach to the problem involves a computationally intractable step.

3. TERMINOLOGY, NOTATION, AND PROBLEM STATEMENT

A *phylogenetic network* is a connected directed acyclic graph (DAG) $G = (V, E)$ with the following properties:

- There is a single node of in-degree zero called the "source."
- No node has both in-degree and out-degree equal to one.
- No node has both in-degree and out-degree greater than one.
- All nodes of out-degree zero have in-degree one. These nodes are called the *tips* of the network.
- The maximum in-degree and out-degree over all nodes is bounded by some constant. (Typically this constant is two, but larger constants are permitted.)

Each edge corresponds to the lifetime of a taxonomic unit (e.g., species), and each internal (non-tip) node corresponds to a divergence or hybridization event involving that taxon. However, in the host network, the source node is restricted to have a single out-going edge to a node that represents the most recent common ancestor (mrca) of the tips. This is required in order to account for events in the parasite phylogeny that predate the mrca in the host phylogeny. In the parasite network, the source node represents the mrca of its tips.

Internal nodes of in-degree one correspond to divergence events whereas non-source nodes of out-degree one correspond to hybridization events. Edges ending at tips of the network correspond to extant taxa. Note that in the special case that all non-source nodes have in-degree one, the network is a tree. In this case, the tips are the leaves of the tree.

By analogy to trees, we say that a node v is a *child* of node u if there is an edge (u, v) in the network. Similarly, we say that v is a *descendant* of u, if there is a path from u to v. (Note that by this definition each node is a descendant of itself.) It is important to note that in this formulation nodes correspond to

events. Thus, the terms "children" and "descendants" refer to the topology of the networks rather than the ancestry of taxa.

Let H and P denote the host and parasite networks, respectively. We consider a map in which every node p in P is associated with either a node or an edge in H. The node or edge with which p is associated is called the *image* of p with respect to this map. We assume a conventional model with four event types: codivergence, duplication, host switching, and loss.

We consider the events that give rise to nodes in P, other than the source and its unique adjacent node. Given an edge (p, q) in P, if p is associated with a node v in H then q arose from codivergence at v. If, on the other hand, p is associated with an edge (u, v) in H then q arose from a duplication on (u, v) and further, if q is associated with an edge that is not a descendant of (u, v), then a host switch also occurred. It is not permitted for all children of p to undergo host switches because this would lead to untraceable and biologically unreasonable solutions (Charleston, 2002).

Loss arises from any one of three processes: lineage sorting, extinction, and sampling failure. Extinction and sampling failure are indistinguishable from the remaining process of lineage sorting, which arises when a path from the image of p to the image of q contains at least one node of H as an intermediate node. The number of such intermediate nodes is the number of losses incurred on that path. Additionally, if parasite node p is associated with host node v, indicating a codivergence, then the out-degree of p, $out(p)$, must be less than or equal to the out-degree of v, $out(v)$. If the out-degrees are not equal, then it assumed that $out(v) - out(p)$ losses are incurred. This treatment enables us to decouple evolutionary event costs from the nodes in P and allows us to treat multifurcating phylogenies appropriately.

The divergence and hybridization events represented in a phylogenetic network may occur at different times, resulting in different overlaps in the lifetimes of the taxa and thus different sets of feasible solutions to the reconstruction problem. From a computational perspective, the precise actual times are unimportant; the relative times of these events suffice since only relative times are needed to determine where host switching may or may not occur. Further, dating divergence times in phylogenetics is notoriously difficult and error-prone, so it is a conservative treatment to use only their relative times (Ho et al., 2005).

The notion of relative time is captured as follows: Given host network $H = (V_H, E_H)$ and parasite network $P = (V_P, E_P)$, let $n_H = |V_H|$ and $n_P = |V_P|$. In general the $n_H + n_P$ events in these two networks can occur in at most $n_H + n_P$ distinct *relative times*. (In fact, this is a generous overestimate since in most cases all of the tips can be assumed to be contemporaneous at "current time.") These possible relative times are represented by the set of positive integers $T = \{1, \ldots, n_H + n_P\}$. Let $\mathcal{I}$ denote the set of subsets of the form $\{a, a+1, \ldots, b-1, b\}$ where $a, b \in T$ and $a \leq b$. A *time* function $t_H : V_H \to \mathcal{I}$ assigns each event in H with a set of relative times when the event may have occurred. Similarly, a time function $t_P : V_P \to \mathcal{I}$ assigns each event in P with a set of relative times when the event may have occurred.[1]

An instance of the Generalized Cophylogeny Reconstruction Problem (GCRP) comprises a 6-tuple $(H = (V_H, E_H), P = (V_P, E_P), t_H, t_P, \varphi, \kappa)$ where H is the host network, P is the parasite network, t_H is the time function for H, t_P is the time function for P, φ is a mapping from the extant taxa of P into the extant taxa of H, and κ is a 4-tuple cost vector $(\kappa_C, \kappa_D, \kappa_S, \kappa_L)$ representing the costs of codivergence, duplication, host switching, and loss events, respectively. The objective is to find a mapping $\Phi : P \to H$ that extends φ, can be constructed by a set of codivergence, duplication, host switching, and loss events with respect to the given time functions t_H and t_P, and is of minimum total cost with respect to the given cost vector.

The sharp complexity boundary that we prove here comprises two results:

Theorem 1. GCRP *is solvable in polynomial time for the set of instances* $(H = (V_H, E_H), P = (V_P, E_P), t_H, t_P, \varphi, \kappa)$ *such that* P *is a tree and for all* $v \in V_H$, $|t_H(v)| = 1$.

Theorem 2. *The decision problem associated with* GCRP *is NP-complete for the set of instances* $(H = (V_H, E_H), P = (V_P, E_P), t_H, t_P, \varphi, \kappa)$ *such that* P *is a tree and for all* $v \in V_H$, $|t_H(v)| \leq 2$.

[1]The assumption that the relative times associated with an event form a consecutive set of integers is biologically motivated but the computational results in this paper apply for the more general case that the sets are arbitrary.

Theorem 1 states that if the relative times of all of the events in H are fixed and P is a tree, the reconstruction problem can be solved in polynomial time. Related ideas have been sketched for the reconstruction problem for pairs of trees (Ronquist, 1998), but differences arise when the host phylogeny may be reticulate. This result also forms the basis of efficient heuristics presented in Section 5.

Theorem 2, our main contribution, shows that if the "fixed time" constraint of Theorem 1 is only slightly relaxed, so that some events may have occurred at one of two different times, the problem becomes computationally intractable.

3.1. A polynomial time algorithm for the "fixed time" case

In this section, we give a polynomial time dynamic programming algorithm under the "fixed time" assumption of Theorem 1. We describe a conceptually simple algorithm to establish the polynomial time bound, although optimizations are possible that further improve the running time.

Let $(H = (V_H, E_H), P = (V_P, E_P), t_H, t_P, \varphi, \kappa)$ be an instance of GCRP such that P is a tree and for all $v \in V_H$, $|t_H(v)| = 1$. Since $|t_H(v)| = 1$ for all $v \in V_H$, we henceforth use $t_H(v)$ to denote either the set or the unique element of that set, where the usage is clear from context. Given two nodes $u, v \in V_H$, the distance from u to v, denoted $dist(u, v)$, is the length of the shortest path from u to v in H and is ∞ if v is not reachable from u.

The *lifetime* of (u, v) in E_H, denoted $\ell(u, v)$, is defined to be the set $\{t_H(u) + 1, \ldots, t_H(v)\}$. An event $p \in V_P$ may be associated with the lineage $(u, v) \in E_H$ at time t if $t \in t_P(p) \cap \ell(u, v)$. Note that in this formulation, a parasite node p is always associated with an edge (u, v) in the host network. A codivergence event arises when a parasite is associated with (u, v) at the endpoint of the lifetime, $t_H(v)$, which effectively associates node p with node v. A duplication event arises when a parasite is associated with (u, v) at some time before $t_H(v)$.

In order to correctly account for loss events, we define the *loss count* function, $losses((u, v), t, (y, z))$, as follows:

1. If $t = t_H(v)$ then $losses((u, v), t, (y, z)) = dist(v, y)$.
2. If $t \neq t_H(v)$ then $losses((u, v), t, (y, z)) = 0$ if $(y, z) = (u, v)$ and otherwise $losses((u, v), t, (y, z)) = dist(v, y) + 1$.

Next, let $cost((p, (u, v), t), (q, (y, z), t'))$ denote the cost that arises from inducing association $(q, (y, z), t')$ from association $(p, (u, v), t)$ where q is a child of p and $t' > t$. The *cost* is defined as follows:

Case 1: $t = t_H(v)$. If y is a descendant of v then a codivergence event is indicated and

$$cost((p, (u, v), t), (q, (y, z), t')) = \kappa_C + \kappa_L losses((u, v), t, (y, z))$$

and otherwise $cost((p, (u, v), t), (q, (y, z), t')) = \infty$.

Case 2: $t \neq t_H(v)$. There are two subcases:

1. If $(y, z) = (u, v)$ or y is a descendant of v then a duplication event is implied and $cost((p, (u, v), t), (q, (y, z), t')) = \kappa_D + \kappa_L losses((u, v), t, (y, z))$.
2. If $(y, z) \neq (u, v)$ and y is not a descendant of v then a duplication followed by a host switch event is implied. In this case, if there exists some edge $(w, x) \in E_H$ such that $t \in \ell(w, x)$ and $(w, x) = (y, z)$ or y is a descendant of x then

$$cost((p, (u, v), t), (q, (y, z), t')) = \kappa_D + \kappa_S + \min_{(w,x) \in E_H \text{ s.t. } t \in \ell(w,x)} \kappa_L losses((w, x), t, (y, z))$$

and otherwise $cost((p, (u, v), t), (q, (y, z), t')) = \infty$.

Finally, we compute an optimal cost solution via dynamic programming. Let τ be a dynamic programming table with dimensions $|V_P| \times |E_H| \times (|V_H| + |V_P|)$ where $\tau(p, (u, v), t)$ denotes the least cost over

all solutions for the subtree of P rooted at p, assuming event p occurs on host (u, v) at time t. When p is a tip, v is a tip, $t \in t_P(p) \cap t_H(v)$, and $\varphi(p) = v$, we set $\tau(p, (u, v), t) = 0$. All tips of P are now marked as "visited." All other entries of τ are initialized to ∞.

In the dynamic programming step, we consider each p such that all of its descendants in P have been previously marked as "visited." Let $children(p)$ denote the set of children of p. For each $(u, v) \in E_H$ and each $t \in t_P(p) \cap \ell(u, v)$ compute $\tau(p, (u, v), t)$ as follows:

- If $t_H(v) = t$ and v is a divergence node then let S denote a set of associations with the following property: For each $q_i \in children(p)$ there exists a single association $(q_i, (y_i, z_i), t_i) \in S$ and $t_i > t$. The set S must have the property that there exists a set of paths in H such that each path starts at v, has last edge (y_i, z_i), and no two such paths have their first edge in common. Let $\mathcal{S}$ denote the set of all such sets S. Then

$$\tau(p, (u, v), t) = \min_{S \in \mathcal{S}} \sum_{a \in S} cost(a) + \tau(a)$$

  In the event that the out-degree of the parasite node p, $out(p)$, is less than the out-degree of v, $out(v)$, then we must add $\kappa_L(out(v) - out(p))$ to the value of $\tau(p, (u, v), t)$.
- If $t_H(v) \neq t$ then let T denote a set of associations with the following property: For each $q_i \in children(p)$ there exists a single association $(q_i, (y_i, z_i), t_i) \in T$ and $t_i > t$, as before. The set T must have property that at least one (y_i, z_i) is descendant from v. Let $\mathcal{T}$ denote the set of all such sets T. Then

$$\tau(p, (u, v), t) = \min_{T \in \mathcal{T}} \sum_{a \in T} cost(a) + \tau(a)$$

When $\tau(p, (u, v), t)$ has been computed for every (u, v) and t, node p is marked as "visited" and the process is repeated until all nodes in P are marked as "visited."

While the algorithm described here computes the cost of an optimal solution, the standard method of keeping annotations in the dynamic programming table can be used to reconstruct the actual optimal solutions. The correctness of this algorithm can be verified by induction. The algorithm can easily be shown to run in time polynomial in the size of the two networks $H = (V_H, E_H)$ and $P = (V_P, E_P)$: The size of the dynamic programming table is $O(|V_P| \times |E_H| \times (|V_H| + |V_P|))$. In order to compute a single entry $\tau(p, (u, v), t)$ in the table, we may consider each combination of associations for the children of p. Let Δ denote the constant upper-bound on in-degree and out-degree. Each of the (at most) Δ children of p can be associated with $O(|E_H|)$ distinct edges at $O(|V_H| + |V_P|)$ distinct times, resulting in a total of $O((|E_H| \times (|V_H| + |V_P|))^{\Delta})$ constant-time lookups into the dynamic programming table. Along with each lookup is an accompanying computation of the *cost* function, each of which requires at most $O(|E_H|)$ invocations of breadth-first search to compute the loss counts. Each breadth-first search, in turn, takes time $O(|V_H| + |E_H|)$. Thus, the worst-case running time is polynomial in the size of the two networks. This completes the proof of Theorem 1.

4. NP-COMPLETENESS

Next, we show that the problem becomes NP-complete when some host taxa can occur at one of two relative times. For precision, we define the Generalized Cophylogeny Reconstruction Decision Problem (GCRDP) as follows:

GCRDP

Instance: Given $(H = (V_H, E_H), P = (V_P, E_P), t_H, t_P, \varphi, \kappa)$, and a cost K.
Question: Does there exist a reconstruction whose cost is K or less?

We prove that GCRDP is NP-complete even when restricted to instances such that that P is a tree and for all $v \in V_H$, $|t_H(v)| \leq 2$. The reduction is from 3-SAT which is stated as follows and is known to be NP-complete (Garey and Johnson, 1979):

3-SAT

Instance: Given a collection of n Boolean variables and m clauses each comprising the disjunction of three literals over the given variables.

Question: Does there exist a valuation of the variables that satisfies all of the clauses?

For convenience, we assume that the in- and out-degrees of nodes can be arbitrary. We can then simply replace any high degree node in our reduction by a tree (of polynomial size) of degree two nodes.

Proof of Theorem 2. First, the problem is clearly in the class NP since a valid solution can be verified in polynomial time. We show hardness by a reduction from 3-SAT.

Let n denote the number variables and let m denote the number of clauses in the given 3-SAT instance. Our reduction consists of several types of gadgets for the host network. In these gadgets, some sink nodes in the host network will not be labeled since they will not be the images of any nodes in the domain of φ.

A *k-thorn gadget*, illustrated in Figure 2a, is a directed rooted proper binary tree in which the subgraph of internal nodes is a directed path of length k (the value of k will be determined later). We introduce k consecutive time slots where the root of every k-thorn gadget is fixed by the function t_H to occur in the first of these slots and each descendant, or *thorn*, at distance i from the root of the gadget occurs i time slots later. The time slots of the leaves are assumed to be a common "current time" as shown in Figure 2b. For clarity, the representation shown in Figure 2c is used henceforth to represent a k-thorn gadget. At most one leaf of a k-thorn gadget will be of interest to us, and thus only one leaf is exposed in this representation.

For each variable x_i in the given 3-SAT instance, we introduce a corresponding *variable gadget*. We describe this gadget in two parts. The first part of the variable gadget is the *truth setting component* shown in Figure 3a. Some of these nodes will have additional in-coming or out-going edges to other types of gadgets, described below. All nodes α_i, $1 \leq i \leq n$, occur at a common fixed time t. Each node T_i and F_i may occur at time $t+1$ or $t+2$, $1 \leq i \leq n$. All nodes β_i, γ_i, and a_i, $1 \leq i \leq n$, occur at times $t+3$, $t+4$, and $t+6$, respectively, where time $t+6$ is current time. Time $t+5$ will be used later to connect gadgets.

Note that there are four possible combinations of times for each pair T_i and F_i as shown in Figure 3a–3d. Other gadgets, described later, will be used to force one of T_i and F_i to occur at time $t+1$ and the other at time $t+2$, as shown in Figure 3c and 3d. The former case will correspond to $x_i = \text{TRUE}$ and the latter to $x_i = \text{FALSE}$.

Next, we assemble k-thorn and truth setting components to construct a *variable gadget* for each variable x_i as shown in Figure 4. Each x_i, $1 \leq i \leq n$, has a unique time and all nodes b_i occur at current time. The thorns on the k-thorn gadget occur at fixed times before the time t associated with α_i.

For each clause C_j we introduce a *clause gadget* as shown in the box on the left side of Figure 5. The clause gadget consists of a node C_j with two children: one the root of a k-thorn gadget with a leaf labeled C'_j and the other a node S_j. Vertex S_j has three outgoing edges representing the three literals that can satisfy C_j. Specifically, if literal x_i appears in clause C_j then there is an edge from S_j to node

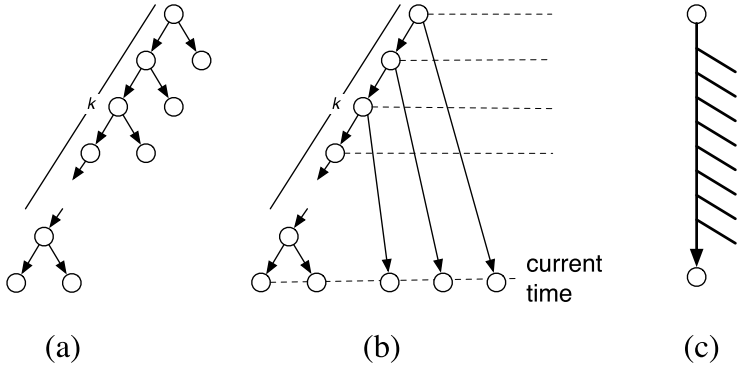

FIG. 2. **(a)** The k-thorns gadget. **(b)** The gadget with times indicated by dashed lines. **(c)** The representation of the k-thorns gadget used in the remainder of the proof.

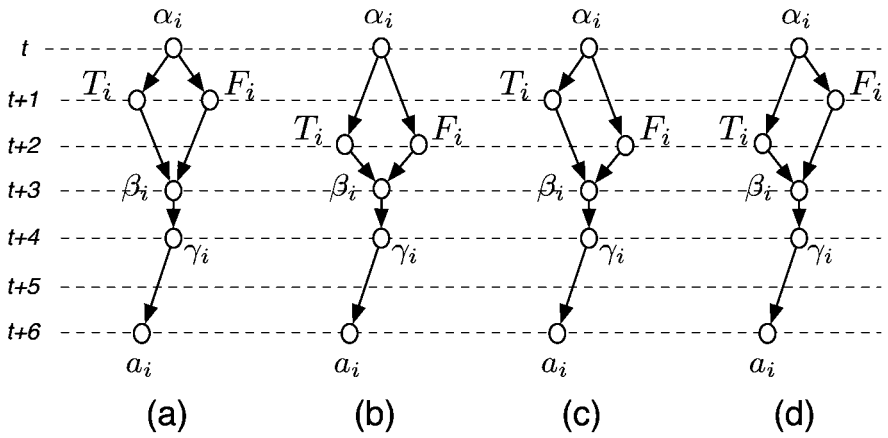

FIG. 3. The truth setting component in each of four possible configurations **(a)–(d)**. The configuration in (c) corresponds to x_i = TRUE, and the configuration in (d) corresponds to x_i = FALSE.

T_i in the variable gadget for x_i. Conversely, if $\overline{x}_i$ appears in C_j, then there is an edge from S_j to node F_i in the variable gadget for x_i. Each node C_j and S_j is set to occur at a time distinct from any other event. We introduce a pair of nodes δ_j, C_j'' with an edge from δ_j to C_j'' for each clause, setting C_j'' to occur at current time and δ_j to occur at the time immediately prior. For each variable x_i that occurs in C_j, we introduce an edge from node γ_i in the x_i variable gadget to the node δ_j. This is illustrated in Figure 5 for the case that literal x_i appears in clause C_j.

Next, we construct the host network and parasite tree as follows: The host network, shown in Figure 6a, comprises a source node h with an edge to a node h'. From h', there is an edge to each variable gadget and to each clause gadget.

The parasite tree, shown in Figure 6b with times aligned to the host network, is relatively simple. There is a source p' that is set to be contemporaneous with h'. For each node x_i in a variable gadget in the host tree, there is a corresponding node x_i in P, occurring at a unique earlier time slot, with two children labeled y_i and z_i, that have fixed times of $t+2$ and $t+1$, respectively. Each of y_i and z_i have two leaf children labeled a_i and b_i occurring at current time. Finally, for each clause C_j there is an edge from p' to a node labeled C_j occurring at time $t+2$, with two leaf children C_j' and C_j'' occurring at current time. We define the mapping φ from the tips of P into the tips of H by $\varphi(p) = h$ if and only if the label of tip p is equal to the label of tip h in our construction. Note that some nodes in P have identical labels,

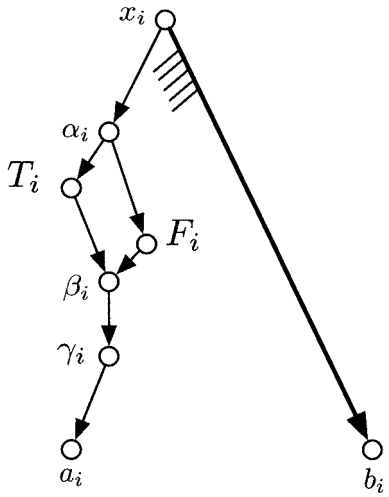

FIG. 4. The variable gadget is comprised of a truth setting component and a k-thorn gadget.

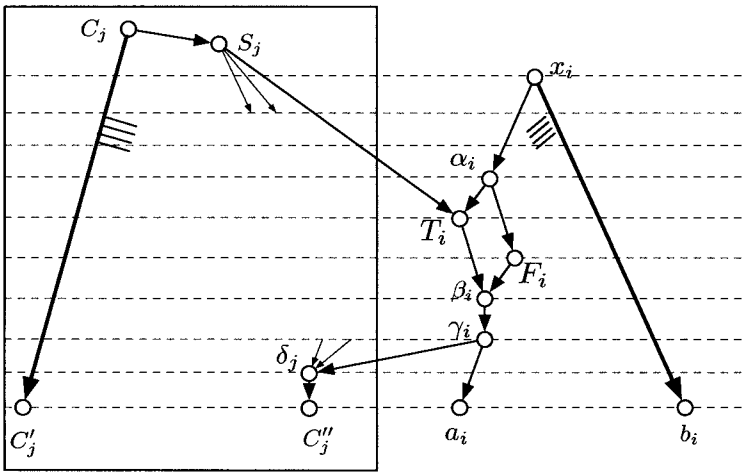

FIG. 5. The clause gadget is shown in the box on the left. The interactions with a variable gadget are shown for the case that literal x_i occurs in clause C_j.

which is permitted since φ need not be one-to-one, potentially permitting multiple parasites to reside on the same host. This completes the description of the instance of GCRDP.

The costs of the four permitted events are as follows: Codivergence has cost 0 and all other events have cost 1. The value of the decision parameter, K, is $18n + 9m$. (Recall that n denotes the number of variables and m denotes the number of clauses.) Let k, the number of thorns in each k-thorn gadget, be $18n + 9m + 1$. Thus, the total number of nodes and edges in H and P is polynomial in $n + m$ and the reduction can be completed in polynomial time.

We now show that the 3-SAT instance is satisfiable if and only if the answer to the constructed GCRDP problem is "yes." First, assume that there is a satisfying valuation for the 3-SAT instance. We associate p'

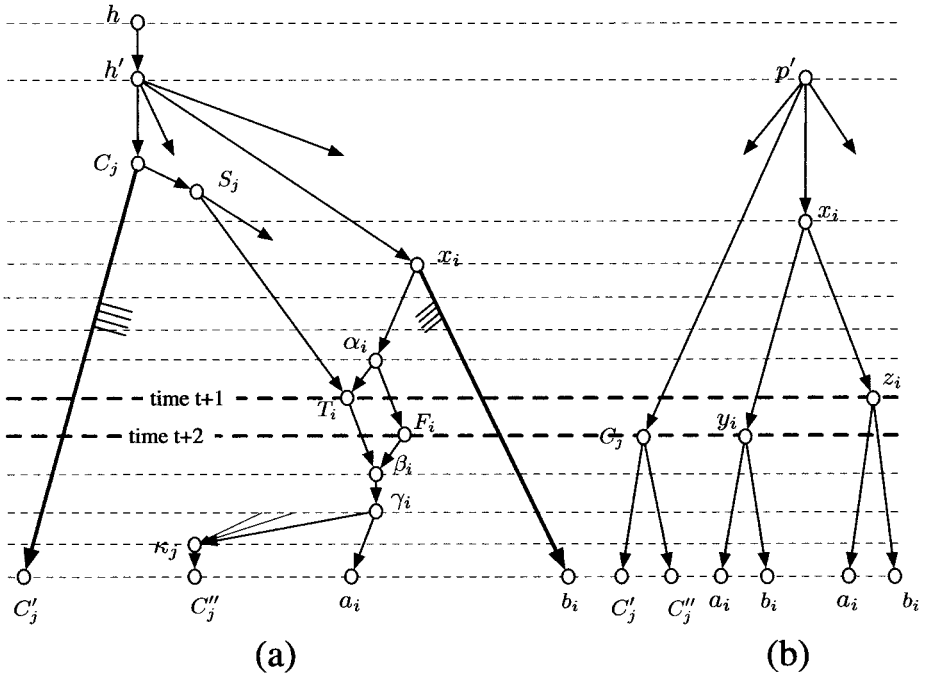

FIG. 6. **(a)** The host network. **(b)** The parasite tree. The special times $t + 1$ and $t + 2$ are indicated in heavy dashed lines.

with h'. If x_i is TRUE, we choose time $t+1$ for T_i and $t+2$ for F_i, and vice versa if x_i is FALSE. Each x_i in P associates with edge (h', x_i) and two duplication events are incurred for its children, contributing $2n$ to the total cost. If T_i is at time $t+1$ and F_i is at time $t+2$ in H then node y_i in P associates with edge (T_i, β_i) at time $t+2$ whereas node z_i in P associates with edge (α_i, F_i) at time $t+1$. For node y_i two duplication events are incurred and its b_i child host switches to the edge (x_i, b_i). Similarly, z_i incurs two duplication events at time $t+1$ and its b_i child host switches to edge (x_i, b_i). The case that T_i is at time $t+2$ and F_i is at time $t+1$ is analogous. In each case, the a_i children of y_i and z_i associate with the a_i tip in H while the b_i children, as a result of the host switch, associate with the b_i tip in H. This involves an additional four duplication events, two host switching events, and ten loss events per variable. Finally, for each clause C_j, we select any one literal that satisfies that clause in the satisfying valuation. If x_i satisfies the clause then T_i occurs at time $t+1$ and parasite node C_j associates with edge (T_i, β_i) at time $t+2$ incurring three loss events on the path C_j, S_j, T_i. Next, parasite C_j incurs two duplication events where its child C'_j host switches to edge (C_j, C'_j) in H whereas its child C''_j associates with the node with the same label in H. The case that $\overline{x}_i$ satisfies the clause is analogous. Thus, there are two duplication events, one host switch event, and six loss events per clause. Therefore, the total cost of this solution is $18n + 9m$ and the answer to the GCRDP instance is "yes."

Conversely, assume that the answer to the GCRDP instance is "yes." Notice that no lineage of P can pass through a k-thorn gadget in H since this cost would exceed the maximum permitted cost K. Thus, since each y_i and z_i node in P has a child labeled a_i, and this child must be associated with the a_i tip in H, we are forced to associate both y_i and z_i in P with an edge or node of the x_i variable gadget in H. However, both y_i and z_i also each have a b_i child that must be associated with the b_i tip in H. This requires that the a_i and b_i children of both y_i and z_i arise from duplication followed by host switching onto the edge (x_i, b_i) in H. Since duplication and host switching can only occur on edges and not on nodes, this further implies that y_i and z_i are associated with edges in the x_i variable gadget. Since y_i is fixed to occur at time $t+2$ and z_i is fixed to occur at time $t+1$, one of T_i or F_i must occur at time $t+1$ and the other at time $t+2$ in order for the host switches to be possible. We construct a valuation for the Boolean variables such that x_i is TRUE iff the node T_i in the x_i variable gadget is at time $t+1$.

Each node C_j in P has two children, C'_j and C''_j. Since the parasite lineage C_j cannot pass through the k-thorn gadget on the edge (C_j, C'_j) in H, the parasite node C_j must be associated with an edge in a variable gadget from which there is a path to C''_j. By construction, there are three such variable gadgets, one for each variable occurring in clause C_j. Since node C_j occurs at time $t+2$ in P, the path from p' to C_j in P must be associated with a path in H that enters one of these variable gadgets at time $t+1$. However, this implies that the variable corresponding to that gadget has a value that satisfies clause C_j, implying that the 3-SAT instance is satisfied by the valuation and is therefore satisfiable. ■

Finally, we consider the case of Pareto optimality. Consider an instance of the cophylogeny reconstruction problem. For any feasible solution for this instance, the *event vector* is a vector (x_1, x_2, x_3, x_4) where x_1, x_2, x_3, and x_4 denote the number of codivergence, duplication, host switching, and loss events, respectively, incurred in that solution. A solution with event vector $\mathbf{x} = (x_1, x_2, x_3, x_4)$ is said to be *Pareto optimal* if, for every feasible solution with corresponding cost vector (y_1, y_2, y_3, y_4), if $y_i < x_i$ for some $i \in \{1, 2, 3, 4\}$ then $y_j > x_j$ for some some $j \in \{1, 2, 3, 4\}$. In other words a solution with event vector $\mathbf{x}$ is Pareto optimal if there is no solution whose event vector is "strictly better" than $\mathbf{x}$. The set of event vectors of Pareto optimal solutions is called the *Pareto front* for the problem instance.

The Pareto optimization problem is that of finding the Pareto front for a given problem instance. The corresponding Pareto decision problem is as follows: Given a host network H, parasite network P, time functions t_H and t_P, mapping φ, and an integer vector $\mathbf{x} = (x_1, x_2, x_3, x_4)$, is $\mathbf{x}$ on the Pareto front?

It is easily shown that the Pareto decision problem is NP-hard via a Cook Reduction (polynomial-time Turing Reduction) (Garey and Johnson, 1979) from GCRDP to to the Pareto decision problem. The reduction is based on two observations: First, for a given problem instance, the number of distinct possible event vectors is polynomial in the size of the problem instance since the number of codivergences, duplications, host switches, and losses are each upper-bounded by the number of nodes and edges in the host and parasite networks. Thus, we can use an oracle for the Pareto decision problem to find the entire Pareto front in polynomial time. Second, a solution of minimum total cost with respect to the cost vector κ in the GCRDP instance must be a Pareto optimal solution, for otherwise there is a solution of even lower total cost

with respect to κ. Thus, we can first use the oracle for the the Pareto decision problem to find the Pareto front. Then, for each vector (x_1, x_2, x_3, x_4) in the Pareto front we compute $x_1\kappa_C + x_2\kappa_D + x_3\kappa_S + x_4\kappa_L$. The answer to the GCRDP instance with target cost K is "yes" if and only if at least one of these sums of products is less than or equal to K. Since this is a polynomial-time reduction, the proof is complete.

5. HEURISTICS

Since the reticulate cophylogeny problem is, in general, NP-complete, good heuristics or approximation algorithms are needed. In this section we show that the the dynamic programming algorithm described in Subsection 3.1 for the "fixed time" case can be used as the basis for meta-heuristics for this problem. We restrict our attention here to the case that the host network may be reticulate but the parasite network is a tree. The case that both networks are reticulate is evidently considerably more complicated, as explained later in this section.

Consider a host network H and a parasite tree P. For simplicity, assume that there are no time constraints on the event times in H and P, so that each node can occur at any time. (The general case of arbitrary time constraints is a trivial extension.) Recall that the internal nodes of H correspond to events. Consider a total ordering of the event times in H, that is, a permutation of internal nodes of H such that if a node u is ancestral to v then u must occur before v in the permutation. We henceforth refer to such a permutation as a *valid ordering* for H.

We can use the dynamic programming algorithm in Subsection 3.1 to find the optimal solution to the cophylogeny problem under a particular valid ordering. Different valid orderings will give potentially different solutions to the cophylogeny problem. However, since the dynamic programming algorithm finds an optimal solution for a particular valid ordering, an optimal solution for the general problem can be found by enumerating all possible valid orderings of the event times in H, applying the dynamic programming algorithm to each one, and selecting a solution of minimum cost. Since there are, in general, an exponential number of host event orderings, this algorithm is exponential time in the worst case. However, we can exploit this idea for efficient and effective heuristics.

For a given valid ordering of the host network events, define the *neighbor set* of a valid ordering to be the set of all valid orderings derived by inverting the order of two consecutive events that are not ancestrally related. That is, if u comes immediately before v in the ordering and u is not the parent of v, then the order of events u and v can be inverted, resulting in another a valid ordering. Note that the size of the neighbor set is bounded by $O(n)$ where n is the number of nodes in the host network.

A meta-heuristic (e.g., gradient descent, simulated annealing, or great deluge) can now be applied as follows: Begin by selecting a random valid ordering for H. Find the optimal solution for this ordering by applying the dynamic programming algorithm. Select another valid ordering from the neighbor set subject to the rules of the meta-heuristic. Repeat this process until the termination condition of the meta-heuristic is reached.

For example, for gradient descent, we begin with an arbitrary ordering and apply the dynamic programming algorithm to find the cost of the solution under this ordering. Next, we apply the dynamic programming algorithm to the host tree under each of these orderings. We choose the ordering that provides the largest reduction in cost and repeat this process until we reach an ordering such that all neighboring orderings do not further reduce the cost.

This meta-heuristic approach has the desirable property that the diameter of the solution space is bounded by $O(n^2)$ where n is the number of internal nodes in H. Thus, in theory, an optimal ordering can be reached from any random ordering in $O(n^2)$ iterations of the meta-heuristic.

Since, to the best of our knowledge, this is the first known heuristic approach to the reticulate cophylogeny problem, we have provided a portable and documented implementation for research purposes at *www.cs.hmc.edu/∼hadas/cophylogeny*.

Finally, we note that this heuristic depends on the parasite network being a tree, since the underlying dynamic programming algorithm makes this requirement. The dynamic programming algorithm does not appear to extend to the case that the parasite network is reticulate and thus different heuristics are needed for the case that both the host and parasite networks are reticulate.

6. CONCLUSION

In this paper, we have examined the computational complexity of the cophylogeny reconstruction problem. For the case that the host phylogeny may be reticulate, we have shown a sharp complexity boundary. In particular, we have shown that the GCRP is polynomial-time solvable when the relative times of host events are fixed but is NP-complete when they are allowed to take one of two values, even if the parasite phylogeny is a tree. Note that since trees are special types of phylogenetic networks, the more general case that both phylogenies may be reticulate is also NP-complete. As a consequence, the Pareto optimization version of this problem is NP-hard. We have also proposed a meta-heuristic approach for the case that the host network is reticulate but the parasite network is a tree.

There are a number of interesting directions for future research. First, it is widely-conjectured that the reconstruction problem is NP-complete when both the host and parasite networks are trees, but this problem still remains open. Moreover, while our results imply that the problem is NP-complete for the general case that the host and parasite trees are reticulate, we conjecture that when both networks are reticulate the problem is NP-complete under even more stringent conditions than those in Theorem 2.

Finally, the development and analysis of heuristics for this problem is an area of practical interest. While we have proposed one family of heuristics, there may be a number of other fruitful approaches. For example, if host switching is disallowed, the cophylogeny problem is easily solvable in polynomial time for arbitrary host and tree networks. Thus, one approach to developing good heuristics, or even approximation algorithms, for this problem may be in bounding the number of host switch events that are considered.

ACKNOWLEDGMENTS

This work was conducted at, and partially supported by, the School of Information Technology at the University of Sydney, Australia. Additional support was provided by the Mellon Foundation.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Bordewich, M., and Semple, C. (2007). Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete Applied Mathematics* 155, 914–928.

Brockelman, W.Y., and Gittins, S.P. 1984. Natural hybridization in the *Hylobates lar* species group: implications for speciation in gibbons. In: Preushcoft, H., Chivers, D.J., Brockelman, W.Y., et al., eds. *The Lesser Apes. Evolutionary and Behavioral Biology*. Edinburgh University Press, Edinburgh.

Charleston, M. 1998. Jungles: a new solution to the hostparasite phylogeny reconciliation problem. *Math. Biosci.* 149, 191–223.

Charleston, M. 2002. Principles of cophylogeny maps. In Lässig, M., and Valleriani, A., eds. *Biological Evolution and Statistical Physics*. Springer-Verlag, Berlin.

Charleston, M., and Page, R.D.M. n.d. TreeMap. Available at *www.it.usyd.edu.au/∼mcharles/software/treemap/treemap.html*. Accessed October 15, 2008.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco.

Harrison, R.G., and Rand, D.M. 1989. Speciation and its consequences, 111–133. In Otte, D., and Endler, J.A., eds. *Mosaic Hybrid Zones and the Nature of Species Boundaries*. Sinauer, Sunderland, MA.

Ho, S.Y.W., Phillips, M.J., Cooper, A., et al. 2005. Time dependency of molecular rate estimates and systematic overestimation of recent divergence times. *Mol. Biol. Evol.* 22, 1561–1568.

Ma, B., Li, M., and Zhang, L. 2000. From gene trees to species trees. *SIAM J. Comput.* 30, 729–752.

Marshall, J.T., and Brockelman, W.Y. 1986. Pelage of hybrid gibbons (*Hylobates lar* x *H. pileatus*) observed in Khao Yai National Park, Thailand. *Nat. Hist. Bull. Siam Soc.* 34, 145–147.

Merkle, D., and Middendorf, M. 2005. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory Biosci.* 123, 277–299.

Merkle, D., and Middendorf, M. n.d. Tarzan. Available at: *informatik.uni-leipzig.de/pv/Software/Tarzan/PV-Tarzan.engl.html*. Accessed October 15, 2008.

Page, R.D.M. 1994. Parallel phylogenies: reconstructing the history of host-parasite assemblages. *Cladistics* 10, 155–173.

Ronquist, F. 1995. Reconstructing the history of host-parasite associations using generalized parsimony. *Cladistics* 11, 73–89.

Ronquist, F. 1998. Three-dimensional cost matrix optimisation and maximum cospeciation. *Cladistics* 14, 167–172.

Ronquist, F. n.d. TreeFitter. Available at *http://www.ebc.uu.se/systzoo/research/treefitter/treefitter.html*. Accessed October 15, 2008.

Than, C., Ruths, D., and Nakhleh, L. 2008. Phylonet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics* 9, 322.

Xu, S. 2000. Phylogenetic analysis under reticulate evolution. *Mol. Biol. Evol.* 17, 897–907.

Zhang, L. 1997. On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *J. Comput. Biol.* 4, 177–187.

Address reprint requests to:
Dr. Ran Libeskind-Hadas
Department of Computer Science
Harvey Mudd College
301 Platt Blvd.
Claremont, CA 91711

E-mail: hadas@cs.hmc.edu