Tackling the Intractability of the Cophylogeny Reconstruction Problem

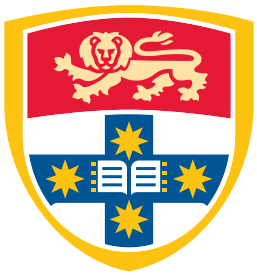
BIN ZHOU
SID: 308207211

Supervisor: Associate Professor Michael Charleston

This thesis is submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Science (Advanced) (Honours)

School of Information Technologies
The University of Sydney
Australia

4 November 2011



THE UNIVERSITY OF SYDNEY

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Bin Zhou

Signature: **Date**:

Abstract

Understanding the interactions between interdependent evolutionary systems is important in explaining the current state of the systems and predicting how they will continue to evolve. Cophylogeny reconstruction is the computational problem of accurately identifying these interactions, and is the central task of the emerging discipline of *cophylogeny*. Recent research has shown that finding the optimal answer to this problem is computationally intractable. As a result, current developed methods focus on fast heuristic and metaheuristic approaches. There is little focus on examining the quality of solutions from these methods – some sacrifice the optimality of the solution without guarantee whilst others will admit infeasible solutions.

This thesis addresses the gaps in current research in cophylogeny reconstruction methods. We provide theoretical results on the assumptions framing the problem and mathematically define the properties that are responsible for the intractable nature of the problem. With these results, we formulate the cophylogeny reconstruction problem as an Integer Linear Program. This formulation is more compact than previous attempts and provides a concrete method for optimally solving the reconstruction problem.

We develop an implementation of our Integer Linear Program on a commercially available optimisation solver platform. Our experiments show that the implementation can solve real-world problems optimally within a reasonable amount of time on inexpensive commodity hardware. Our implementation provides the first alternative to the current generation of unguaranteed heuristic approaches to the cophylogeny reconstruction problem. Our cophylogeny reconstructions are guaranteed to be optimal, which dramatically improves the fundamental data used by fields as diverse as virology, linguistics, and anthropology.

Acknowledgements

I would like to thank my supervisor, Michael Charleston, for his guidance throughout this project. Thank you, Mike, for being forever tolerant of my unannounced visits to your office as I hurriedly sketched out the flaws of my derivations from previous visits and assured you of the merits of my newest “proof”.

I would also like to thank my close friends Sophie Liang, Dominick Ng, David Rizzuto and George Karpenkov for their constant encouragement and feedback. I could not have survived the year without the motivation and support they all selflessly gifted.

Finally, I would like to thank my family – my parents Weiling and Paul, and my sister Jess – to whom I dedicate this thesis. In everything that I do, they are forever and unquestioningly understanding and supportive. For that, I am grateful.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

Evolution is a process that can model the underlying mechanism governing the temporal behaviour of systems critical to a diverse range of research disciplines including biology, geography, linguistics, and anthropology. However, evolution is rarely an independent or isolated process – for example, the evolution of a virus would be highly influenced by the evolution of its host. Therefore, it is natural to study not only the evolutionary mechanisms of a system within itself, but also the dynamic relations and dependencies between multiple evolving systems. The field of *cophylogenetics* aims to model the interactions between evolutionary processes through time in order to reveal the underlying dependencies. This differs from the classic biological study of phylogenetics, which seeks to study the history of events (phylogeny) which best explains the evolution of single clades (species descendant from a common ancestor). Rather, cophylogenetics focuses on studying the history of interactions (cophylogeny) between two or more evolutionary systems.

The study of cophylogenetics is of interest to a wide range of biology disciplines. Systematics often studies the association between evolving organisms and their genes; biogeography studies the associations between evolving geography and species; and parasitology studies the association between parasite and host phylogenies. Understanding cophylogeny is vital in a wide variety of applications, including identifying the origins of an invading species, comparing the evolution rates and virulence of diseases, and predicting the lateral transmission of animal-borne diseases into humans.

Whilst the analysis of evolutionary systems is traditionally biologically motivated, cophylogenetic techniques can be applied to any system modelled in terms of an evolutionary process. Several recent studies have used the idea of modelling culture and language in terms of evolution [1, 17, 20] to study the associations between language and geography, even explicitly using cophylogenetic techniques developed within a biological context to study cultural/geographical co-evolution [52].

The central task in cophylogenetic research is *cophylogeny reconstruction* – the task of finding the historic associations between two evolving systems that are the most likely explanation for the observed contemporary associations. Given two evolutionary systems, we can determine the interactions within the current state of the systems by direct observation or experimentation. However, a single snapshot of associations is rarely enough to model how the interactions change with time. In most systems evolution is a slow mechanism. This rules out the possibility of waiting for the state of the system to change to take further snapshots. Instead, the history of association between the systems produced by cophylogeny reconstruction can be used as a basis to model the system dependencies.

For all but the most trivial examples, the number of possible cophylogeny reconstructions is intractably large, hence computational approaches are used in order to find the better (and hopefully, the best) reconstructions.

A number of computational techniques have been described to approach this task, but few are guaranteed to find the best solution. The proven intractability of the problem means there is little hope for finding an efficient *and* optimal approach. Most approaches instead offer solutions that have been demonstrated to be empirically "good" through comparison against canonical problem instances with known best solutions. However, the authors of these heuristic methods do not provide mathematical arguments that bound how close these "good" solutions are from the optimal given arbitrary problem instances.

The lack of bounds or guarantees presents several problems in current research. The solution space of reconstructions is not smooth – small changes in objective value can lead to vastly different solutions. Thus "good" heuristic reconstructions may be misleading and conclusions drawn from such solutions may be fallacious. New heuristic methods are often benchmarked against other heuristic methods. There does not currently exist a practical exact method that heuristic solutions can be compared against. Whilst benchmarks between heuristics can identify which are *better*, the quality of reconstruction methods remain a relative measure without exact solutions for comparison. A second gap in current literature is a full mathematical description of the reconstruction problem. There exists work mathematically detailing some aspects of cophylogeny reconstruction [7], but some more troublesome aspects – such as temporal constraints in feasible reconstructions – never receive such treatment. A mathematical foundation is important for formulating practical exact solutions and is the fundamental basis for proving bounds on the quality of solutions found by non-exact solutions.

1.1 Contributions

The work presented in this thesis aims to address gaps in current cophylogeny research, providing both a mathematical foundation for the cophylogeny reconstruction problem as well as a practical exact method for this task.

1.1.1 Study of Temporal Compatibility

We present an analytical study of the temporal compatibility constraints of cophylogeny reconstruction. Widely identified as the main roadblock in tractability, temporal compatibility is an understudied but vital aspect of the reconstruction problem. Temporal compatibility refers the requirement that the reconstructed associations between a host-parasite system does not induce circular, and thus contradictory, time lines. Without such restriction, the cophylogeny reconstruction problem is tractable – a polynomial time exact solution using dynamic programming techniques has been proposed to solve such a problem [30].

In our study, we mathematically frame the notion of temporal compatibility. We focus on the most difficult part of temporal compatibility – host switch timings. We show that conditions stated in current literature are insufficient and demonstrate the exact conditions actually required to maintain temporally feasible sets of host switches in a reconstruction.

Using this exact formulation, we prove the correctness of an existing common method of detecting temporal incompatibilities – maintaining a partial ordering on a set (*poset*) of host-parasite pairs known as *j-vertices*. We argue that this is an impractically large poset to maintain and propose an alternate method requiring an asymptotically smaller poset.

1.1.2 Practical ILP Formalisation

We propose a new ILP formulation of the cophylogeny reconstruction problem and show the correctness of the formulation. A very early attempt at an exact solution to the cophylogeny reconstruction problem exists – the "Jungle method" of [6] – but was far too inefficient for real-life datasets. Recently, Libeskind-Hadas and Charleston [28] proposed an ILP formulation, but this was also too inefficient for practical use.

Our formulation builds on some novel ideas presented in Libeskind-Hadas and Charleston's ILP, but also uses common ideas found in heuristic methods. Finally, it incorporates the temporal compatibility constraints proposed in this thesis.

We produce a reference implementation of our proposed ILP. Using this implementation, we benchmark our solution to a comparable implementation of the Libeskind-Hadas and Charleston [28] ILP as well as the recent heuristic solution JANE 3 [2]. The results show that whilst our ILP is far slower than JANE 3 (as expected from an exact solution to an NP-hard problem), it was sufficiently efficient to solve real-life sized datasets on readily available commodity hardware. The solutions from the ILP are also guaranteed to be exact, unlike the heuristic solutions of JANE 3. The benchmarks also demonstrated that the new ILP far outpaced the previous formulation. The new formulation can solve problem sizes of up to 25 host - 40 parasite within reasonable time (a few hours) compared to the previous formulation which can only solve problem sizes of up to 5 host - 5 parasite within the same time.

To demonstrate its practicality, we applied our new solution to the complete dataset from a recent study on butterfly mimicry [23].

We conclude that our new formulation is a practical exact method for solving the cophylogeny reconstruction problem. The implementation will also benefit from any developments in the well established research area of linear optimisation. As a purely mathematical representation, with further work, our ILP could also form the basis for approximation algorithms for cophylogeny reconstruction.

1.2 Outline

Chapter 2 describes the cophylogeny reconstruction problem in detail, formalising the constraints and objectives of the problem as well as discussing some common problem variations. Chapter 3 reviews previous work in cophylogeny reconstruction, including the results that led to current approaches in research.

Our detailed analytical study into temporal compatibility in cophylogeny reconstruction is in Chapter 4. Building on this work, our proposed ILP formulation is discussed in Chapter 5. Our reference implementation and benchmarks against existing methods are discussed in Chapter 6. Chapter 7 summarises the results of the thesis and explores the impact of our work in the field of cophylogeny.

CHAPTER 2

The Cophylogeny Reconstruction Problem

In this chapter, we formally define the cophylogeny reconstruction problem and enumerate the conditions that constitute a feasible solution. We discuss the common methods of evaluating the quality of a reconstruction. Finally, we discuss the effect of changing some assumptions made in the reconstruction problem. Chapter 3 summarise the literature in cophylogeny that led to these current definitions and characterisations of the cophylogeny reconstruction problem.

2.1 General Definitions

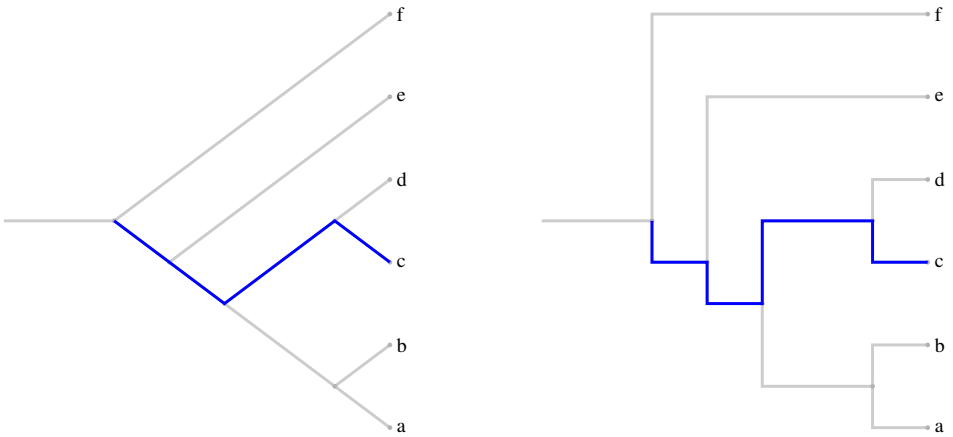
2.1.1 Phylogeny Trees

An evolving system can be described by a group of *species* arranged in a phylogeny tree that represents the ancestor/descendent relationship between these species.

Formally, a phylogeny tree is a rooted full binary tree (Fig. 2.1). The edges of the tree represent the species of the system and the nodes represent speciation events – when a species evolves two populations that are sufficiently distinct to warrant consideration as two new species. Thus the edge is the *lifetime* of a species, from when it was considered a new species as the result of the evolution of its parent species, until its own speciation event. In a tree there is a one-to-one correspondence between nodes and edges, so we can also associate each species (edge) with its corresponding speciation event (node). Thus, when we refer to a species of a phylogeny tree, we may be referring to either the corresponding node or edge of the tree, or both.

The ancestor/descendent structure of the tree directly represents the ancestor/descendent relationship of the species in the system. There is then a natural time dimension in a phylogeny tree – the root being the earliest time whilst the leaves representing the most recent time. Generally, we consider all leaves to be

contemporaneous and current. The root of the tree is augmented with an edge to represent the common ancestor of all the species in the system.



(A) Phylogeny Tree – angled edge representation

(B) Phylogeny Tree – orthogonal edge representation

FIGURE 2.1. Two representations of the same phylogeny tree. A single lineage highlighted.

A *lineage* is a sequence of species related by ancestry. In a phylogeny tree, a lineage is synonymous to a branch or sub-branch of the tree (see Fig. 2.1, highlighted).

Although termed biologically, evolving systems are not limited to ecological species – an equally viable notion of species is landmasses that 'speciate' by drifting; languages, cultures, and genes also evolve and can be grouped into phylogenies (see Table 2.1).

2.1.2 Host-Parasite Associations and Dependence

An evolving system P is described as being dependent on another system H if the evolution of species in H is a selective pressure on the species in P. A host-parasite system is such a pair of dependent evolving systems. Whilst it is often the case that the two systems will exert selective pressures on each other [55], we consider the group of species exerting more pressure to be the hosts. In fact, the parasites often depend on the host for survival, but the host can survive quite independently of the parasites.

Parasite	Hosts
Pollinating Insects	Flowering Plant
Viruses	Viral Hosts
Genes	Organisms
Animal Species	Landmasses
Culture and Customs	Societies and Tribes
Languages	Civilisations

TABLE 2.1. Example Host-Parasite systems from a diverse range of disciplines

A parasite species $p \in P$ is *associated with* or *related to* a host species $h \in H$ if it is h that exerts selective pressure on p. We say that parasite p infests or infects host h (thus depends on h for survival). If a parasite is associated with more than a single host at any given time, we say the parasite is *widespread*.

In the basic cophylogeny reconstruction problem, we do not consider widespread parasites (see Section 2.3.1).

2.1.3 Tanglegrams in Cophylogeny Reconstruction

A *tanglegram* is a pair of trees and a set of associations between the leaves of the trees [34]. In cophylogeny reconstruction, a tanglegram is the input to the problem. The two trees form the host-parasite system and the leaf associations are the relations between current parasites and hosts, obtained by direct observation or experimentation (Fig. 2.2).

If we do not consider widespread parasites, then we require that the leaf mapping be a surjection from the parasite system to the host system. If the set of associations is not a map, then we must have a leaf parasite infecting multiple leaf hosts – that is the parasite must be widespread. If the map is not surjective, then we can trim the host tree of any unmapped subtrees and obtain a surjective leaf map.

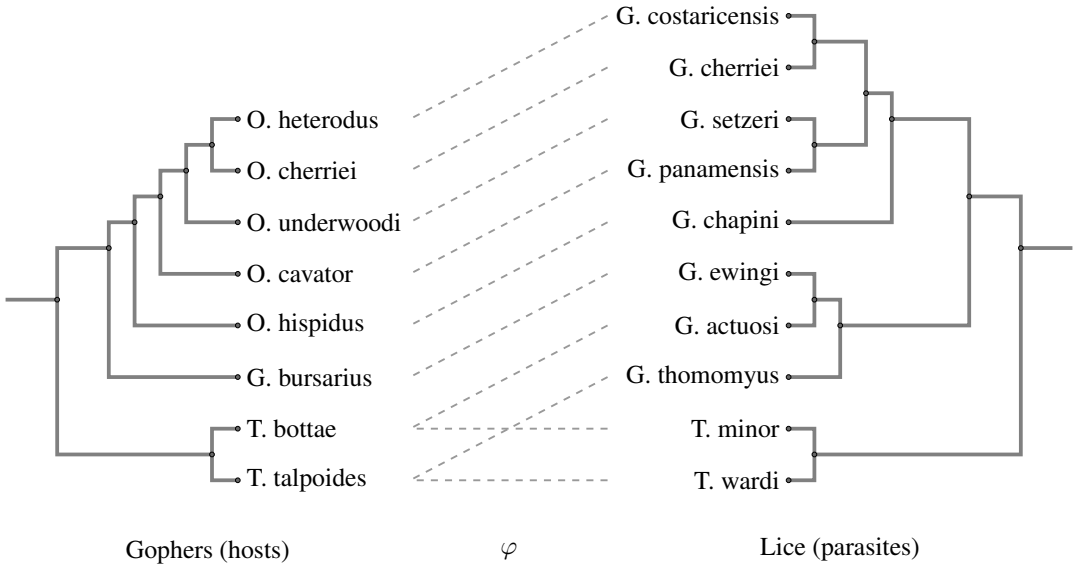


FIGURE 2.2. An example tanglegram showing the association between current gopher species and chewing lice, as well as the historic phylogenies of the hosts and parasites.

2.2 Problem Definition

COPHYLOGENY RECONSTRUCTION PROBLEM. Given two phylogeny trees P and H, and a surjective map, φ, from the leaf species of P to the leaf species of H, construct the minimal cost feasible set of associations, Φ, between P and H that is consistent with φ.

Intuitively, we are given some parasite-host system and a set of associations between the contemporary host and parasite species. We want to determine the most likely set of associations between the historic parasite and host species that lead to the observed associations (φ).

It is important to note that in many current heuristic methods, the reconstruction output is a set of associations of parasite speciation events with hosts – that is, how nodes of the parasite tree map to edges or nodes of the host tree. However, we do not load this extra semantics on how we define "associations". Hence, throughout this thesis, when we say a parasite p is associated with a host h in a reconstruction, we do not necessarily mean that parasite p's speciation event occurred on host h – or imply, in the language of mapping methods, that the node of p in the parasite phylogeny is mapped to the edge or node of h in the host phylogeny. Our definition of association is simply that p has infected h at some stage, consistent with the general definition in Section 2.1.2.

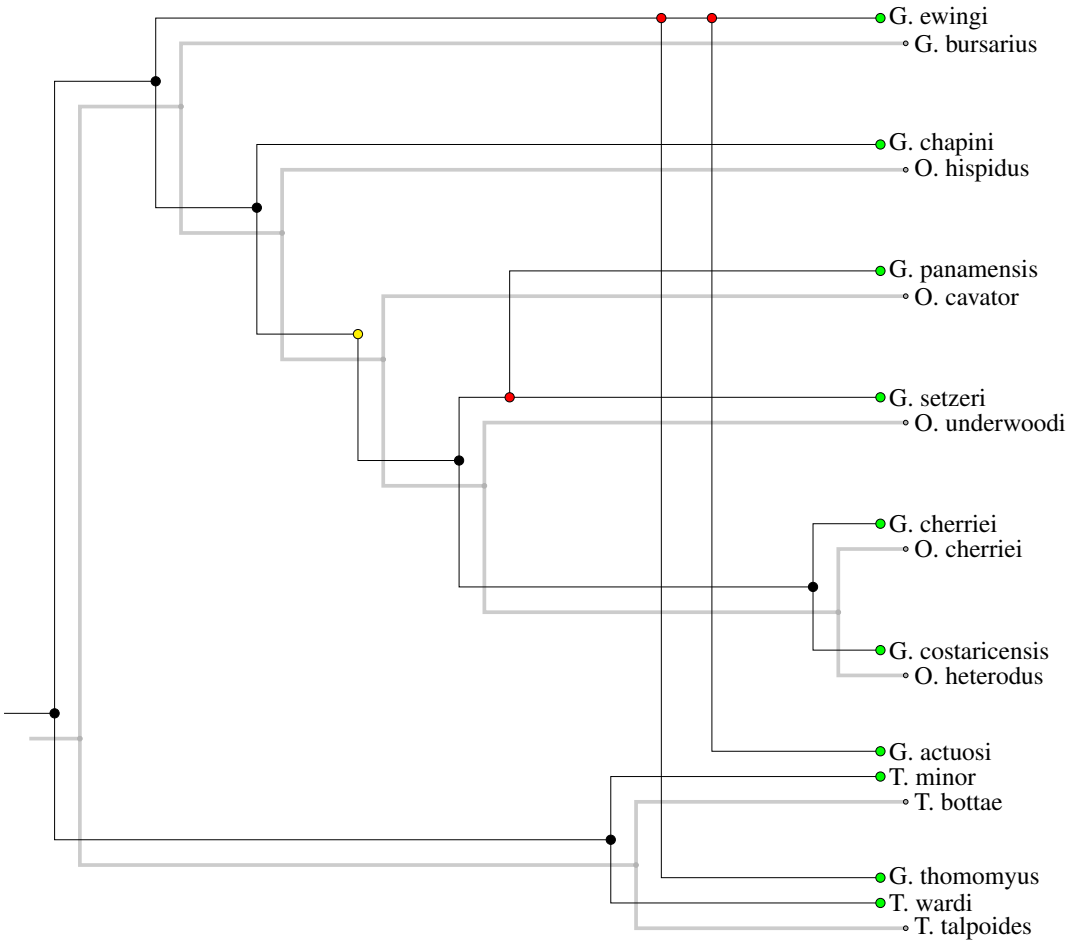


FIGURE 2.3. Possible reconstruction for tanglegram in Fig. 2.2. lighter: host tree; darker: parasite tree.

2.2.1 Events

When a parasite is associated with a host, we expect the parasite to continue to be associated with the host until some *event* occurs to change these associations. Hence, an *event* is a valid or feasible transformation of associations. In a reconstruction, we require a set of associations that are consistent under some set of feasible events that ultimately lead to the observed leaf associations. By costing events inversely proportional to the likelihood of the event occurring, finding the *most likely* set of historic associations is equivalent to the *minimal cost* sequence of events implied by a feasible set of associations.

There are six basic events commonly used in host-parasite models [48]:

- Cospeciation (or generally, codivergence);
- Sorting (or loss);
- Extinction;
- Duplication;
- Host switch (or colonisation, or lateral transfer); and
- Full host switch.

An *extinction event* occurs when a parasite lineage ceases to exist entirely. Thus a parasite associated with a host will terminate its association with the host at an extinction event and is not associated with any host from that time onwards. A *full host switch* occurs when a parasite associated with one host re-associates with a non-descendent host without speciating. Both *extinction* and *full host switch* are considered untraceable (see Section 2.2.2.3) in cophylogeny reconstruction, so only four events are considered recoverable [8]. Any transformation of associations that does not fall into the definition of these four events are considered infeasible.

2.2.1.1 Cospeciation

Cospeciation (or codivergence) occurs when a parasite and its associated host both speciate. More formally:

DEFINITION 2.1 (Cospeciation). A *cospeciation event* occurs when a parasite p is associated with a host h and, without loss of generality, the children of p, p_1 and p_2, are associated with the children of h, h_1 and h_2, respectively.

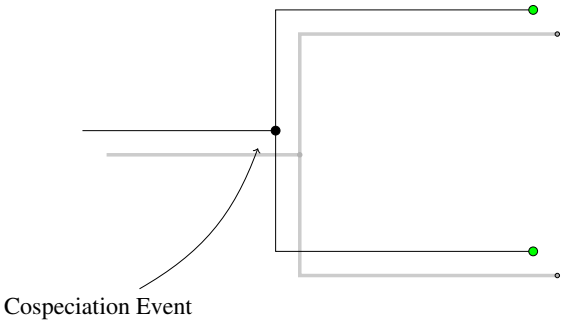


FIGURE 2.4. A reconstruction showing a cospeciation event. lighter: host tree; darker: parasite tree.

Cospeciation does not require both parasite and host speciations to occur simultaneously. In fact, in most systems, speciations are not instantaneous events, but take some finite amount time. We only require that the speciation of parasite and host happen sufficiently close together as to be indistinguishable from simultaneous events [8].

Cospeciation does require that both host children are mapped to by the two parasite children. If both parasite children map to only one of the host children, we require the event to be interpreted as a *duplication* instead.

In many models, cospeciation is given a negative or zero cost [6, 39, 47]. This is based on the "maximising hypotheses of codivergence" [38] (see Section 3.1.2). By assigning a non-positive cost to cospeciation, minimal cost reconstructions will focus on maximising cospeciation events.

2.2.1.2 Loss

Also known as lineage sorting or "missing the boat" [42], a *loss event* occurs when a host speciates but an associated parasite does not. More formally:

DEFINITION 2.2 (Loss). A *loss event* occurs when a parasite p is associated with both a host h and a single child of h, h_1.

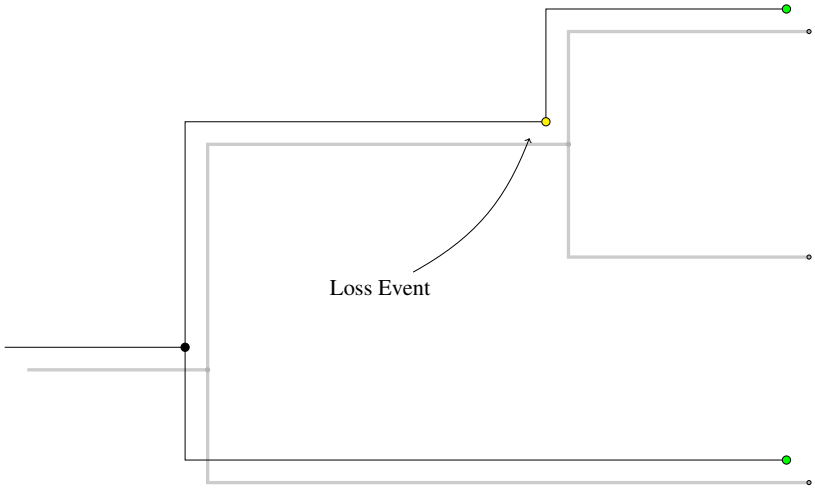


FIGURE 2.5. A reconstruction showing a loss event. lighter: host tree; darker: parasite tree.

The loss event represents the situation where a parasite fails to speciate and track both child species after the associated host speciates. Failure to sample a parasite lineage offers an alternate explanation to loss events – the parasite did speciate to track both children of the speciating host (so cospeciation occurred), but one of the children parasite lineages was not sampled or became extinct. All of these options would lead to the same observed event, so in cophylogeny reconstruction, we do not distinguish between them [8].

Although sampling errors do occur, it is not the only cause of loss events – "missing the boat" events have been shown to be realistic [42]. Hence we still associate positive cost to a *loss event* as loss is less desirable than cospeciation.

2.2.1.3 Duplication

A *duplication event* occurs when a parasite is associated with a host and speciates independently of the host. More formally:

DEFINITION 2.3 (Duplication). A *duplication event* occurs when a parasite p and a child of p, p_1, are both associated with a single host h.

The presence of both a parasite and its child on a single host suggests that the parasite must have speciated and the host did not.

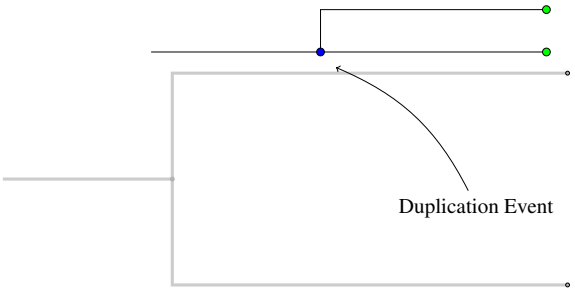


FIGURE 2.6. A reconstruction showing a duplication event. lighter: host tree; darker: parasite tree.

We do not require both children to remain associated with the host – a second event, *host switch*, may follow allowing one of the children to be associated with an unrelated host (see Section 2.2.1.4).

In a host-parasite system, the host asserts strong selective pressures on the parasite. Parasite speciation events independent of the host are considered uncommon events. Thus we consider duplication less likely than cospeciation, and so a positive cost is associated with duplication events in cophylogeny reconstruction.

2.2.1.4 Host Switch

A host switch, or more generally "lateral transfer", is an event that follows a duplication in which one of the child parasites resulting from the duplication colonises an unrelated host. More formally:

DEFINITION 2.4 (Host Switch). A *host switch event* occurs when a parasite p associated with host h duplicates, but, without loss of generality, only one child of p, p_1, is associated with h, whilst the other child, p_2, is associated with a contemporaneous host h_2 not of the same lineage (branch) as h.

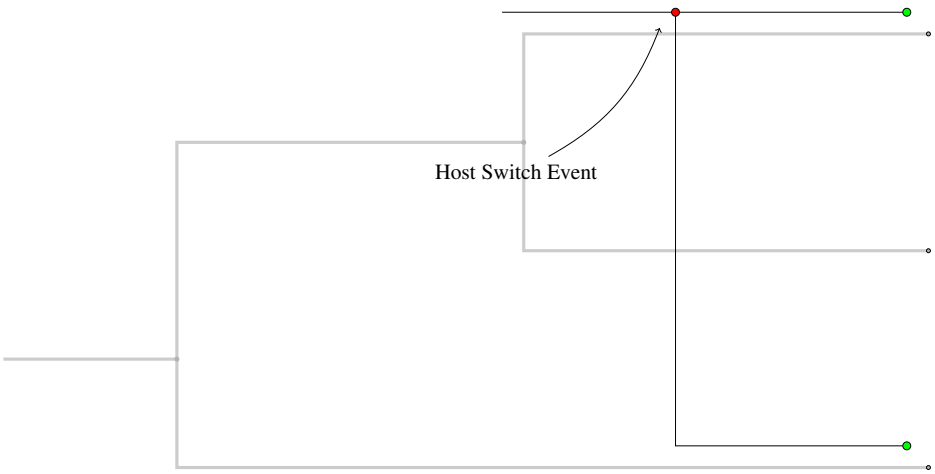


FIGURE 2.7. A reconstruction showing a host switch event. lighter: host tree; darker: parasite tree.

In the basic model of cophylogeny reconstruction we considered, a host switch is the only event that allows a parasite lineage to stop tracking one host lineage and start tracking a second host lineage. However, the traceable condition (see Section 2.2.2.3) requires that one child of any duplicating parasite remain on the original host lineage.

Host switches infer complex time information on a reconstruction and multiple host switches can be incompatible within a single reconstruction. What precisely constitutes a feasible set of host switches is an often ignored aspect of cophylogeny reconstruction. An in-depth study of the host switch event and compatibility issues are provided in Chapter 4.

2.2.1.5 Cost Schemes

Each evolving system has unique characteristics, so it would be incorrect to find one "global" cost assignment for each of the allowable events. Even between ecological host-parasite relations, there is no "best consensus likelihood" of some event occurring. One of the major difficulties with cophylogeny reconstruction is choosing the "correct" cost scheme that most closely models the relative probabilities of the occurrence of each event.

Any method of cophylogeny reconstruction then has to be "cost scheme agnostic" – that is, the method should work regardless of what the input cost scheme is.

However, there are some sensible restrictions on cost schemes that can be assumed [7, 48]:

- Cospeciation has the lowest cost (normally 0 or negative);
- All other events have larger or positive costs;

These cost scheme restrictions are not only heuristic, they are required to guarantee a desirable property of reconstructions known as the "isomorphism condition" [7] or "potent isomorphic property" [48].

2.2.1.6 Event Counting

There is also contention on how events should be counted in event based cophylogeny reconstruction. There are two main event count models used in modern reconstruction methods: "node based" model advocated by Merkle and Middendorf [30], Merkle et al. [31]; and "edge based" model used in various TREEMAP versions [11, 13, 40]. Newer versions of JANE are able to use both models [14, 16].

Node based models count one event for each node associated with the event. A loss is associated with a host speciation, and each of duplication, cospeciation, and host switch are associated with a parasite speciation.

Edge based models associate event counts with edges in the parasite phylogeny. Losses are counted in the same way as node based models. However, for each edge in the parasite phylogeny, we count the event that created the edge. A single duplication (or cospeciation) event as counted by node based models would result in two duplication (or cospeciation) counts in the edge based models. Edge based methods also consider a host switch separately to the associated duplication. Hence, a single host switch

in the node based model would be counted as two duplications and a host switch in the edge based model.

Method	Cospeciation	Duplication	Host Switch	Loss
Node based	6	3	3	1
Edge based	12	6	3	1

TABLE 2.2. Event counts for the reconstruction in Fig. 2.3 for different event counting methods.

The two models differ only in how cospeciation and duplications are counted – loss and host switch counts would remain the same. For binary phylogenies (assumed by the basic model of cophylogeny reconstruction), any edge based scheme can be transformed into an equivalent node based scheme by doubling the costs of duplication and cospeciation and adding the cost of a duplication to the cost of host switching. Similarly, any node based scheme can be transformed into an equivalent edge based scheme using the reverse process. Since the two schemes are interchangeable, the differences are mostly cosmetic. The only important case is that if the host switch cost in a node based scheme is lower than duplication cost, the cost of host switching in the equivalent edge based cost scheme would be negative. This may contradict the assumptions of methods which may rely on the host switch costs being positive.

However, if the cophylogeny reconstruction model is extended to take into account non-binary trees, then the edge based cost scheme can be more consistently extended to the generalisations of duplication and cospeciation. Concretely, a node based cost scheme applied to non-binary phylogenies will suggest that a duplication into two species is equally costly (and hence equally likely) to a duplication into three species, whilst an edge based cost scheme will cost the later more than a normal duplication.

2.2.2 Constraints

There are several constraints that are non-implicit in the mathematical problem of finding optimal reconstructions that we need to impose on what is considered a feasible reconstruction. These constraints are independent of the other assumptions we might make about the reconstruction model (such as assumptions about widespread parasites or feasible events).

Enumerations of these constraints or properties can be found in Charleston [7], Ronquist [48].

2.2.2.1 Temporal Consistency

The reconstruction must imply a time line of events that is consistent – the implied order of events cannot create cyclic time dependencies.

Evolution is a temporal process that moves forward with time, hence we require that time cannot move backward. There are several sources of timing information implicit to the input as well as to the reconstruction.

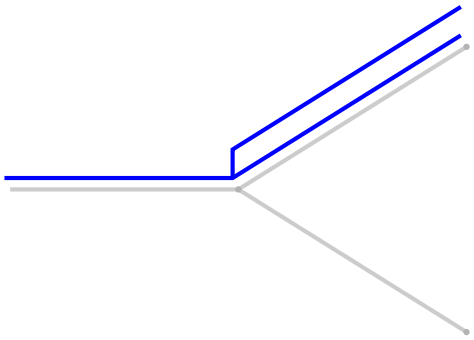
(1) The phylogeny of the hosts or parasites impose timing conditions on the speciation events – speciation of a parasite must occur after the speciation of any ancestor parasites and similarly for hosts.
(2) An association between a parasite and host implies that the parasite and host were contemporaneous.
(3) A host switch implies that the two hosts involved were contemporaneous.

The problem of temporal consistency has not been extensively studied in literature. A thorough study of the problem of parsimonious reconstruction is found in Ronquist [48] but temporal consistency is only briefly mentioned under the broad label of "consistency". We provide a detailed exploration of these constraints in Chapter 4.

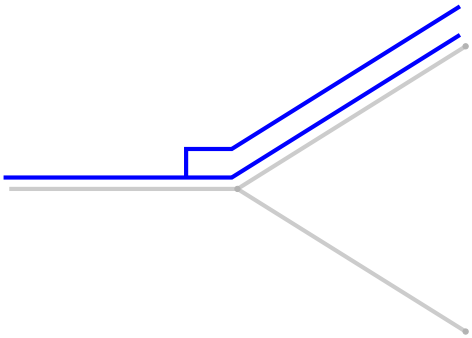
2.2.2.2 Resolvability

Whilst we consider a reconstruction a time line of events, the events themselves are rarely instantaneous. Evolution is generally a slow process and evolutionary events span over some time period.

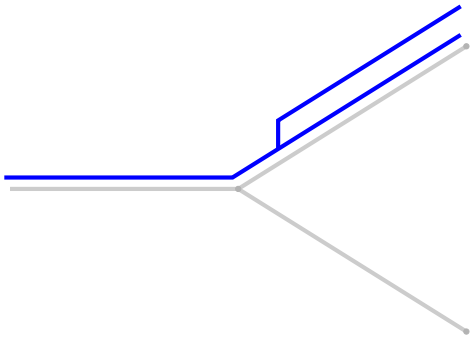
However, in cophylogeny reconstruction, we require that the timespan of the events are small enough in comparison to the timespan of the entire phylogenies that we can resolve each event as a point in some time line. This is not to suggest that the exact time of the event is required, or can be resolved, but that we need to be able to determine the order of events.



(A) Unresolved duplication.



(B) Duplication resolved before host speciation, yielding two loss events.



(C) Duplication resolved after host speciation, yielding one loss event.

FIGURE 2.8. Duplication concurrent with host speciation can have multiple interpretations with a different number of loss events.

In some cases, such as cospeciation and host switching after duplication, we require that the timespan of events are sufficiently small and well intersected that the events can be considered simultaneous. In other cases, we require a clear separation between events. Host switches and duplications cannot occur concurrently with a host speciation – that is, the host switch or duplication of a parasite cannot also be the cospeciation of the parasite. The ambiguous interpretations of concurrent host switching and host

speciation is discussed in [7]. Fig. 2.8 illustrates the multiple interpretations of concurrent duplication and host speciation events. Problematically, different loss events are inferred by each interpretation.

2.2.2.3 Traceability

Traceability, or *factuality* [48], is the requirement that any association in a reconstruction can be traced by descendence to a leaf association. That is, if p is associated with h, then there must be a branch from p to some leaf in the parasite phylogeny and a branch from h to some leaf in the host phylogeny such that the parasite branch associates only with the host branch.

Put more concisely, if a parasite p speciates on h, then at least one of its children must be associated with h or a child of h. Any event that satisfies this condition is *traceable*.

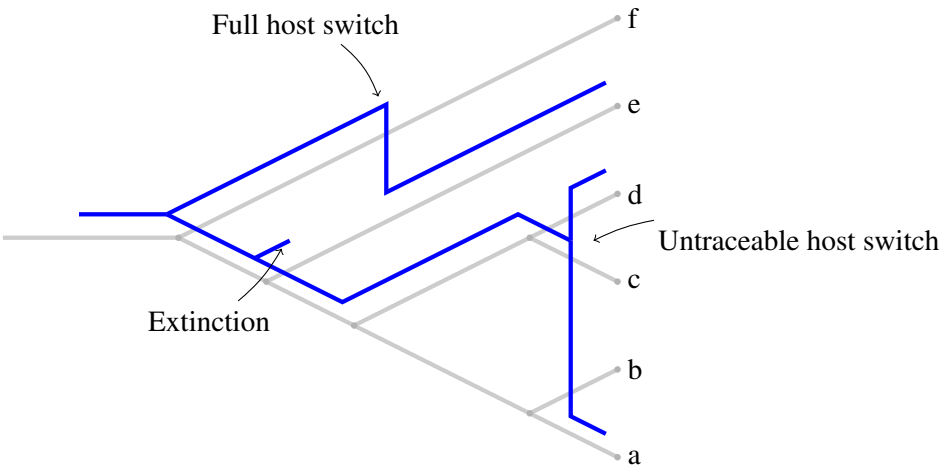


FIGURE 2.9. Example "ghost events" considered untraceable.

Some untraceable events are shown in Fig. 2.9. Events that are not traceable are *ghost events* [48]. Ghost events should not be postulated by a reconstruction as there is no evidence (in terms of the input leaf associations) to support its existence. Ghost events may support more parsimonious reconstructions that are less likely to have occurred – an undesirable trait when we require parsimony to estimate likelihood.

The main effect of requiring traceability is on host switching – a host switch can only occur after duplications; and only one child parasite can switch to a new host lineage. Traceability also prevents reconstructions from postulating extinctions, apart from those implied by loss/sorting events.

2.2.2.4 Continuity

We require that every parasite association is produced by an event. That is, if a parasite p is associated with a host h, then there was necessarily historic associations that allowed some valid event led p to be associated with h. The only exception is the first association of the root in the parasite phylogeny. In conjunction with the traceability condition, this condition ensures that a parasite must continuously track a host lineage until its speciation.

With the four event model, the continuity constraint implies that a parasite p can be associated with a host h that is on a different lineage from p's parent only through a host switching event. Hence, if a parasite p did not host switch or p did host switch but h is not p's earliest association, then:

- p must be associated with h's parent (loss event); or
- p's parent must be associated with h's parent (cospeciation event); or
- p's parent must be associated with h (duplication event).

In the case where p is the root of the parasite phylogeny, only the first option exists (as p has no parent).

2.3 Problem Variations

The cophylogeny reconstruction problem described in Section 2.2 and used throughout the rest of the thesis is a very basic version of the problem. Several strong assumptions are made to make the problem easier to solve and model mathematically. These assumptions are all based on good biological reasoning [48].

There are many variations to the cophylogeny reconstruction problem when assumptions are removed or added. The most common of these variations are discussed below.

2.3.1 Widespread Parasites

One of the basic assumptions made is that each parasite can only be associated with a single host at any given time. This commonly acknowledged assumption greatly simplifies the difficulty in defining parsimony in reconstructions [50]. However, it can be the case that a single parasite species has two populations associated with two separate host species. If the two populations are not sufficiently different, it would be incorrect to consider the two as separate species.

In cophylogeny reconstruction, it may be the case that the leaf associations between parasite and host phylogenies does not form a map, that is, the leaf parasites may be associated with more than one leaf host. One method for dealing with this case is to simply split the leaf parasite into as many new leaf parasites as required for the associations to form a mapping again [41]. However, this assumes that the widespread nature of the leaf parasite is a recent event and does not take into account the possibility of ancestral parasite being widespread as well. Also, it is unclear how these new parasite leaves should be arranged in place of the single parasite leaf in the original phylogeny – differing arrangements of the leaves may lead to different optimal reconstructions.

A second possible solution is to include further events to account for the possibility of widespread parasites, such as the *failure to diverge event* [9, 12]. A failure to diverge event occurs when a parasite is associated on a host that speciates, and the parasite continues to associate with both child hosts without speciating. However, such an event is severely under-defined. It is unclear how events would then continue to apply to these now potentially independent parasite populations. For instance, should the speciation of the parasite be independent amongst the two resulting populations? Furthermore, it is

postulated that the computational complexity of taking into account failure to diverge events would be far too great for current methods to incorporate [9].

2.3.2 Non-Independent Parasites

Whilst we assume that a single parasite can only be associated with a single host at any given time, the basic cophylogeny reconstruction problem allows multiple parasites to be associated with a single host. In doing so, we are implicitly assuming that multiple parasites on a single host still evolve independently of each other [48]. There is evidence that it is possible for multiple parasites to evolve on a single host without interference [22]. However, if the parasites do not evolve independently, our current approach for cophylogeny reconstruction would need to be modified to model the interdependence between parasites.

One approach is to impose further restrictions to only allow a single parasite to be associated with any one host. In this model, the leaf mapping between parasite and host phylogenies would be a bijection. In many systems, this altered model would be far to restrictive, but may be sufficient for some less complex cophylogeny comparisons (such as between gene trees and species trees in systematics).

2.3.3 Reticulate Networks

Another fundamental assumption is that the phylogenies of the evolving systems are trees. However, in the general case, phylogenies of evolving systems may allow recombination or convergence events [7].

Under such a model, the phylogeny graph of an evolving system is a directed acyclic graph with a single vertex of zero in-degree. This single *source* vertex is the *root* of the phylogeny graph and any *sink* vertices (vertices of zero out-degree) are the contemporaneous and current *leaf* species. The direction of the graph edges indicate the direction of time (and hence the acyclic condition is required to prevent cycles in time).

It is unclear what events need to be introduced to model cophylogeny reconstruction on reticulate networks. Whilst a modification to the *jungle method* [7] claims to work on reticulate networks, it still uses the standard four event model and so does not capture the full possibility of events introduced by the non-tree structure of the parasite and host phylogenies.

CHAPTER 3

Background Work

In this chapter, we review the current state of cophylogeny research. We focus on the developments that led to the contemporary model of cophylogeny reconstruction, and survey the recent reconstruction methods and results relating to the theoretical hardness of the problem.

3.1 Early Developments

To understand the structure of the cophylogeny reconstruction problem and the reasoning behind assumptions made in contemporary computational methods, it is necessary to review the historic contributions to the field.

Cophylogeny reconstruction grew from independent developments in several biological sub-disciplines. The fields of molecular systematics, parasitology, and biogeography all studied associations between hosts and clades dependent on these hosts. In studying these associations, it was noted that there was often substantial incongruence between presumably coevolving clades – for instance, incongruence between gene and organismal evolution [19]. These yet unexplained incongruence has led each discipline to develop techniques for what is now known as cophylogeny reconstruction.

3.1.1 Brooks Parsimony Analysis

Brooks [3] proposed one of the first methods to form "representation[s] of natural host-parasite relationships", establishing a method later coined Brooks' Parsimony Analysis (BPA) [54]. The method borrowed from the well-established Wagner Tree methods developed for phylogeny reconstruction by attempting to rephrase the cophylogeny reconstruction problem in some compatible way. This was achieved by coding parasite taxa as characters of the hosts (in phylogeny reconstruction, gene features

are used as characters of their *host* organisms). The paper provided clear motivating examples of application, but it failed to describe several important details of the protocol. The output of BPA requires *a posteriori* interpretation to recover the implied historic association events, but how this interpretation should be performed is left unspecified.

Wiley [54] gave a more formal and expanded description of the BPA method, applying the method in a biogeography context. Wiley explored the issue of how to interpret inconsistencies in BPA results caused by incongruence between the analysed clades. Nonetheless, some problematic ambiguities are still present in any interpretation. Importantly, Wiley acknowledged that the need for "careful *a posteriori* inspection" of any BPA result as a weakness, but believed that this is a limitation of the then-current technology and implementations. Many of these weaknesses were addressed in Brooks et al. [5], clarifying the need for two distinct processes when applying the BPA method: Primary BPA which is essentially the method described in the original Brooks [3] paper; and Secondary BPA which encapsulates some of the required *a posteriori* interpretations.

3.1.2 Reconciled Tree Methods

An alternate approach to BPA was proposed in Page [36, 37] and concretely defined in Page [38]. Page's work drew from independent and parallel developments in systematics, biogeography, and parasitology. One of the earlier conclusions reached by all three fields was that the incongruence between dependent clades may be apparent rather than real – the clades that exhibit incongruity are only partial (or erroneous) observations of the true evolutionary tree. All fields converged on a solution of attempting to reconstruct the true evolutionary tree from the partial observations [19, 33]. Noticing the similarity between the approaches used across disciplines, Page [38] generalised the problem of modelling the relations between *associates* and their hosts, formally describing the concept of reconciled trees. This landmark paper was one of the first to explicitly deal with the abstract problem of identifying historic relations between dependent evolutionary systems. It also formed an important basis for developing an alternative to BPA in cophylogeny analysis, necessary to avoid the large amount of possibly inconsistent *a posteriori* interpretation required to recover cophylogeny events from BPA results. The reconciled tree is described as a hypothetical *true* phylogeny tree of the parasite that *is* congruent to the host tree, and the original parasite tree is an incomplete observation (subtree) of this hypothetical tree.

Several such reconciled trees might exist and there needs to be some discriminatory criterion to identify which is more likely to be the true evolutionary tree. To this extent, Page identified the important

"maximising hypotheses of codivergence" – the more codivergence events implied by a reconstruction, the more likely it is to be the true reconstruction. Although the validity of the hypotheses is questioned in Ronquist [46] and Charleston [8], the hypothesis (and later variations of it) is still used in many contemporary optimal and heuristic methods [6, 15, 30, 39].

Whilst Page [38] was used as the foundations for a number of later cophylogeny reconstruction methods, it was in itself an incomplete solution. A crucial weakness in the reconciled tree method is the assumption of underlying congruence. There are two categories of historic associations: "association by descent" implied by congruence, and "association by colonisation" implied by incongruence [4]. Reconciled trees only consider descent associations, but Page recognised the need to incorporate both types of associations for any complete cophylogeny analysis. As a result, several techniques incorporating colonisation into reconciled trees were developed. These are collectively known as *Tree Mapping Methods* for cophylogeny reconstruction.

3.2 Toward Contemporary Cophylogeny – Tree Mapping

Page [39] made a first attempt to augment the reconciled tree method to take into account association by colonisation (more commonly know as horizontal transfer or host switching). Page's influential work described several critical concepts that form the foundations for most contemporary developments in cophylogeny reconstruction. First, Page presented a new representation of reconciled trees, superimposing the host and parasite clades with the associations given by the reconciled tree mapping, This is now the standard visualisation used to represent cophylogeny reconstructions. Page also provided a thorough discussion on some of the difficulties in incorporating host switches by demonstrating the existence of impossible host switches, and importantly, provided a simple criterion to discover them. Also identified are possible ambiguities related to multiple host switches. However, the discussion fell short of identifying the existence of weakly incompatible host switches [6, 46] – Page only believed that such ambiguities formed a problem of representing reconstructions rather than a parsimony and computational problem. The most critical contribution of the paper was the description of a concrete computational method for cophylogeny reconstruction. As with reconciled trees, the objective function used was maximum codivergence. The method begins by finding the reconciled tree mapping, and then iteratively incorporating host switches until codivergence cannot be increased. Whilst the algorithm was optimal (according to the maximum codivergence hypothesis), it is computationally expensive, requiring exponential time in the general case. Page acknowledged the inefficiency of the approach, but only provided heuristics for

improving performance. This limits the usefulness of implementations on large input cases with very low codivergence. Regardless, the method described was an important contribution: the first concrete algorithm presented to solve cophylogeny reconstruction that incorporated both descent and colonising associations. Page's work defined the structure to the cophylogeny reconstruction problem, codified in terms of tree mappings and four distinct event types, that has been used in computation approaches since.

One of the main problems with the approach in Page [38, 39] is the overly simplistic optimality criterion. By only trying to maximise codivergence, too little attention is given to the cost of the incongruence events (host switching, duplication, sorting). Early methods such as BPA and reconciled trees, whilst forming an important base for developing reconstruction algorithms, tended to disagree on the definition of optimality or parsimony. It is important in the design of any computational approaches to cophylogeny reconstruction, including the proposed investigation into approximation approaches, to correctly identify the optimality criterion, otherwise direct comparisons with other methods are spurious at best. The two related approaches governing contemporary interpretations of the cophylogeny reconstruction problem are reviewed: event cost estimates and Pareto optimality.

3.2.1 Event Costs

One interpretation of optimality in the cophylogeny reconstruction problem is to assign costs to each event (cophylogeny association) and find histories that minimise total cost. The method described by Page [39] can be rephrased in terms of this interpretation by assigning a negative cost to codivergence and zero cost to all other events. Ronquist [46] argued that this overly simplistic approach to parsimony is insufficient to distinguish between all the possible reconstructions with the same number of codivergences and that costs need to be assigned to all events for a more complete analysis. Ronquist suggested assigning costs inversely related to the likelihood of each event and described an algorithm to optimally solve cophylogeny reconstruction given these costs. However, this method only took into account two types of events – host tracking and host switching, whereas contemporary methods and also Page's tree method recognise four events. The salient idea of assigning costs to events and finding minimum maps has now become a popular approach to cophylogeny reconstruction, forming the optimality criteria for several recent methods [15, 30].

3.2.2 Pareto Sets

One of the difficulties in working with event costs is identifying the correct costs to use. Different costs can yield different minimal reconstructions [46]. Instead, Charleston [7] detailed a rephrasing of the cophylogeny reconstruction problem in terms of Pareto optimality. The number of each non-codivergent event for a given mapping is considered and the mapping is only admitted if it can be proven that no other mapping has equal or lower event costs *for all non-codivergent events* (that is, the mapping is not dominated by any other mapping). Rather than finding the optimal solution under a given cost scheme, this approach eliminates mappings that are definitely suboptimal, but accepts mappings that can be made optimal under *some* cost scheme. For example, a mapping with 2 host switches and 2 sorting events is definitely dominated by a mapping with 2 host switch and 1 sorting event, but not necessarily more costly than a mapping with 1 host switch and 3 sorting events. This admits a much larger number of solutions than Page [39] or BPA and will include the optimal results found by other parsimony and event cost methods, in stark contrast to the Ronquist [46] costs scheme which was partly motivated by the need to distinguish between "equally likely" solutions. By considering all Pareto optimal solutions, more equally likely solutions are introduced. However, this admits new approaches to cophylogeny analysis such as finding strong recurring signals across Pareto optimal solutions.

Without specific and accurate knowledge of the prior distribution of cophylogeny events, assigning costs to events may only yield erroneous solutions. Any method claiming to solve the cophylogeny reconstruction problem must therefore work for any set of event costs or alternatively take the Pareto optimality approach.

3.3 Current Heuristics and Approaches

By the late 1990s, the cophylogeny reconstruction problem had become well defined and concrete. Ronquist [48] described the contemporary methods of cophylogeny reconstruction as "event based", reflecting the trend toward algorithms that explicitly enumerated the events or associations describing the historic reconstruction. Event costing also became the *de facto* measure of optimality replacing the less sophisticated maximum codivergence hypothesis.

Development first focused on exact solutions, but the apparent intractability of optimal solution methods led to the development of heuristic methods.

3.3.1 Cost Matrix

Ronquist [46] described a method to find optimal solutions in a two event model of cophylogeny reconstruction. The algorithm described relies on constructing a step cost matrix representing the cost of moving between hosts. The cost matrix allowed general parsimony algorithms to be applied to find minimal cost maps between parasite and host. Ronquist's work is the first to identify the full extent of difficulties in forming reconstructions with compatible host switch interactions. Two methods were described: a "quick" method that may admit incompatible switches; and the slower "exact" method which constructs optimal maps without such host switch issues. Both methods also depart from the traditional "maximum codivergence" criterion, instead identifying and costing two event types as an optimality objective. Ronquist established that it is possible (and more importantly, tractable) to find optimal cophylogeny reconstructions given timing information of the hosts – given sufficient timing information, it is possible to use the "quick" method without admitting internal inconsistencies. The "exact" method described essentially either requires timing information, or searches over the space of possible timings. Unfortunately, the number of possible timings is exponential in the number of hosts – the reconstruction problem becomes intractable again. A second weakness of the algorithms proposed is the basis of a two event model – "host tracking" and "host switching". This means that the cost of duplication events is not taken into account, so whilst the solutions found are optimal in terms of the two event criteria, it is not necessarily accurate against the true reconstruction and limits the ability to adapt the method to different models of co-evolution (duplication will always be favoured over host switch and loss events).

3.3.2 Jungles

A second exact approach was described in Charleston [6] using the more familiar and biologically compatible setting of four distinct event types. The main contribution of the paper was the "jungle" data structure which contains all cophylogeny mappings that are optimal under some cost scheme. The jungle structure thus captured the Pareto set of the four event cophylogeny reconstruction problem. Charleston noted the fact that events and associations can be viewed independently and are often shared amongst many Pareto optimal solutions. This led to the construction of the jungle data structure that compactly captures this information. It was the first method to tackle the difficulty of accurately estimating event costs by considering Pareto optimality. Charleston demonstrated a dynamic programming algorithm to traverse the jungle structure given any event cost vector. The reconstruction yielded by the traversal is not only the global optimal for the given event costs, but also correctly handles multiple host switches.

However, as with Ronquist's exact method, the computational complexity of the algorithm prevents it from being practical for non-trivial input cases or where there is little congruence between host and parasite clades.

The jungle data structure was later improved to account for the more general reticulate cophylogeny reconstruction problem – both host and parasite phylogenies could be general directed acyclic graphs rather than trees [7]. Whilst still computationally intractable, this is one of the few solutions able to handle reticulate phylogenies and still maintain optimality and internal consistency. Charleston's work also included an enumeration and formalisation of some of the assumptions and constraints that form the cophylogeny reconstruction problem, giving the problem some preliminary mathematical grounding.

3.3.3 TARZAN

The prohibitive complexity of finding the exact solution for cophylogeny reconstruction resulted in the development of a number of heuristic algorithms. Merkle and Middendorf [30] proposed a heuristic that takes advantage of provided timing information to constrain the search space. The described algorithm is implemented in the software tool TARZAN [26]. Merkle and Middendorf's method also used the jungle structure described by Charleston [6]. The performance advantage of the heuristic is gained by relaxing the constraints on compatible host switches and using the extra timing information to eliminate a large number of possible reconstructions. However, the heuristic method admits inconsistent solutions and attempts to resolve them at the end of the search. This is a crucial weakness of the heuristic as minimal cost resolution of inconsistent host switches is intractable. This means that resolution of the host switches is done non-optimally using a simple heuristic. Whilst it is expected that such a fast algorithm might not be guaranteed to find the optimal solution, the larger problem with *post hoc* resolution of host switches is the possibility that no consistent resolution of the host switches may be found by the heuristic. This problem was acknowledged by Merkle and Middendorf and further demonstrated in Conow et al. [15]. Nonetheless, TARZAN is an extremely fast heuristic for cophylogeny reconstruction and can find solutions constrained by partial timing information – a featured not considered by previous implementation.

3.3.4 JANE

Improving on the concepts developed in TARZAN, Conow et al. [15] proposed a new heuristic implemented in the software package JANE [14]. Conow et al.'s work was based on a recent result shown

in Libeskind-Hadas and Charleston [29] that described a polynomial time dynamic programming algorithm for optimally solving cophylogeny reconstruction (with a four event model) if divergence timing information is fixed for the host tree. This is similar to the ideas used in TARZAN to bound the search space using timing information. The method proposed by Conow et al. is one of the first to utilise common local search heuristics in approaching the intractability of the cophylogeny reconstruction problem. Rather than sacrificing consistency (or feasibility) of solutions as in TARZAN [30], the performance gain in the JANE method is derived from its search for locally optimal rather than globally optimal reconstructions. Conow et al. proposed the application of genetic programming algorithms on a "population" of fixed divergence timings, with the cost of the optimal solution for a timing as the fitness function. This work is important in realising a heuristic that is guaranteed to provide feasible solutions – whilst optimality may not be guaranteed, the results will always be internally consistent and so can provide useful bounds on event counts and costs. However, Conow et al. did not demonstrate mathematically provable bounds on just how far a local optimum is from the global optimum.

Recently, two further iterations of the JANE software were released. JANE 2 made significant implementation improvements to the dynamic programming algorithm used, allowing the software to very rapidly converge on locally optimal solutions even for large input sizes [16]. JANE 3 incorporates a new event type, "failure to diverge" [2], to account for widespread taxa, usually prohibited by other cophylogeny reconstruction methods.

3.3.5 CORE-PA

A novel method of finding event based optimal solutions was proposed in Merkle et al. [31] and implemented in the CORE-PA software program [32]. Merkle et al.'s method used a dynamic programming approach to solve the cophylogeny reconstruction problem given some cost scheme without taking into account host switch incompatibilities. Attempts are made to resolve the incompatibilities *a posteriori* as in Merkle and Middendorf [30]. However, the novel contribution in Merkle et al. [31] was the proposal to adaptively seek the best cost scheme rather than consider some single fixed input cost scheme. The common consensus is that the correct cost scheme to use has an inversely proportional relation with the probability of each event occurring [48]. Merkle et al. [31] observed that if this is the case, then when correct cost scheme is used to find the optimal reconstruction, the distribution of the event counts should be inversely proportional to the cost scheme. In this way, it is possible to evaluate how close any given cost scheme is to the optimal scheme – the reconstruction yielded by the optimal cost scheme should

have a distribution of event counts that matches the predicted counts. Merkle et al. [31] proposed the use of a metaheuristic (local search) to seek the optimal cost scheme, using the described dynamic program to evaluate the quality of a given cost scheme for an input tanglegram.

3.3.6 TREEMAP 3

A recent development made by Charleston [10][priv. comm.] is the TREEMAP 3 software (successor to the jungle implementation of TREEMAP 2 [40] and modified reconciled tree implementation of TREEMAP [13]). Like JANE, TREEMAP 3 uses common local search algorithms to find locally optimal solutions that are guaranteed to be feasible [11]. Extending the concepts of TREEMAP 2, the algorithm still finds and lists multiple Pareto optimal solutions, however, rather than guaranteeing global optimality, the solutions found are only guaranteed to be locally Pareto optimal (that is, no dominating solution has been encountered in the search space). Charleston's algorithm utilises a beam search seeded with the reconciled tree as well as random feasible reconstructions. The search is across the space of reconstructions rather than the space of timings as in Conow et al. [15]. This is a novel approach to the problem that, when coupled with the Pareto optimality objective, essentially eliminates reconstructions that are provably non-optimal rather than explicitly searching for a provably optimal result. This is a heuristic algorithm to provide reconstructions without the need for explicit event costs. However, as with JANE, no bounds have been proven regarding the quality of the reconstructions found by TREEMAP 3.

3.4 Intractibility and Approximations Results

Even from very early work in cophylogeny reconstruction it was apparent that the problem was computationally difficult. All methods discussed so far are either exponential time or require a trade-off between guaranteed optimality and feasibility. More recently, several mathematical proofs have surfaced to quantify how difficult the reconstruction problem actually is. These proofs demonstrate that many aspects of cophylogeny reconstruction problem are NP-hard and thus, assuming that P does not equal NP, computationally intractable to solve optimally.

3.4.1 Moving Back Landing Site Problem

In producing the TARZAN algorithm, Merkle and Middendorf [30] also presented an interesting computational complexity result to demonstrate the difficulties in dealing with minimum cost host switch

compatibility, extending the discussion in Charleston [6]. Merkle and Middendorf defined the "Moving Back Landing Sites Problem" which essentially captures the problem of resolving a set of incompatible switches whilst minimising total cost. The problem is shown to be NP-complete by reduction from the Feedback Arcs Set Problem (FASP), a known NP-complete problem [18]. This was one of the first intractability proofs presented in relation to the cophylogeny reconstruction problem and offered concrete insight into the computational hardness of the problem. Of course the computational intractability of the Move Back Landing Site Problem does not automatically imply the intractability of the cophylogenetic reconstruction problem.

3.4.2 The Cophylogeny Reconstruction Problem

An initial approach to proving the intractability of the the cophylogeny reconstruction problem was made in Libeskind-Hadas and Charleston [29]. The reticulate cophylogeny reconstruction problem is considered (where the host and parasites are organised into an evolutionary network rather than a tree). Libeskind-Hadas and Charleston demonstrated a polynomial time algorithm for solving the reticulate problem if timings of the networks are fixed. This is similar to earlier results in Ronquist [46, 47], but generalised for the reticulate case. However, the case where even just two possible timings are given for codivergence events leads to the intractability of the problem. The proof is by reduction from 3-SAT, a known NP-complete problem [18]. The truth value of a variable is associated with the ordering of two codivergence events (out of even just the two possible orderings). Whilst an important result in showing the hardness of the cophylogeny reconstruction problem, it still does not give a definitive proof of intractability. The proof allows the host history to be a directed acyclic graph rather than a tree and relies on the ability to assign relative timing ranges to the networks.

A proof showing that the common cophylogeny reconstruction problem is NP-complete was presented by Ovadia et al. [35]. This work extended on Libeskind-Hadas and Charleston's proof, also reducing from 3-SAT. A similar approach was used, defining gadgets such as the *k-thorn*, but also introducing new structures to remove the reliance on the reticulate nature of the networks or the need for attaching timing ranges. Ovadia et al. demonstrated mathematically the intractability of cophylogeny reconstruction, justifying the need for efficient heuristic or approximation approaches to the problem.

3.5 Summary

The study of cophylogeny is a generalisation of problems studied across a wide range of biological disciplines. This has led to the early developments of many different approaches and interpretations of cophylogeny reconstruction. Recently, a more concrete definition of cophylogeny reconstruction has led to more focused development of algorithms. Initial attempts focused on finding optimal results (and defining optimality), but recent intractability results showed that this cannot be achieved efficiently (assuming P does not equal NP). Thus there has been a shift in focus from exact solutions to local search heuristics – a common feature in the most recent approaches.

Whilst some effort has been applied to identifying local search methods, there is still a need for solutions with guaranteed bounds. To date, the only exact solution implementation – the TREEMAP 2 program based on Charleston [6] – is impractically slow for all but the smallest real life datasets. None of the existing heuristics have been proven to have bounded approximation ratios – there exists very few analytical results about the reconstruction problem in general.

In the next chapter, we supply an in-depth analysis of the temporal constraints on a feasible cophylogeny reconstruction. This is an aspect of cophylogeny reconstruction that is often not explored in any great detail in literature, but is an underlying cause of the intractability of the problem.

CHAPTER 4

Temporal Incompatibilities and Host Switch Resolution

In this chapter, we present a careful examination of temporal incompatibilities in cophylogeny reconstruction. We formally characterise the temporal aspects of reconstructions, review existing approaches in terms of this formalisation, and present an alternate set of conditions to enforce temporal compatibility.

It is widely accepted that *host switch* incompatibilities is the main source of complexity when trying to solve the Cophylogeny Reconstruction Problem using event costing methods [6, 7, 15, 30, 46, 49]. However, there is very little existing literature that discusses in depth precisely what conditions cause incompatibilities. The seminal work in Charleston [6] and later extended in Charleston [7] provide a set of conditions for detecting host switch incompatibilities and is used in several popular heuristic methods for cophylogeny reconstruction. However, the condition is costly to maintain and no proof has been given for the sufficiency of the conditions.

In Section 4.1, we discuss the definitions of *host switches* and *incompatibilities* and how these affect existing event cost methods of cophylogeny reconstruction. In Section 4.2, we discuss the conditions required to maintain compatible host switches. In Section 4.3, we first review the conditions from Charleston [6, 7] and prove the sufficiency of these conditions before proposing a new set of conditions of less complexity and show that it is equally sufficient in maintaining host switch compatibility in Section 4.4.

4.1 Background

4.1.1 Anatomy of a Host Switch

A host switch represents some parasite that has colonised a new host after duplicating. Host switches have three feature points:

- the *take-off site* is the point at which the parasite leaves the host its parent speciated on;
- the *landing site* is the point at which the parasite started to colonise the new host;
- the *termination site* is the final point at which the parasite speciates on a descendent of the host the parasite landed on.

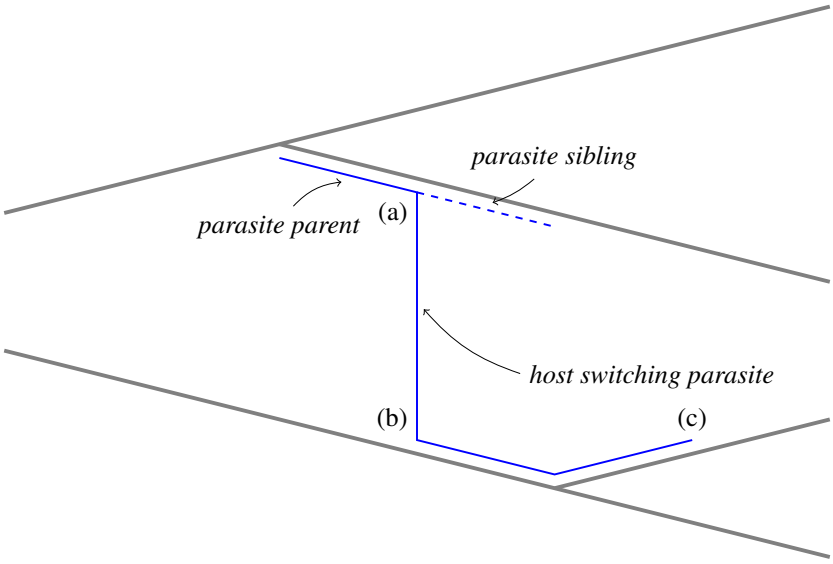


FIGURE 4.1. A host switch event: (a) Take-off site; (b) Landing site; and (c) Termination site.

Each *site* is a location on a host branch. For brevity, we will refer to the host on which the take-off site is located the *take-off host*. Similarly, we will refer to the *landing* and *termination host*.

In the model of cophylogeny reconstruction we are considering, we don't consider widespread parasites, so each parasite can only infect one host lineage. Hence, any colonisation of a new host by a parasite must be preceded immediately by a speciation event. The take-off site is then also the point at which the host switch parasite's parent speciated. Therefore, the parent parasite must also have infected the take-off host.

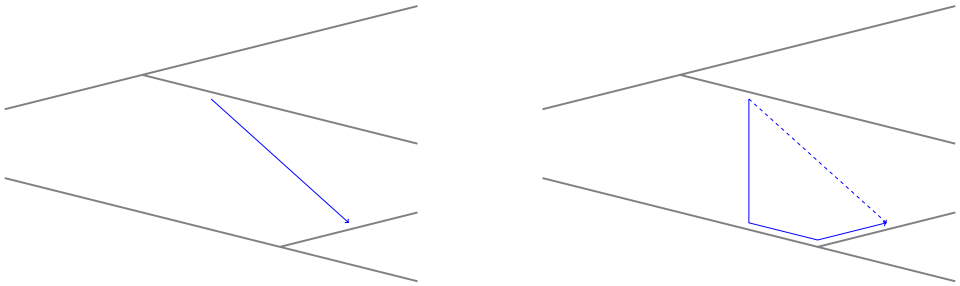
We also require that the sibling of the host switch parasite remain on the take-off host. This is known as the *traceable condition* [7] and is required to ensure that we only consider events that ultimately affect the contemporary parasite-host associations (events that are *traceable*).

4.1.2 Incompatibilities

In traditional *mapping* methods, such as TREEMAP, TREEMAP 2 and TARZAN, each parasite taxon is associated with a host taxon on which it speciates [6, 30, 39]. The parasite speciation is associated with either the edge of the host or the host speciation event. In this sense, the solutions are true *mappings* in the mathematical sense.

The main difficulty found in mapping methods is in choosing host switches in a compatible manner. The tree structure of the two phylogenies implicitly imply some ordering restrictions on events. However, these are static as the tree structures don't change during cophylogeny reconstruction. It is therefore easy to ensure that events are ordered in a way compatible with these ordering restrictions. However, host switches introduce extra ordering information on the take-off and landing sites [39]. For a set of host switches in a reconstruction, we need to ensure that no contradictory orderings are introduced. We call a set of host switches *incompatible* if there are contradictory orderings.

In mapping methods, only the take-off and termination sites are explicit as only the location of parasite speciations are recorded. This means that the landing site may be any ancestor of the termination site. Some set of host switches can be made compatible by moving the landing site of some host switches earlier in the host tree than the termination site, incurring extra loss events. These are known as *weak incompatibilities*. Host switches that cannot be resolved are known as *strong incompatibilities* [7].



(A) Host switch (B) Alternate landing site

FIGURE 4.2. A host switch defined only by a take-off and termination site can have many possible landing sites.

Mapping methods including TARZAN, TREEMAP, and TREEMAP 2 try to ensure that the host switches are compatible *a posteriori*. A reconstruction is first created without full regard to compatibility and then adjusted (if possible) to take into account the event orderings implied by host switches. Whilst reconstructions with strongly incompatible host switches can be disregarded as infeasible, weak incompatibilities are a much harder problem. Ignoring or non-optimal resolution of weak incompatibilities can hide the existence of better (and even optimal) reconstructions, but optimal resolution *a posteriori* is an NP-complete problem [30].

4.1.3 Newer Approaches

A new approach to the reconstruction problem avoids the problem of host switch incompatibilities by defining *relative times* in which events including host switches can occur. All active parasite-host associations are considered for all possible relative times of events, an idea proposed in [29]. Host switches cannot cross multiple relative times, and so do not introduce any ordering information. This inverts the problem of incompatibilities by restricting possible host switches based on predetermined ordering information, rather than choosing host switches and then considering the consequences on speciation orders. Whilst this ensures that solutions based on this approach cannot have host switch incompatibilities, a lot of complexity and redundant information is introduced to model the notion of relative event times.

The approach is the basis for an ILP proposed in [28]. The excess modelling required to capture all possible relative event times made the ILP infeasibly complex to solve even for very small problem instances. JANE [15] is also based on the same concept captured in a dynamic programming algorithm [29] and uses a genetic algorithm to search for optimal reconstructions.

4.2 Feasible Reconstructions and Event Orders

In cophylogeny reconstruction, we are interested in the historic association (*events*) between phylogenies. This makes it necessary to consider the relative ordering of events: some events may only be possible if certain other events have previously occurred. For a reconstruction to be feasible, it is necessary that the ordering of events remain consistent. There are three sources of ordering information on events – the implicit ordering implied by the host phylogeny; the implicit ordering implied by the parasite phylogeny; and the ordering implied by host switches.

4.2.1 Phylogeny Implicit Orders

Phylogenies implicitly have a time dimension – the root of the phylogeny is the *earliest time*, and paths toward leaves are *later* in time. This imposes a partial ordering on the speciation events in each phylogeny.

Each event in a cophylogeny reconstruction is associated with a speciation event:

- duplications and host switches are associated with parasite speciation events;
- losses are associated with host speciation events; and
- cospeciations are associated with both host and parasite speciation events.

Hence the partial order implied by the phylogenies must also enforce an ordering on reconstruction events.

The parasite phylogeny requires:

CONDITION 4.1. If two events e_1 and e_2 are associated with parasites p_1 and p_2 respectively and p_2 is a descendant of p_1, then e_2 must have occurred after e_1.

Similarly, the host phylogeny requires:

CONDITION 4.2. If two events e_1 and e_2 are associated with hosts h_1 and h_2 respectively and h_2 is a descendant of h_1, then e_2 must have occurred after e_1.

If these are the only ordering conditions required by events, we can easily ensure the feasibility of any reconstruction – all pairs of infeasible events are known *a priori* as the tree structure of the host and parasite phylogenies are known *a priori*.

4.2.2 Host Switch Orders

The take-off and landing sites of a host switch are considered to be simultaneous – both are a part of a single atomic event. This implies extra ordering information on events not otherwise implicit in the host or parasite phylogenies. For the take-off and landing sites to represent the same time, the take-off and landing hosts have existed together at some stage. This is equivalent to requiring the two host edges in the host phylogeny to overlap, or in terms of host speciation events:

CONDITION 4.3. The speciation of the parent of the take-off host precedes the landing host speciation.

CONDITION 4.4. The speciation of the parent of the landing host precedes the take-off host speciation.

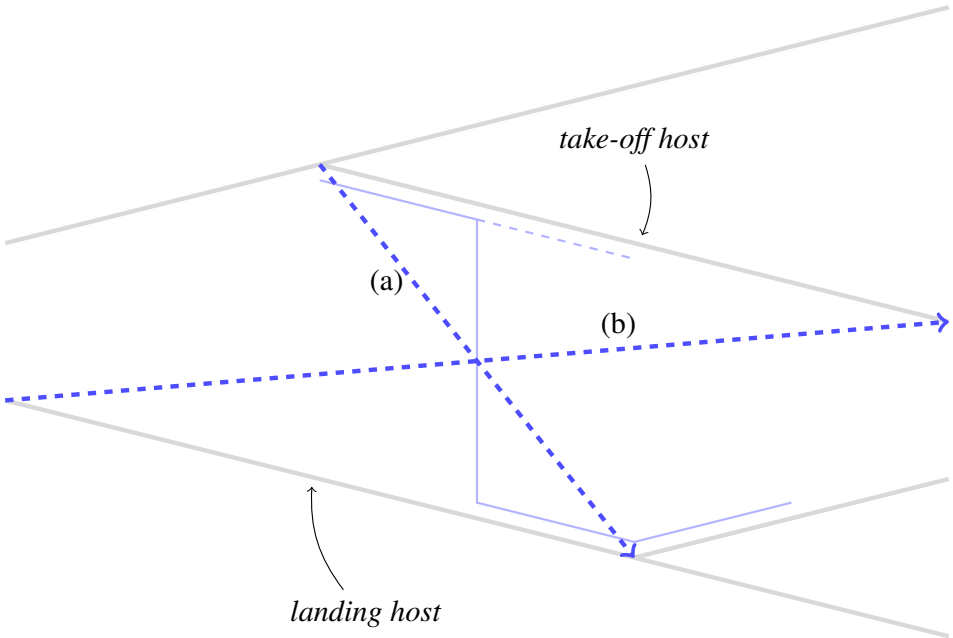


FIGURE 4.3. Host speciation ordering necessary for the host switch in Fig. 4.1 to be feasible. A dashed arrow indicates that the speciation at the tail of the arrow must occur before the speciation at the head of the arrow. (a) Condition 4.3; and (b) Condition 4.4

These conditions were first postulated in Page [38] as conditions that could prohibit host switches. However, these conditions alone are not sufficient. Although the hosts are guaranteed to overlap by

these conditions, it does not ensure that both the take-off and landing sites are simultaneous – the take-off and landing sites may still occur outside the overlap.

The stronger conditions necessary and sufficient to guarantee that both sites are within the overlap are:

CONDITION 4.5. The speciation of the parent of the take-off host precedes the landing site.

CONDITION 4.6. The speciation of the parent of the landing host precedes the take-off site.

CONDITION 4.7. The speciation of the landing host succeeds the take-off site.

CONDITION 4.8. The speciation of the take-off host succeeds the landing site.

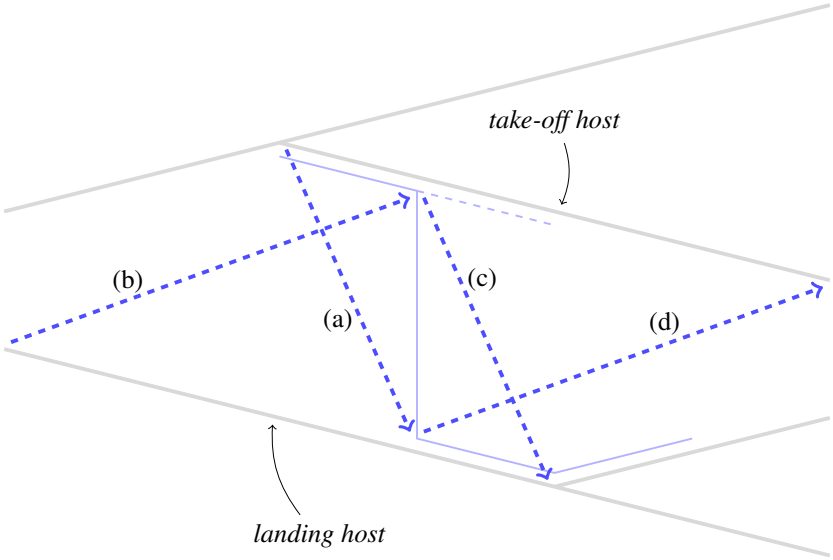


FIGURE 4.4. Event ordering (dashed arrows) necessary *and* sufficient for a feasible host switch. (a) Condition 4.5; (b) Condition 4.6; (c) Condition 4.7; and (d) Condition 4.8

We prove that these conditions are in fact sufficient to ensure that both sites are contemporaneous.

THEOREM 4.1. *If Conditions 4.5 to 4.8 are the only event timing constraints enforced for all host switches in an otherwise feasible reconstruction, then for any host switch, there does not exist another event that must occur at a time between the take-off and landing sites.*

PROOF. Let p be the host switch parasite and let h_1 be the take-off host and h_2 be the landing host. By assumption, Conditions 4.5 to 4.8 hold, so h_1 and h_2 overlap and the take-off and landing sites occurs in this overlap.

Suppose there does exist some event e_1 associated with a parasite p_1 that must occur at a time between the two sites.

Suppose further that p_1 is related to p. If p_1 is a descendent of p, then p_1 is also a descendent of the parent of p. By the ordering imposed by the parasite phylogeny, e_1 must have occurred after both the take-off and landing sites of p, contradicting the assumption that e_1 occurred between the two sites. If p_1 was an ancestor of p, then p_1 must also be an ancestor of the parent of p, since by definition, there are no parasites between p and the parent of p. By the ordering imposed by the parasite phylogeny, e_1 must have occurred before both the the take-off and landing sites of p, again a contradiction.

So p_1 must be unrelated to p. Then the parasite phylogeny does not impose any ordering constraints between e_1 and the host switch sites of p.

e_1 must occur between the take-off and landing sites and so cannot occur on an ancestor or descendent of either h_1 or h_2. Hence the host phylogeny does not impose any ordering constraints between e_1 and the host switch sites of p.

So host switches must be imposing the ordering between e_1 and the host switch sites of p. If e_1 is between the take-off and landing sites, then the landing site must succeed e_1. So the host switches must be imposing some ordering that prevents the landing site from moving back to before e_1. However, by assumption, host switches only enforce Conditions 4.5 to 4.8 so any ordering information preventing the landing site from moving back cannot be direct – it is only transitive through the parent speciation of the take-off or landing hosts. But any such condition would also apply to the take-off site of p, the take-off site would also succeed e_1, contradicting the definition of e_1.

Hence, in no situation could such an e_1 exist. □

So any reconstruction that ensures Conditions 4.5 to 4.8 hold for all host switches must be fully compatible.

We now show that in the context of cophylogeny reconstruction, Condition 4.8 is not necessary.

THEOREM 4.2. *If there exists a host switch of parasite p from host h_1 to host h_2 such that Conditions 4.5 to 4.7 holds but Condition 4.8 does not hold, then it is always possible to move back the landing site such that all four conditions will hold.*

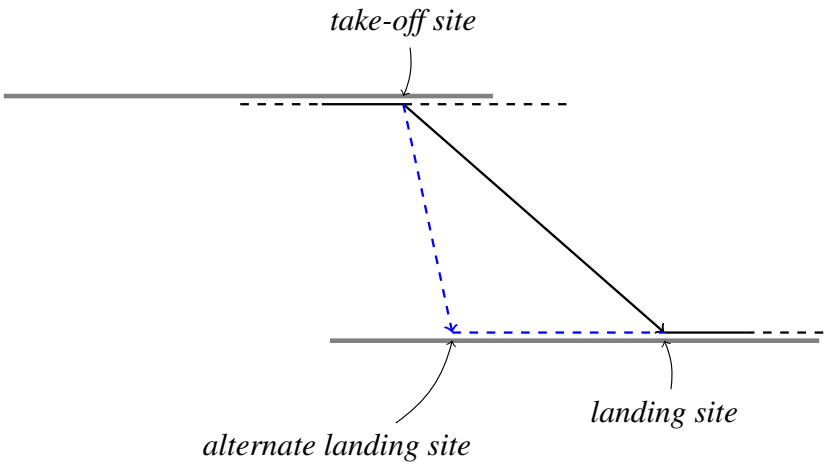


FIGURE 4.5. A host switch that is consistent with Conditions 4.5 to 4.7 but violates Condition 4.8 can move back to a *alternate landing site* that is consistent with Condition 4.8 as well.

PROOF. Condition 4.8 does not hold, so we must have that the landing site succeeds the take-off host speciation. However, the other conditions hold, so the take-off site must overlap h_2. h_1's speciation succeeds the take-off site (by definition). Hence, there is an *alternate landing site* on h_2 at or after the take-off site such that h_1's speciation succeeds this alternate landing site. We need to show that it is always possible to move back the landing site to this alternate landing site.

We consider each source of ordering constraints in turn.

Any ancestor of p that imposes some ordering that prevents p from moving back must also impose the same condition on p's parent, hence cannot prevent the landing site from moving back to the alternate landing site which is at or after the landing site (and hence p's parent speciation event). So the parasite phylogeny does not prevent the landing site from moving back.

Any ordering implied by the host phylogeny that prevents the landing site from moving back must apply to either the parent speciation of h_1 or the parent speciation of h_2. If it applies to the parent speciation of h_1, then clearly it must apply to the take-off site (which is on h_1). If it applies to the parent speciation of h_2, then it must also apply to the take-off site by Condition 4.6.

We have shown that Conditions 4.5 to 4.8 are the only conditions implied by host switches, hence any host switch conditions do not directly impose ordering constraints on the landing site of p. Host switches can only indirectly impose constraints prevent p from moving back by transitive ordering constraints

through the parent speciations of h_1 or h_2, but we have shown that this does not prevent the landing site moving back to the alternate landing site.

Hence, it is always possible to move back the landing site to an alternate landing site within the overlap of h_1 and h_2. □

So to ensure that a set of host switches are compatible, we need only ensure that Conditions 4.5 to 4.7 hold for all host switches in the set.

4.3 j-vertex – The Jungle Method

The method used in many recent mapping heuristics [6, 30] to detect host switch incompatibilities in reconstructions is based on the method employed in the *jungle data-structure* [6].

The nodes of the *jungle*, known as j-vertices, are associations between parasite taxa and host taxa.

DEFINITION (j-vertex). A j-vertex is a three tuple $(p, h, t \in \{1, 2\})$ where p is a parasite, h is a host, and $t = 1$ if the parasite p is associated with the speciation of h (a node in the host phylogeny) otherwise $t = 2$ (associated with an edge of the host phylogeny) [7].

For convenience, we will denote $(p, h, *)$ to mean, without loss of generality, either $(p, h, 1)$ or $(p, h, 2)$. A set of conditions impose partial orderings on j-vertices of a reconstruction:

- If host h_1 succeeds h_2 (e.g., implied by the host phylogeny), then $(p_1, h_1, *) \preceq (p_2, h_2, *)$ for any parasites p_1, p_2.
- If parasite p_1 succeeds p_2 (implied by parasite phylogeny), then $(p_1, h_2, *) \preceq (p_2, h_2, *)$ for any hosts h_1, h_2.
- $(p_1, h, 2) \preceq (p_2, h, 1)$ for all hosts h and parasites p_1, p_2.

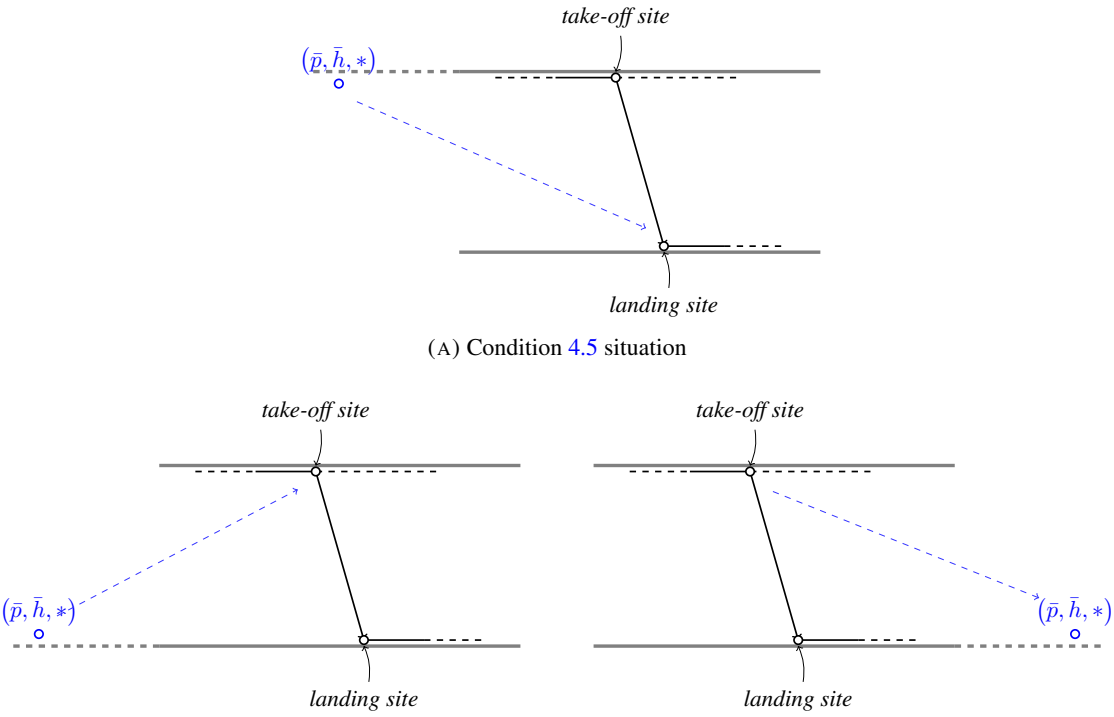
These conditions captures the requirement that mappings are compatible with the event orderings implied by both parasite and host phylogeny (Conditions 4.1 and 4.2). Contradictions in this partial ordering of j-vertices means the reconstruction is infeasible due to temporal constraints.

A host switch in jungle terms is a pair of j-vertices of the form $((p', h_1, 2), (p, h_2, *))$ where p is the host switching parasite with take-off site $(p', h_1, 2)$ and landing site $(p, h_2, *)$. For each such host switch in the reconstruction, Conditions 4.3 and 4.4 are enforced [30].

4.3.1 Sufficiency

We have shown that these conditions are insufficient alone to ensure host switch compatibility. However, the defined partial ordering on j-vertices along with Conditions 4.3 and 4.4 are together sufficient.

THEOREM 4.3. *For any set of j-vertices that form a reconstruction, if the j-vertices remain in a consistent partial order when Conditions 4.3 and 4.4 are imposed, then Conditions 4.5 to 4.7 must also hold.*



(A) Condition 4.5 situation

(B) Condition 4.6 situation

(C) Condition 4.7 situation

FIGURE 4.6. j-vertices affected by imposing Conditions 4.5 to 4.7 on a host switch.

PROOF. Consider a host switch $((p', h_1, 2), (p, h_2, *))$. Let $h_1{}'$ be the parent of h_1, $h_2{}'$ be the parent of h_2.

Consider some j-vertex $(\bar{p}, \bar{h}, *)$ where $\bar{h}$ precedes $h_1{}'$ (Fig. 4.6a). But by Condition 4.3, we have that $h_1{}'$ precedes h_2, so $\bar{h}$ precedes h_2 by transitivity. Hence $(\bar{p}, \bar{h}, *)$ precedes $(p, h_2, *)$, as required by Condition 4.5.

Now, consider some j-vertex $\left(\bar{p}, \bar{h}, *\right)$ where $\bar{h}$ precedes ${h_2}'$ (Fig. 4.6b). But by Condition 4.4, we have that ${h_2}'$ precedes h_1, so $\bar{h}$ precedes h_1 by transitivity. Hence $\left(\bar{p}, \bar{h}, *\right)$ precedes $(p', h_1, 2)$, as required by Condition 4.6.

Finally, consider some j-vertex $\left(\bar{p}, \bar{h}, *\right)$ where h_2 precedes $\bar{h}$ (Fig. 4.6c). The certainly the landing site $(p, h_2, *)$ precedes $\left(\bar{p}, \bar{h}, *\right)$ (by host phylogeny ordering). But the take-off site precedes the landing site (by parasite phylogeny ordering). Hence by transitivity, the take-off site $(p', h_1, 2)$ precedes $\left(\bar{p}, \bar{h}, *\right)$, as required by Condition 4.7. □

4.3.2 Complexity

We have proven that the j-vertex partial order conditions are sufficient to prevent host switch incompatibilities. However, it is a very expensive condition to maintain.

For a tanglegram of n hosts and m parasites, the number of possible j-vertices is $O(mn)$. The most expensive axiom of the partial order to check and maintain is transitivity, applying to all triples of the poset. Hence maintaining the partial ordering on j-vertices is a $O(n^3m^3)$ operation.

4.4 Proposed Method

Our proposed method only maintains a partial order on the host speciation events. Any violation of the host speciation order represents some temporal infeasibility in the reconstruction, including host switch incompatibilities.

Associations in this model are pairs (p, h) where p is some parasite and h is some host. A reconstruction is a set of such association pairs – for any parasite p that infects host h at any time in the reconstruction, we include the pair (p, h).

We begin with a partial ordering on host speciations according to the host phylogeny. The first additional condition we enforce on host speciation ordering is:

CONDITION 4.9. If a parasite p_1 is mapped to a host h_1 and p_2 is a descendent of p_1 mapped to h_2, then the parent of h_1 must have speciated before h_2.

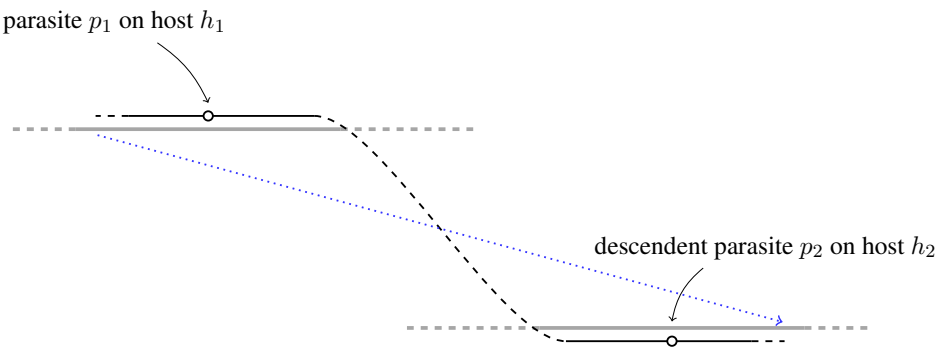


FIGURE 4.7. Host speciation order under Condition 4.9.

We show that without host switching, Condition 4.9 is equivalent to Conditions 4.1 and 4.2:

THEOREM 4.4. *Consider some reconstruction without host switching. Two associations* (p_1, h_2) *and* (p_2, h_2) *violates Condition 4.9 if and only if the associations violate Condition 4.1 or Condition 4.2.*

PROOF. Suppose (p_1, h_2) and (p_2, h_2) violate Condition 4.9. So we have p_2 is a descendent of p_1, but the parent of h_1 speciated after h_2 speciated. By Condition 4.1 we have that (p_1, h_2) preceded (p_2, h_2), but by Condition 4.2 we would have that (p_2, h_2) preceded (p_1, h_2), a contradiction. So one of Condition 4.1 or Condition 4.2 is violated.

Now, suppose (p_1, h_2) and (p_2, h_2) violates Condition 4.1. So p_2 is a descendent of p_1, but (p_2, h_2) precedes (p_1, h_2). We don't consider host switches, so the only condition enforcing the fact that (p_2, h_2)

precedes (p_1, h_2) is Condition 4.2. So h_1 is a descendent of h_2. Clearly then h_2 could not have speciated after the parent of h_1 speciated, so Condition 4.9 is also violated.

Finally, suppose (p_1, h_2) and (p_2, h_2) violates Condition 4.2. So h_2 is a descendent of h_1, but (p_2, h_2) precedes (p_1, h_2). We don't consider host switches, so the only condition enforcing the fact that (p_2, h_2) precedes (p_1, h_2) is Condition 4.1. So p_1 is a descendent of p_2. But by Condition 4.9 we must have that the parent of h_2 speciated before h_1 speciated, which is impossible as h_2 is a descendent of h_1. So Condition 4.9 is also violated. □

We also enforce Condition 4.4 on the host speciation partial order. We show that Conditions 4.4 and 4.9 is sufficient to guarantee that the take-off and landing hosts of any host switch must overlap.

THEOREM 4.5. *Condition 4.9 implies Condition 4.3.*

PROOF. Consider some host switch of parasite p (with parent p') from h_1 to h_2. So p and h_2 are associated and p' and h_1 are associated.

p is a descendent of p', so by Condition 4.9, we have that h_1's parent speciated before h_2 speciated. This is precisely Condition 4.3. □

So Conditions 4.4 and 4.9 must imply Conditions 4.3 and 4.4 which is enough to guarantee the overlaps between take-off and landing hosts.

Finally, we introduce one further constraint on host speciation orders:

CONDITION 4.10. For any parasite p that is first associated with host h_1, and for any descendent of p's sibling p^+, if p^+ is associated with some host h_2, then h_1's parent speciation must precede h_2's speciation.

4.4.1 Necessity

We first show the necessity of Conditions 4.3, 4.9 and 4.10 to argue that these conditions are not too strong.

THEOREM 4.6. *Any reconstruction that violates any of Conditions 4.3, 4.9 and 4.10 must have some temporal inconsistency.*

PROOF. First suppose the reconstruction violates Condition 4.9. If the violation does not involve a host switch, then by Theorem 4.4, this is equivalent to some inconsistency between host phylogeny and parasite phylogeny implied event orderings. If the violation does involve a host switch, by Theorem 4.5, this is also a violation of Condition 4.3 – a necessary condition. Hence in either case, the reconstruction must have some temporal inconsistency.

Now suppose the reconstruction violates Condition 4.3. But we know Condition 4.3 to be a necessary condition: otherwise there is no overlap between take-off and landing hosts of some host switch, contradicting host switch implied event timings.

Finally suppose the reconstruction violates Condition 4.10. So there is some parasite p which first maps onto some host h_1 and some descendent of p's sibling p^+ which maps onto some host h_2, and we have h_2's speciation preceding h_1's parent speciation. Let p' be the parent of p (and so the lowest common ancestor of p and p^+).

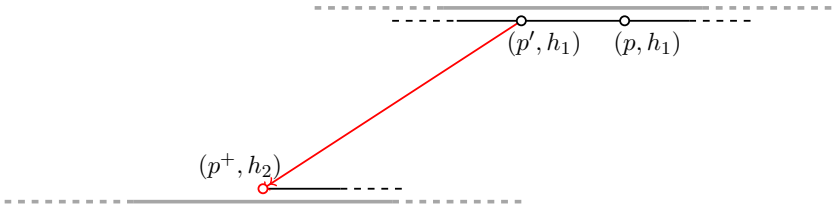


FIGURE 4.8. A violation of Condition 4.10 prevents the take-off and landing site from being contemporaneous.

First suppose p host switched. Then p' must have speciated after h_1's parent speciation, as h_1 is the landing host and the host switch (and this p''s speciation) must have occurred in the overlap between the take-off and landing hosts (Fig. 4.8). But p' is an ancestor of p^+, so by parasite phylogeny, any event associated with p^+ must succeed the speciation event of p'. But we assumed that p^+ was associated with a host h_2 that speciated before the h_1's parent speciation, so by host phylogeny, such an association must have occurred before p''s speciation – a temporal inconsistency.

So now suppose p did not host switch. Then p' must have speciated on h_1 or cospeciated with h_1's parent speciation (Fig. 4.9). In either case, any event of p^+, a descendent of p', must occur later than the speciation of h_1's parent (by parasite phylogeny). However, we assumed that p^+ is associated with some host h_2 that speciated before h_1's parent speciation, so by host phylogeny, this association must have occurred before h_1's parent speciation – again a temporal inconsistency.

Hence, any violation of Conditions 4.3, 4.9 and 4.10 implied some temporal inconsistency.

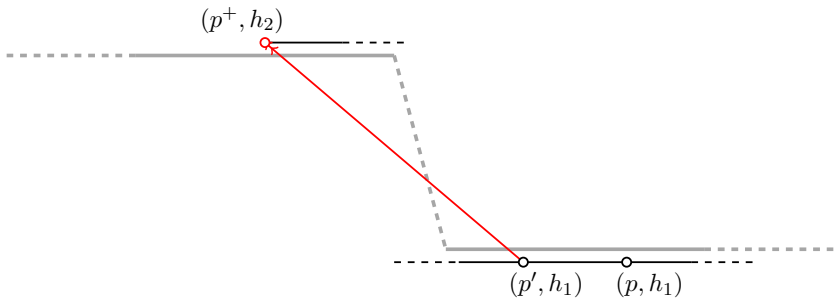


FIGURE 4.9. A violation of Condition 4.10 without a host switch contradicts parasite or host phylogeny implied orderings.

□

4.4.2 Sufficiency

Now we show that Conditions 4.3, 4.9 and 4.10 are sufficient to enforce Conditions 4.5 to 4.7.

DEFINITION (Non-overlap host switch). A host switch is *non-overlapping* if it violates any of Conditions 4.5 to 4.7.

THEOREM 4.7. *If some reconstruction is consistent under Conditions 4.3, 4.9 and 4.10, then for any non-overlapping host, we can move the take-off or landing sites along the take-off or landing hosts to alternate sites that satisfy Conditions 4.5 to 4.7 without creating more non-overlapping host switches.*

PROOF. Conditions 4.3 and 4.9 hold, so the take-off and landing hosts of any host switch overlap at least.

Moving an event along its host edge does not change any host speciation orders enforced by Conditions 4.3, 4.9 and 4.10. Hence we only need to show that moving landing or take-off sites will not create more non-overlapping host switches.

Consider some non-overlapping host switch of parasite p (with parent p') from h_1 to h_2.

CASE 1. Suppose Condition 4.5 does not hold for the host switch. Then the landing site of the host switch precedes the overlap (that is, precedes h_1's parent speciation). We show that it is always possible to move the landing site forward into the overlap.

Suppose that moving the landing site forward would make some other host switch (say of parasite $\bar{p}$) non-overlapping. This can only happen if $\bar{p}$ is a descendent of p and $\bar{p}$ lands on some host h_3 where

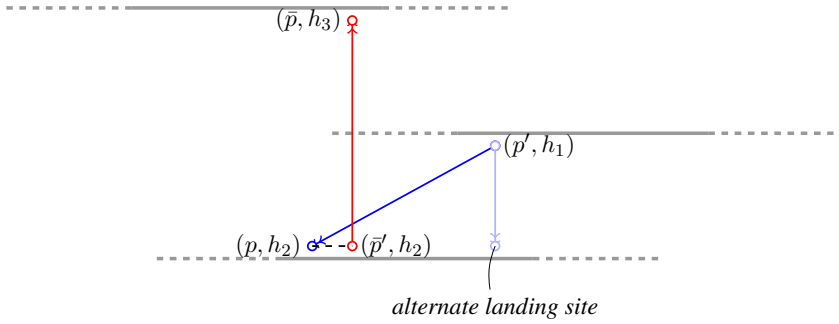


FIGURE 4.10. A non-overlapping host switch (blue) violates Condition 4.5 and is prevented from moving to a non-violating *alternate landing site* by some other hypothetical host switch (red).

h_3 speciates before the parent of h_1 speciates (Fig. 4.10). But $\bar{p}$ is also a descendent of p', so by Condition 4.9, h_1's parent speciation precedes h_3's speciation, a contradiction.

So we can always move the landing site forward along the landing host such that Condition 4.5 holds.

CASE 2. Suppose Condition 4.6 does not hold for the host switch. Then the take-off site of the host switch precedes the overlap (that is, precedes h_2's parent speciation). We show that it is always possible to move the take-off site forward into the overlap.

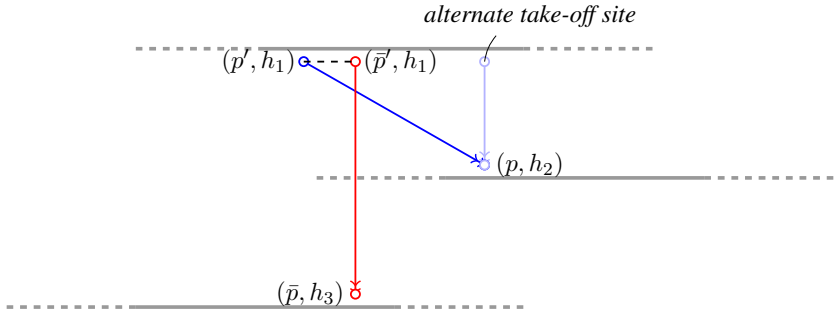
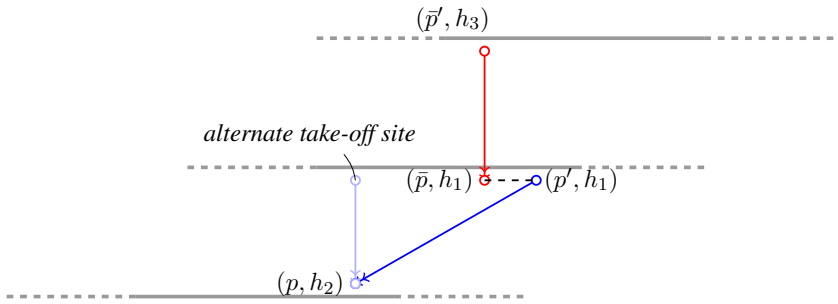


FIGURE 4.11. A non-overlapping host switch (blue) violates Condition 4.6 and is prevented from moving to a non-violating *alternate take-off site* by some other hypothetical host switch (red).

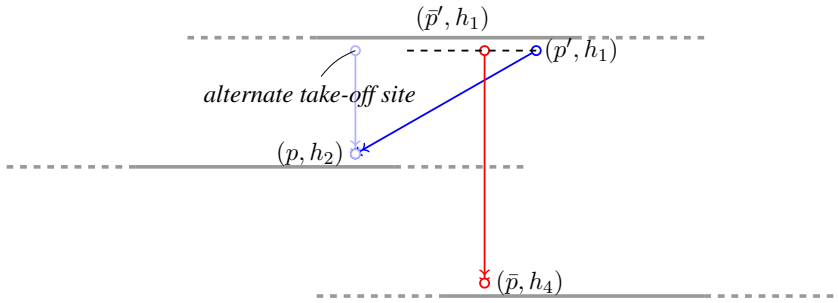
Suppose that moving the take-off site forward would make some other host switch (say of parasite $\bar{p}$) non-overlapping. This can only happen if $\bar{p}$ is a descendent of p' and $\bar{p}$ lands on some host h_3 where h_3 speciates before the parent of h_2 speciates (Fig. 4.11). But $\bar{p}$ is a descendent of p's sibling, so by Condition 4.10 h_3 must speciate after the parent of h_2 speciates, a contradiction.

So we can always move the take-off site forward along the take-off host such that Condition 4.6 holds.

CASE 3. Suppose Condition 4.7 does not hold for the host switch. Then the take-off site of the host switch succeeds the overlap (h_2's speciation precedes the take-off site). We show that it is always possible to move the take-off site backward into the overlap.



(A) Blocking host switch (red) landing site ancestral to non-overlapping host switch (blue) take-off site.



(B) Blocking host switch (red) take-off site ancestral to non-overlapping host switch (blue) take-off site.

FIGURE 4.12. A non-overlapping host switch (blue) violates Condition 4.6 and is prevented from moving to a non-violating *alternate take-off site* by some other hypothetical host switch (red).

Suppose that moving the take-off site backward would make some other host switch (say of parasite $\bar{p}$) non-overlapping. This can only be the case if either $\bar{p}$ is an ancestor of p' or the parent of $\bar{p}$, $\bar{p}'$, is an ancestor of p'.

Let h_3 and h_4 be the take-off and landing sites of $\bar{p}$ respectively.

If $\bar{p}$ is an ancestor of p' and moving back the take-off site of p cause the switch of $\bar{p}$ to become non-overlapping, then h_3's parent must speciate after h_2 (Fig. 4.12a). However, $\bar{p}$ is also an ancestor of p, hence $\bar{p}'$ is an ancestor of p, so by Condition 4.9, h_3's parent must speciate before h_2, a contradiction.

So $\bar{p}$ is not an ancestor of p'. So $\bar{p}'$ must be an ancestor of p' and moving back the take-off site of p cause the switch of $\bar{p}$ to become non-overlapping. So h_4's parent speciation must precede the speciation of h_2

(Fig. 4.12b). But p' is a descendent of $\bar{p}'$, but not $\bar{p}$, so p' must be a descendent of the sibling of $\bar{p}$. So p is also a descendent of the sibling of $\bar{p}$. By Condition 4.10, we must have that h_2 precedes h_4's parent speciation, a contradiction.

So we can always move the take-off site backward along the take-off host such that Condition 4.7 holds.

□

COROLLARY. *Any reconstruction that satisfies Conditions 4.3, 4.9 and 4.10 can be altered without changing event count to also be consistent with Conditions 4.5 to 4.7.*

PROOF. Given any reconstruction that satisfies Conditions 4.3, 4.9 and 4.10, we move the take-off or landing sites of each non-overlapping host switch in turn to satisfy Conditions 4.5 to 4.7.

Theorem 4.7 shows that we can do this independently for each non-overlapping host switch, without introducing or removing events, and without changing any existing host speciation orders. □

Hence we have show that our proposed conditions are both necessary and sufficient to enforce all types of temporal consistency required by a feasible cophylogeny reconstruction.

4.4.3 Complexity

Our proposed conditions only maintaining the partial order of the host speciations. For a tanglegram of n parasites and m hosts, there are m host speciations to consider, and so maintaining the partial order is $O(m^3)$ (the transitivity condition being most expensive).

Both Conditions 4.9 and 4.10 are conditions on pairs of parasite-host associations, and so are $O(m^2n^2)$ to maintain.

Hence, these set of conditions enforce temporal consistency with complexity $O(m^3 + m^2n^2)$ – asymptotically smaller than the $O(m^3n^3)$ complexity of maintaining j-vertex orderings.

4.5 Summary

In cophylogeny reconstruction, correctly accounting for the temporal constraints introduced by host switches is an extremely difficult task. It is possible to avoid this difficulty by defining *relative times* for events to occur in [29]. However, this approach greatly increases the problem size, making it infeasible for certain implementations such as ILP approaches [28].

The j-vertex method provides a set of conditions to correctly detect host switch incompatibilities. Whilst we have proven that these conditions are sufficient, they are very complex to maintain.

Our proposed set of conditions have also been proven to be sufficient in maintaining feasible reconstructions without host switch incompatibilities and has asymptotically smaller complexity than the j-vertex conditions.

In the next chapter, we propose a new ILP formulation of the cophylogeny reconstruction problem using the proposed condition set described in this chapter. In Chapter 6, we demonstrate that a reference implementation of the new formulation is efficient enough to exactly solve the reconstruction problem on real-life sized datasets.

CHAPTER 5

Integer Linear Program Formulation

In this chapter we review a previous ILP formulation and propose a new ILP formulation for the cophylogeny reconstruction problem. In Section 5.1 we examine the ILP proposed by Libeskind-Hadas and Charleston [28] and identify the weaknesses of the formulation that cause it to be impractical to implement and use for real life datasets. We then propose an alternate formulation in Section 5.2 and prove the correctness of the formulation in Section 5.3.

In Chapter 6 we describe a reference implementation of our proposed ILP and evaluate it against a reference implementation of the previous ILP formulation as well as the latest version of the JANE heuristic.

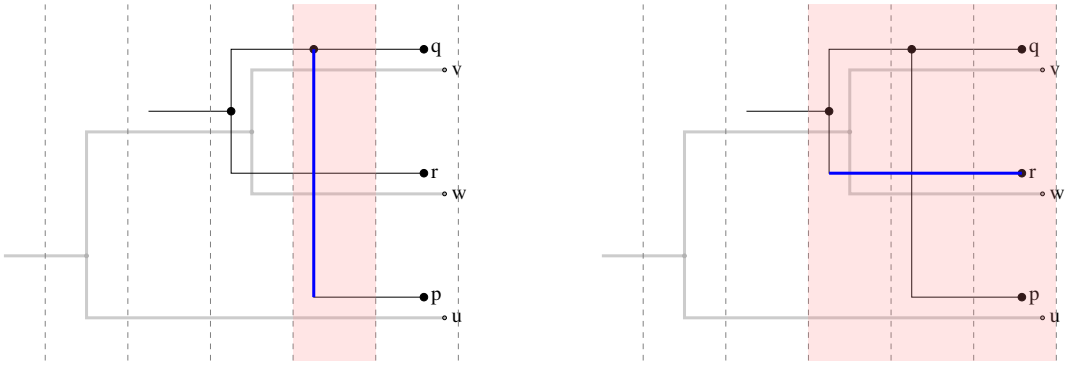
5.1 Previous ILP Formulation

There has been previous work on an ILP formulation of the cophylogeny reconstruction problem by Libeskind-Hadas and Charleston [28]. The existing formulation was the basis for the "fixed event time" approach of reconstruction discussed in Libeskind-Hadas and Charleston [29] and used in the JANE heuristic [15]. Whilst the ideas in the formulation were influential in establishing a novel approach in reconstruction, the ILP formulation itself was far too cumbersome for practical implementation, as admitted by the authors [28].

5.1.1 Fixed Event Times

The novel contribution of the formulation is to consider the explicit time in which a parasite and host are associated. Whilst the associations are continuous, in a sense, it is fully determined by a number of discrete events – it is the order of the events that characterises a reconstruction, not the exact timing of the associations.

The maximum number of distinct event times to unambiguously represent a reconstruction is equal to the total number of host and parasite speciations in the input tanglegram – each parasite speciation can generate at most a single distinct event and each host speciation may need to be distinct from any other event (in the case of parasite duplication and host switching) [28]. The host and parasite trees are then distributed across these "time zones" and associations can only occur within a single zone.



(A) Host switch within a single time zone

(B) Verbose associations spanning time zones

FIGURE 5.1. Event time based reconstruction – events are distributed between time zones.

This approach has the advantage of preventing host switch incompatibilities from occurring. Host switches can only occur between hosts that occupy the same time zone, hence are guaranteed to be contemporaneous (Fig. 5.1a). The take-off and landing sites must then occur within the same time zone and so can only be resolved as simultaneous events.

However, this type of representation of the parasite-host associations is extremely verbose. Each parasite-host association needs to be represented for each time zone the association spans (Fig. 5.1b). Extra constraints are required to ensure that these redundant associations remain consistent.

The distribution of the parasite and host phylogenies into the time zones suffers from the same redundancy. Rather than representing a species by its speciation event (a node in a tree) and relating the

species with its children and parent, a species needs to be represented once for each time zone it spans. Constraints are then required to ensure that the time zones spanned by a species is contiguous.

This level of verbosity is inappropriate for practical ILP implementations. The time required to find general solutions to ILPs is exponential in the number of variables and constraints. General optimisation methods do not necessarily exist to remove the redundancy introduced by the ILP formulation.

5.1.1.1 Errors

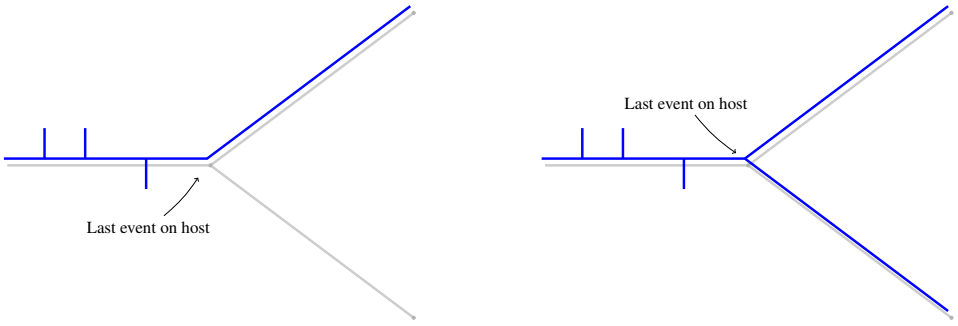
There were a number of logical errors in the formulation presented in Libeskind-Hadas and Charleston [28], corrections for which are provided in Appendix A.

5.2 Proposed ILP Formulation

Our proposed ILP formulation is based on the classic "event mapping" methods of cophylogeny reconstruction. However, concepts from the Libeskind-Hadas and Charleston formulation are also included when advantageous. In constructing the proposed ILP formulation, the focus was on reducing redundancy in the variables and constraints used.

Like previous "mapping" methods, we do not explicitly consider the time at which a parasite-host association occurs – a parasite is simply associated with a host without explicit regards to *where* (or when) on the host edge the association began. Furthermore, our formulation takes this one step further, not distinguishing between associations on host edges and host nodes (host speciations). This distinction is made in many mapping methods [6, 30] but is a redundancy. The only situation when an event is associated with a host node is cospeciation or loss. In either case, only the last event on a given host in each parasite lineage can be associated with the host speciation. In fact, the last event on a given host in each parasite lineage must be a cospeciation event, loss event, or a leaf mapping. So more strongly, an event is associated with a host speciation *if and only if* it is the last event on the host for any parasite lineage (Fig. 5.2).

However, in contrast to mapping methods, and borrowing from the Libeskind-Hadas and Charleston formulation, we explicitly record every host with which a parasite is associated. Mapping methods tend to only associate parasite speciations events, hence each parasite is mapped to a single host. The disadvantage of this method is that the landing site of any host switch is not explicit, allowing weak



(A) Last association is a loss event

(B) Last association is a cospecation event

FIGURE 5.2. The last association on a host is a node associated events.

incompatibilities to occur. These incompatibilities require *a posteriori* resolution to form a feasible reconstruction – an NP-hard problem in itself. This is undesirable in an ILP formulation, the constraints of which should model precisely the feasible solution space.

Finally, the set of conditions proposed in Section 4.4 are used to ensure timing compatibility. This reduces the required variables from cubic in the total number of species to quadratic.

5.2.1 Model Assumptions

Our formulation applies to the non-reticulate (tree) and non-widespread taxa event costing version of the problem. Under this scheme, we consider only tanglegrams of two complete binary phylogeny trees with surjective leaf mappings from parasite tips to host tips. Costs for the four permissible events under the non-widespread model are also given: cospeciation, duplication, host switching, and loss/sorting. A more formal and complete explanation of the model and assumptions can be found in Chapter 2. Our solution assumes cospeciation costs 0 and other events costs are positive.

5.2.2 ILP Formulation

5.2.2.1 Notation

Let p be a node in a full binary tree. Then p' is the parent of p, p^+ is the sibling or p, and, without loss of generality, p_L and p_R are the left and right children of p.

5.2.2.2 Known Variable Descriptions

The following are the known variables that are used in the ILP description. All of these variables can be calculated in polynomial time (in fact, no more than $O(n^2)$ time) given the input tanglegram.

H**:** HOSTS :

Set of host species.

P**:** PARASITES :

Set of parasite species.

$H_{\mathcal{L}}$**:** HOST LEAVES :

Set of host species that are leaves in the host phylogeny (extent taxa).

$P_{\mathcal{L}}$**:** PARASITE LEAVES :

Set of parasite species that are leaves in the parasite phylogeny (extent taxa).

p_0**:** PARASITE ROOT :

Root species of the parasite phylogeny.

$\preceq_H$**:** HOST PARTIAL ORDER :

Strict partial order of host speciations implied by the host phylogeny.

$(h_1, h_2) \in \preceq_H$ iff h_1 is an ancestor of h_2.

$\preceq_P$**:** PARASITE PARTIAL ORDER :

Strict partial order of parasite speciations implied by the parasite phylogeny.

$(p_1, p_2) \in \preceq_P$ iff p_1 is an ancestor of p_2.

$\not\sim_H$**:** UNRELATED HOST SPECIES :

Set of host speciation pairs that are incomparable by $\preceq_H$.

$(h_1, h_2) \in \not\sim_H$ iff $h_1 \neq h_2$ and $(h_1, h_2) \notin \preceq_H$ and $(h_2, h_1) \notin \preceq_H$.

φ**:** LEAF MAP :

Map of extent taxa - $\varphi : P_{\mathcal{L}} \to H_{\mathcal{L}}$.

$\varphi(p) = h$ iff extent parasite p is associated with extend host h.

5.2.2.3 Decision Variable Descriptions

The following are the decision variables of the ILP. All the decision variables are boolean.

$\ll_{h_1,h_2}$**:** STRICT TOTAL ORDERING :

Variable $\ll_{h_1,h_2}$ is true iff host h_1 speciated before host h_2 in the *a posteriori* strict total ordering of host speciation events.

$\Phi_{p,h}$**:** MAPPING :

Variable $\Phi_{p,h}$ is true iff parasite p is associated with host h at some time.

$\mathcal{X}_{p,h_1,h_2}$**:** HOST SWITCH :

Variables $\mathcal{X}_{p,h_1,h_2}$ is true iff parasite p host switched, the take-off site being on h_1 and landing site being on h_2.

$\mathcal{C}_{p,h}$**:** COSPECIATION :

Variable $\mathcal{C}_{p,h}$ is true iff p and h cospeciated.

5.2.2.4 Objective Function

We count the events in a way that is compatible with TREEMAP, TREEMAP 2, and JANE, where each point of cospeciation or duplication counts for two events (see Section 2.2.1.6). Under this scheme, the number of cospeciation and duplication events must add up to the number of non-root parasite nodes (see Section 5.3.3.2). Event counts are given by the following:

Cospeciation: $\#C = 2\sum \mathcal{C}_{p,h}$
Duplication: $\#D = |P| - 1 - \#C$
Host Switch: $\#H = \sum \mathcal{X}_{p,h_1,h_2}$
Loss/Sorting: $\#L = -|P| + \sum \Phi_{p,h}$

Then the objective function is given by:

$$min \quad Cost_{dup} \times \#D + Cost_{switch} \times \#H + Cost_{loss} \times \#L \tag{5.1}$$

5.2.2.5 Constraint Descriptions

We describe the linear constraints in terms of logical formulas. Systematic methods can be used to convert these logical formulas into Conjuctive Normal Form (CNF) which can then be converted to integer linear constraints, which are listed in Appendix B.

Host Total Order Constraints

The following constraints ensure that the variable $\ll_{h_1,h_2}$ is mathematically a strict total order on $h \in H$.

CONSTRAINT 5.1 (Host Order Totality).
The ordering of host speciation events must be total. That is, for any two distinct host speciations, one must have occured strictly before the other.

$$\ll_{h_1,h_2} \veebar \ll_{h_2,h_1} \qquad \forall h_1 \neq h_2 \in H$$

CONSTRAINT 5.2 (Host Order Strictness).
The ordering of the host speciation events must be strict. That is, a host speciation event is not considered to be ordered before itself.

$$\neg \ll_{h,h} \qquad \forall h \in H$$

CONSTRAINT 5.3 (Host Order Transitivity).
The ordering of the host speciation events must be strictly transitive.

$$\ll_{h_1,h_2} \wedge \ll_{h_2,h_3} \Rightarrow \ll_{h_1,h_3} \qquad \forall h_1 \neq h_2 \neq h_3 \in H$$

Temporal Constraints

The following constraints enforce temporal compatibility in the reconstruction (see Section 4.2). We use the compact set of temporal constraints proposed in Section 4.4.

CONSTRAINT 5.4 (Host Phylogeny Implicit Ordering).
The total ordering of host speciations must be fully compatible with the host phylogeny implict ordering.

$$\ll_{h_1,h_2} \qquad \forall (h_1, h_2) \in \preceq_H$$

CONSTRAINT 5.5 (Strong Incompatibility).

If a parasite and its descendent are mapped to two hosts, then the two hosts cannot be reverse ordered (see Condition 4.9).

$$(\Phi_{p_1,h_1} \wedge \Phi_{p_2,h_2}) \Rightarrow \ll_{{h_1}',h_2} \qquad \forall\, (p_1, p_2) \in \preceq_P, h_1, h_2 \in H$$

CONSTRAINT 5.6 (Weak Incompatibility).

The take-off and landing hosts of any host switch must overlap (see Condition 4.4).

$$\mathcal{X}_{p,h_1,h_2} \Rightarrow \ll_{{h_2}',h_1} \qquad \forall p \in P, (h_1, h_2) \in \not\sim_H$$

CONSTRAINT 5.7 (Sibling Weak Incompatibility).

If a parasite and a descendent of the parasite's sibling are mapped to two hosts, then the hosts cannot be reverse ordered (see Condition 4.10).

$$(\Phi_{p_1,h_1} \wedge \Phi_{p_2,h_2}) \Rightarrow \ll_{{h_1}',h_2} \qquad \forall p_1 \in P, \left({p_1}^+, p_2\right) \in \preceq_P, h_1, h_2 \in H$$

Mapping Constraints

The following constraints ensure some necessary conditions on the set of associations in a feasible reconstruction.

CONSTRAINT 5.8 (Leaf Mapping).

The associations between leaf parasites and hosts must preserve the input leaf mapping.

$$\Phi_{p,h} \qquad \forall \varphi\,(p) = h$$

CONSTRAINT 5.9 (Non-Widespread Mapping).

Any single parasite species can only be associated with a single lineage of the host phylogeny. That is, there cannot be widespread parasites.

Logically, no parasite can be associated with two unrelated hosts.

$$\neg\,(\Phi_{p,h_1} \wedge \Phi_{p,h_2}) \qquad \forall p \in P, (h_1, h_2) \in \not\sim_H$$

CONSTRAINT 5.10 (Root Mapping).

The root parasite must be associated with a single origin host.

Logically, if the root parasite maps to a host, it must either map to the parent of the host or only be mapped to descendent of the host.

$$\Phi_{p_0,h} \Rightarrow \Phi_{p_0,h'} \vee \neg \left(\bigvee_{u \neq h : (u,h) \in \preceq_H} \Phi_{p_0,u} \right) \qquad \forall h \in H$$

Host Switch Definition

The following constraints define the non-temporal properties of a host switch (see Section 4.1.1).

CONSTRAINT 5.11 (Take-off Parent).

By definition, the parent of a host switching parasite must be mapped to the take-off host.

$$\mathcal{X}_{p,h_1,h_2} \Rightarrow \Phi_{p',h_1} \qquad \forall p \in P, (h_1,h_2) \in \not\sim_H$$

CONSTRAINT 5.12 (Landing Site).

By definition, the host switching parasite must be mapped to the landing host.

$$\mathcal{X}_{p,h_1,h_2} \Rightarrow \Phi_{p,h_2} \qquad \forall p \in P, (h_1,h_2) \in \not\sim_H$$

CONSTRAINT 5.13 (Take-off Sibling).

The sibling of a host switching parasite must be mapped to the take-off host. This is also a part of the resolvability condition, requiring the host switch event to be distinct from the speciation of the take-off host (see Section 2.2.2.2).

$$\Phi_{p,h} \wedge \Phi_{p',h} \Rightarrow \Phi_{p^+,h} \vee \left(\bigvee_{u : (u,h) \in \not\sim_H} \mathcal{X}_{p,u,h} \right) \qquad \forall p \in P, h \in H$$

Cospeciation Definition

The following constraint defines a cospeciation event.

CONSTRAINT 5.14 (Cospeciation).
A cospeciation occurs when a parasite is mapped to a host and both children of the parasite are mapped to both children of the host.

$$\mathcal{C}_{p,h} \Rightarrow \Phi_{p,h} \wedge ((\Phi_{p_L,h_L} \wedge \Phi_{p_R,h_R}) \vee (\Phi_{p_L,h_R} \wedge \Phi_{p_R,h_L})) \qquad \forall p \in P - P_{\mathcal{L}}, h \in H - H_{\mathcal{L}}$$

Logically, we should require the constraint to be *if and only if* rather than just an *implies*. However, the objective function always rewards increase in cospeciation variables, hence it is unnecessary to require conditions implying cospeciation variables.

Duplication Definition

The following constraints define a duplication event. In terms of this formulation, a duplication occurs when a parasite and its child are both mapped to a single host – this implies that the parent parasite speciated on the host edge, hence did not cospeciate with the host.

CONSTRAINT 5.15 (Duplication Sibling).
If a parasite and its parent are both associated with a single host, then a duplication occurred. Hence, the sibling must either host switch or also associate with the parasite.

$$\Phi_{p,h} \wedge \Phi_{p',h} \Rightarrow \Phi_{p^+,h} \vee \left(\bigvee_{u:(u,h)\in\gamma_H} \mathcal{X}_{p,u,h} \right) \qquad \forall p \in P, h \in H$$

CONSTRAINT 5.16 (Parallel Siblings).
If sibling parasites both map to a single host, then the common parent must have duplicated. If the duplication occurred on the current host, then the parent must also be mapped to the host. Otherwise, the duplication occurred on some ancestor of the current host, hence both siblings must also map to the parent of the current host.
Logically, if sibling parasites are mapped on the same host, then either the parent parasite is also mapped to the host or both sibling parasites are mapped to the parent host.

$$\Phi_{p,h} \wedge \Phi_{p^+,h} \Rightarrow \Phi_{p',h} \vee \left(\Phi_{p,h'} \wedge \Phi_{p^+,h'} \right) \qquad \forall p \in P, h \in H$$

Traceability

The following constraint is the main traceability condition (see Section 2.2.2.3).

CONSTRAINT 5.17 (Traceability).
If a parasite maps onto a host, then the parasite must map onto a child of the host, or a child of the parasite must map onto the host or a child of the host.

$$\Phi_{p,h} \Rightarrow \Phi_{p,h_L} \vee \Phi_{p,h_R} \vee \Phi_{p_L,h} \vee \Phi_{p_L,h} \vee \Phi_{p_L,h_L} \vee \Phi_{p_L,h_R} \vee \Phi_{p_R,h_L} \vee \Phi_{p_R,h_R} \qquad \forall p \in P, h \in H$$

Continuity

The following constraint is the continuity condition – every association must be the result of some previous event (see Section 2.2.2.4).

CONSTRAINT 5.18 (Continuity).
If a parasite is mapped to a host, then either the parasite or the parasite's parent is mapped to the host's parent, the parasite's parent is mapped to the host, or the parasite host switched onto the host.

$$\Phi_{p,h} \Rightarrow \Phi_{p',h} \vee \Phi_{p,h'} \vee \Phi_{p',h'} \vee \left(\bigvee_{u:(u,h)\in\nsim_H} \mathcal{X}_{p,u,h} \right) \qquad \forall p \in P, h \in H$$

5.3 Formulation Correctness

We now prove that all the constraints are both sufficient and necessary, and hence the ILP formulation is an exact solution of the cophylogeny reconstruction problem.

5.3.1 Necessity

We prove the necessity of each of the constraints in turn by proving that a violation of any constraint results in a violation of some basic conditions on the feasibility of a reconstruction, as discuss in Chapter 2.

Host Total Order Constraints and Temporal Constraints

The *host total order constraints* and *temporal constraints* are restatements of Conditions 4.3, 4.9 and 4.10 and have been show to be necessary for temporal compatibility (see Section 4.4.1).

Mapping Constraints

Constraint 5.8 is required by the problem definition – the reconstruction must contain the input leaf mapping.

We now show that Constraint 5.9 is necessary.

THEOREM 5.1. *Any violation of Constraint 5.9 results in widespread parasites.*

PROOF. Let $(h_1, h_2) \in \not\sim_H$ and let p be associated with both h_1 and h_2. If h_1 and h_2 are contemporaneous, then we are done – p is clearly widespread on h_1 and h_2. So without loss of generality, assume that h_1 existed strictly before h_2 – that is, assume that h_1 speciated before the parent of h_2, ${h_2}'$ speciated.

${h_2}'$ cannot be an ancestor of h_1 as h_1 speciated before ${h_2}'$. h_1 cannot be an ancestor of ${h_2}'$ otherwise h_1 is an ancestor of h_2, contradicting the assumption that $(h_1, h_2) \in \not\sim_H$. Hence $\left(h_1, {h_2}'\right) \in \not\sim_H$.

If p host switched, it could have only host switched to the earliest host associated with p (by definition of host switching). p is associated with h_1 which exists strictly earlier than h_2, so p could not have host switched to h_2.

Now suppose p is not associated with ${h_2}'$, p did not host switch to h_2, so by continuity, p' must be associated with ${h_2}'$ or h_2. But p' cannot be associated with h_2 as p is associated with h_1, causing a temporal incompatibility (Condition 4.9).

So p' must have cospeciated with ${h_2}'$. But, h_1 is not a descendent of ${h_2}'$, so by continuity, p must have host switched to h_1 (Section 2.2.2.4). But this violates resolvability as p' cospeciated (Section 2.2.2.2).

Hence p must be associated with ${h_2}'$. Thus, we can then repeat this argument with $h_2 := {h_2}'$. We must eventually have h_1 and h_2 being contemporaneous, hence p is widespread. □

We also show that Constraint 5.10 is necessary.

THEOREM 5.2. *Any violation of Constraint 5.10 results in the root parasite being widespread or a violation of continuity (Section 2.2.2.4).*

PROOF. Let p_0 be the root of the parasite phylogeny and suppose Constraint 5.10 is violated. So p_0 is associated with some host h_1 but is not associated with ${h_1}'$ but *is* associated with some other host h_2 that is not a descendent of h_1.

If h_2 is not an ancestor of h_1, then $(h_1, h_2) \in \not\sim_H$. By Theorem 5.1, p must be widespread.

So h_2 is an ancestor of h_1. But then h_2 existed at an earlier than h_1, hence h_1 is not the earliest associate of p_0. So by *continuity*, p_0 must be associated with ${h_1}'$, a contradiction. □

Host Switch Definition

Constraints 5.11 and 5.12 define the take-off and landing sites of a host switch and hence are necessary. Hence Constraint 5.13 is necessary.

THEOREM 5.3. *A violation of Constraint 5.13 violates resolvability (Section 2.2.2.2).*

PROOF. Let p be a parasite host switching from h_1 to h_2 that violates Constraint 5.13. Let p' be the parent of p and p^+ be the sibling of p. So p' is associated with h_1 but p^+ is not.

By traceability, p^+ must be associated with a child of h_2, say h_3. p' speciated on h_2 (by definition of the take-off site of p), hence cannot be associated with h_3. But this implies that p' cospeciated with h_2, violating resolvability. □

Duplication Definition

THEOREM 5.4. *A violation of Constraint 5.15 violates resolvability (Section 2.2.2.2).*

PROOF. Suppose p duplicates on host h and let p_L and p_R be the children of p. Further, suppose Constraint 5.15 is violated by this duplication. So, without loss of generality, p_L is mapped to h, but p_R is not mapped to h and does not host switch.

Then by continuity, p_R must be associated with a child of h, say h_2. But p speciated on h, and so cannot be associated with h_2, hence p must have cospeciated. But this violates the required resolvability of p's duplication. □

Hence Constraint 5.15 is necessary.

THEOREM 5.5. *A violation of Constraint 5.16 violates resolvability (Sections 2.2.2.2 and 2.2.2.3).*

PROOF. Suppose p_L and p_R are siblings associated with some host h and that Constraint 5.16 was violated. So the common parent of p_L and p_R, p, does not associate with h. c First suppose, without loss of generality, p_L is associated with h', but p_R does not.

If p_R did not host switch, then by continuity, p must be associated with h'. But by Theorem 5.4, this would violate resolvability.

Hence, p_R must have host switched. Let h_2 be the take-off site of the host switch. So p' is associated with h_2. h_2 must be unrelated to h, and so is either unrelated to h' or is a descendent of h'.

If h_2 is unrelated to h', then $tchlp$ must have also host switched, but this violates the traceability of p. Hence h_2 is a descendent of h'. But p_L is associated with h' and p' is associated with h_2, creating a temporal incompatibility (violates Condition 4.9).

Hence it must be the case that neither p_L nor p_R is associated to the parent of h, h'. Then by continuity, either p maps to h', or both p_L and p_R host switched. If p_L and p_R both host switched, p would be untraceable. Hence p maps to h'. But by assumption, p did not associate with h, so p cospeciated on h'. Both children of p, p_L and p_R, associated with only a single child of h', hence p duplicated. This creates a violation of the resolvability condition for duplications. □

Cospeciation Definition

Constraint 5.14 is just the definition of a cospeciation event.

Traceability

Constraint 5.17 is just a logical restatement of the traceability condition (see Section 2.2.2.3).

Continuity

Constraint 5.18 is just a logical restatement of the continuity condition for the four event model (see Section 2.2.2.4)

5.3.2 Sufficiency

To prove sufficiency, we consider each assumption made in Chapter 2 and show how any reconstruction that is infeasible under these assumptions must violate an ILP constraint.

Temporal Compatibility

The *host total order constraints* and *temporal constraints* are logical formulations of Conditions 4.3, 4.9 and 4.10 and have been show to be sufficient for temporal compatibility (see Section 4.4.2).

Traceability

Constraint 5.17 is just a logical restatement of the traceability condition, hence is sufficient.

Continuity

Constraint 5.18 is just a logical restatement of the continuity condition for the four event model, hence is sufficient.

Non-Widespread Parasites

We show that any reconstruction postulating widespread parasites will contradict Constraint 5.9.

THEOREM 5.6. *If a parasite p is associated with two distinct hosts h_1 and h_2 concurrently, then p violates Constraint 5.9.*

PROOF. p is associated both h_1 and h_2 concurrently, so h_1 and h_2 are contemporaneous. Thus, h_1 cannot be an ancestor of h_2 and h_2 cannot be an ancestor of h_1. Hence h_1 is not related to h_2, so $(h_1, h_2) \in \not\sim_H$. But p is associated with both h_1 and h_2, contradicting Constraint 5.9. □

Resolvability - Host Switches

We show that any reconstruction postulating unresolved host switches – that is, hosts switches that are simultaneous with the take-off host speciation – must violate Constraint 5.13.

THEOREM 5.7. *If parasite p host switches from host h_1 to h_2 but the parent of p, p', cospeciated with h_1, then Constraint 5.13 must be violated.*

PROOF. We assume that p' cospeciated with h_1, so any child of p' cannot be associated with h_1. Specifically, the sibling of p, p^+ cannot be associated with h_1, contradicting Constraint 5.13. □

Resolvability - Duplications

We now show that any reconstruction postulating unresolved duplications – that is, a duplication on a host that is simultaneous with the host speciation (but not a proper cospeciation) – must violate Constraint 5.16.

THEOREM 5.8. *If a parasite p cospeciates with h, but both children of p, p_L and p_R, are associated with only one child of h, say h_L, then Constraint 5.16 must be violated.*

PROOF. Both p_L and p_R are associated with h_L. By Constraint 5.16, we require either both p_L and p_R be associated with h, or p be associated with h_L. But p cospeciated with h, so cannot be mapped to any child of h. Also, any child of p cannot be mapped to h. Hence Constraint 5.16 cannot be satisfied. □

5.3.3 Event Counts

We now show that the event counts are correct.

5.3.3.1 Cospeciation

The number of cospeciation events, $\#C$, counted in the objective function is given by

$$\#C = 2\sum \mathcal{C}_{p,h} \tag{5.2}$$

where $\mathcal{C}_{p,h}$ is a decision variable of the ILP constrained only by Constraint 5.14. There are two possible sources of miscounts in this formulation:

(1) If a parasite p duplicates on h and both children of h are associated with children of p, then Constraint 5.14 will allow $\mathcal{C}_{p,h}$ to be TRUE.
(2) If a parasite p does cospeciate with a host h, Constraint 5.14 only implies that $\mathcal{C}_{p,h}$ *can* be TRUE, not that it must be TRUE.

However, we show that both these cases are not possible in the optimal reconstruction.

THEOREM 5.9. *If a feasible solution of the ILP postulates a parasite p which duplicates on h and both children of h are associated with children of p, and $\mathcal{C}_{p,h}$ is* TRUE*, then the solution is not optimal.*

PROOF. Consider a new solution in which both p_L and p_R are not associated with h. This solution implies the cospeciation of p on h. Since p_L and p_R are both associated with children of h, this does not affect the speciation events of p_L and p_R, and hence nothing else in the solution need change to remain feasible. But in this new solution $\#L$ would decrease by 2 whilst all other counts remain the same. $\#L$ has a positive coefficient in the objective function, so the new solution has a lower objective value than the original solution. Hence the original solution could not have been optimal. □

THEOREM 5.10. *If a feasible solution of the ILP postulates a parasite p cospeciating with host h but $\mathcal{C}_{p,h}$ is* FALSE*, then the solution is not optimal.*

PROOF. Consider a new solution in which $\mathcal{C}_{p,h}$ is TRUE. The only constraint that might be affected is Constraint 5.14. But we assumed that p did cospeciate with h, so Constraint 5.14 is still satisfied. Hence all other variables can remain the same and the new solution is still feasible.

In the new solution, $\#C$ increases by 2, and all other event counts remain the same. $\#C$ has a negative coefficient in the objective function, so the new solution has a lower objective value than the original solution. Hence the original solution could not have been optimal. □

5.3.3.2 Duplication

The number of duplication events, $\#D$, counted in the objective function is given by

$$\#D = |P| - 1 - \#C \tag{5.3}$$

We have shown in Section 5.3.3.1 that $\#C$ is a correct count of the cospeciation events in the optimal reconstruction. Hence, to show that $\#D$ is the correct count of duplication events, we need to show:

THEOREM 5.11. *For a feasible reconstruction where $\#D$ and $\#C$ are the number of duplication and cospeciation events respectively, and $\#P$ is the number of parasites:*

$$\#D + \#C = \#P - 1$$

PROOF. Each parasite in the parasite phylogeny, apart from the root, must have a parent parasite. Hence we can associate each non-root parasite with the speciation event type of the parasite's parent. The speciation of each such parent is responsible for either 2 duplication or 2 cospeciation events and has exactly two children, hence there is a one-to-one correspondence between non-root parasites and duplication and cospeciation events. □

5.3.3.3 Host Switches

The number of host switch events, $\#H$, counted in the objective function is given by

$$\#H = \sum \mathcal{X}_{p,h_1,h_2} \tag{5.4}$$

It is clear from Constraints 5.11 and 5.12 that any $\mathcal{X}_{p,h_1,h_2}$ that is TRUE must define a host switch. Conversely, if there is a host switch in the reconstruction, then Constraint 5.18 (continuity condition) requires that some corresponding $\mathcal{X}_{p,h_1,h_2}$ be TRUE. Hence $\#H$ is a correct count of host switch events.

5.3.3.4 Loss

The number of loss events, $\#L$, counted in the objective function is given by

$$\#L = -|P| + \sum \Phi_{p,h} \tag{5.5}$$

It is clear that losses occur when a parasite is associated with more than one host. In fact, the number of sorting events associated with a parasite is exactly the number of hosts the parasite is associated with after the first. Summing this across all parasite and we get the formula for $\#L$. Hence loss events are correctly counted.

5.4 Variable and Constraint Counts

5.5 Summary

The existing ILP formulation [28], whilst novel, is not a sufficiently compact representation of the cophylogeny reconstruction problem to provide a practical exact method. However, taking ideas from Libeskind-Hadas and Charleston's work and combining it with established approaches, we formulate a new ILP with far fewer variables and constraints. This was achieved by building on the concise temporal constraints proposed in Section 4.4.

In the next chapter, we evaluate a reference implementation of the proposed ILP, comparing it to the Libeskind-Hadas and Charleston formulation.

CHAPTER 6

Reference Implementation

In this chapter we describe and evaluate a reference implementation of the ILP proposed in Section 5.2. We compare the results of our new formulation against the Libeskind-Hadas and Charleston [28] ILP formulation as well as the most recent release of the novel JANE heuristic [15], JANE 3 [2].

6.1 Implementations

6.1.1 JANE

The latest version of the JANE heuristic implementation, JANE 3 [2], was used for comparison.

6.1.2 Proposed ILP

The proposed ILP was implemented using a subset of the IBM ILOG CPLEX Optimization Studio[1]. The ILP was modelled using the Optimization Programming Language (OPL) and solved using the CPLEX Optimizer. The exact objective function and constraints used are described in Appendix B. These constraints are a direct translation of the logical formulae described in Section 5.2.2.

A Python script was written to translate input file formats, including the `.tree` and `.nex` file formats, into OPL descriptions of the input tanglegrams. OPL Script was then used to preprocess the tanglegrams into the required variables (see Section 5.2.2.2) and to postprocess the decision variables into a suitable output file format.

The entire process was tied together with a shell script. This formed the *reference implementation* of the proposed ILP.

[1] `Version IBM ILOG OPL 6.3, IBM ILOG CPLEX 12.1.0, IBM ILOG CP Optimizer 2.3`

6.1.3 Previous ILP

The Libeskind-Hadas and Charleston [28] ILP was also implemented using the same procedures and technology as our proposed ILP. This allowed a fair comparison to be made between the two formulations – any differences between the two ILP implementations will be due to the formulation of the ILPs rather than implementation differences.

The previous ILP was modelled to use the same input variables as the proposed ILP and postprocessed to produce the same output. Hence, the file format translation, preprocessing, and shell scripting could be completely shared between the two methods, further reducing possible implementation differences.

The constraints and objective function modelled are taken directly from Libeskind-Hadas and Charleston [28], but with the corrections listed in Appendix A.

6.2 Experiments

The three implementations were run across a number of datasets, both synthetic and real world.

The cost of the optimal reconstruction and the wall clock time (as reported by the UNIX `time` utility) were recorded for each implementation on each input tanglegram.

All experiments were run using JANE's default cost scheme (Loss 2, Duplication 1, Host Switch 1, and Cospeciation 0). JANE was run using default parameters, including defaults for population size (30) and generation limit (30). The two ILP implementations were run with default CPLEX settings and without any adjustments of solver parameters.

6.2.1 Platform

The experiments were run on a three year old (2008) commodity desktop computer. The computer had 3.8GB of system memory and contained an Intel® Core™ 2 Duo Processor E8400 (released in Q1 2008), running two cores at 3.0GHz. The computer ran Ubuntu 11.04 (Natty Narwhal) 64-bit, a Debian based Linux distribution.

6.2.2 Datasets

The datasets used in the benchmark experiments are split into two broad categories. *Synthetic* datasets were used to control the size of input tanglegrams and are used to give some idea of performance degradation as the dimensions of the input tanglegrams are altered. *Real world* datasets were used to benchmark the methods on actual experimental data to assess the practicality of the implementations.

We adopt the notation $h \times p$ to describe the dimension of a tanglegram with h hosts and p parasites.

6.2.2.1 Synthetic Datasets

The synthetic datasets were produced by constructing random tanglegrams of some given dimensions. The tanglegrams were produced by a Python script that generated random trees from a uniform distribution of structurally different full binary trees of a given leaf count. A random surjective map of the leaf sets is also generated to complete the tanglegram.

Small Dataset

This dataset consists of very small tanglegrams, with 10 random tanglegrams of each of the following dimensions:

- 4×4
- 5×5
- 6×6
- 7×7
- 8×8
- 9×9

Balanced Dataset

This dataset consists of large tanglegrams with balanced dimensions (same number of hosts leaves as parasite leaves). It contains 10 random tanglegrams of each of the following dimensions:

- 5×5
- 10×10

- 15×15
- 20×20
- 25×25

Unbalanced Dataset

This dataset consists of large tanglegrams with unbalanced dimensions (more parasite leaves than host leaves). It contains 10 random tanglegrams of each of the following dimensions:

- 15×20
- 15×25
- 15×30
- 20×25
- 20×30
- 20×35
- 20×40

6.2.2.2 Real World Datasets

The real world datasets are taken from a number of studies requiring cophylogeny reconstruction.

Butterflies Dataset

This dataset is the full dataset from Hoyal Cuthill and Charleston [23], a study of feature mimicry between butterflies. The dataset contains some 30 tanglegrams of dimensions between 8×8 and 12×12.

Assorted Datasets

This dataset contains a collection of tanglegrams draw from a variety of studies, some of which were used in Conow et al. [15] to benchmark the JANE heuristic.

gopher: is the well studied pocket gopher and chewing lice tanglegram [21].

seabird: is a tanglegram of seabirds and lice [43].

pinworm: is a tanglegram of *Hystricognath* rodent and pinworms [25].

stinkbugs: is a tanglegram of stinkbugs and gut symbiotic bacteria [27].

ficus: is a tanglegram of *Ficus* plants and *Ceratosolen* wasp pollinators [53].

pelecan: is a tanglegram of *Pelecaniform* birds and *Pectinopygus* lice [24].

fungus: is a tanglegram of *Caryophyllaceous* plants and fungi [45].

vidua: is a tanglegram of *Estrildae* finches and *Vidua* finches (brood parasites) [51].

hanta: is a tanglegram of rodents and hatavirus [44].

6.3 Results

The full results tables for all datasets are found in Appendix C.

6.3.1 Synthetic Datasets

6.3.1.1 Small Dataset

All three implementations were run across the small dataset.

	Previous ILP			Proposed ILP			JANE		
Dimension	Q_1 (s)	Median (s)	Q_3 (s)	Q_1 (s)	Median (s)	Q_3 (s)	Q_1 (s)	Median (s)	Q_3 (s)
4x4	5.95	8.76	16.24	0.04	0.05	0.05	0.16	0.16	0.17
5x5	224.60	281.69	620.72	0.06	0.07	0.08	0.18	0.20	0.21
6x6	36,630.70	85,495.87	108,078.08	0.11	0.12	0.14	0.22	0.23	0.24
7x7	–	–	–	0.19	0.20	0.21	0.27	0.28	0.29
8x8	–	–	–	0.28	0.30	0.30	0.37	0.39	0.76
9x9	–	–	–	0.46	0.47	0.58	0.44	0.71	0.83

TABLE 6.1. Median, upper and lower quartiles of running times for tanglegrams of a given dimension for all implementations on the small dataset.

The implementation of our proposed ILP ran on par with JANE in terms of running time – in fact, our ILP implementation outpaced JANE in almost all instances in this dataset (see full table of results, Table C.1). This was expected as the tanglegrams in this dataset start from toy example sizes to small real world sizes. Linear program solvers such as CPLEX have been well optimised to solve small ILPs.

JANE managed to find the optimal result in every instance in the dataset, as expected – the problem sizes are still small enough that with the default parameters, JANE should be able to search the entire solution space.

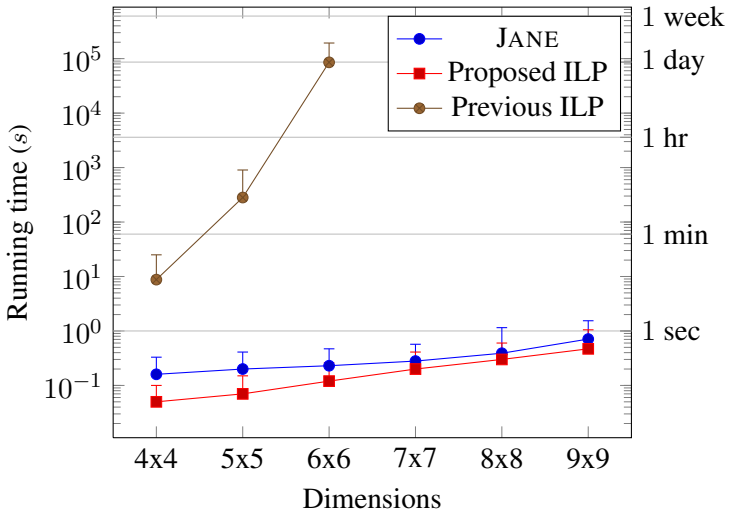


FIGURE 6.1. Plot of median (and upper quartile as errorbars) of running times for each tanglegram dimension in the small dataset.

The implementation of Libeskind-Hadas and Charleston [28] ILP could only run on instances of dimensions up to 6×6. Beyond this problem size, the ILP took too long to solve or required too much memory. Due to time constraints and memory constraints of the testbed computer, the Libeskind-Hadas and Charleston [28] ILP was not benchmarked on any other dataset.

6.3.1.2 Balanced and Unbalanced Dataset

The implementation of our proposed ILP and JANE were run over the large balanced and unbalanced dataset.

	Proposed ILP			JANE		
Dimension	Q_1 (s)	Median (s)	Q_3 (s)	Q_1 (s)	Median (s)	Q_3 (s)
5x5	0.06	0.08	0.08	0.18	0.20	0.21
10x10	0.71	0.81	0.83	0.82	1.13	1.39
15x15	4.39	4.97	7.25	1.91	2.26	2.35
20x20	45.94	96.13	192.64	2.55	2.59	2.75
25x25	665.45	7,176.22	36,255.30	3.61	3.65	3.73

TABLE 6.2. Median, upper and lower quartiles of running times for tanglegrams of a given dimension on the balanced dataset.

Whilst at these larger (and more realistic) problem sizes JANE performed much faster than the ILP (Fig. 6.2), our proposed solution still optimally solved all instances within a reasonable and practical

Dimension	Proposed ILP Q_1 (s)	Median (s)	Q_3 (s)	JANE Q_1 (s)	Median (s)	Q_3 (s)
15x20	8.94	15.61	44.85	2.27	2.42	2.45
15x25	46.79	85.74	115.38	2.34	2.65	2.88
15x30	46.91	148.76	259.31	2.56	2.73	2.87
20x25	43.01	159.52	354.34	3.00	3.18	3.26
20x30	1,546.90	4,326.93	7,374.95	3.28	3.55	3.57
20x35	1,179.54	4,161.58	10,851.58	3.55	3.81	3.86
20x40	3,730.19	19,797.15	45,008.41	3.78	4.08	4.12

TABLE 6.3. Median, upper and lower quartiles of running times for tanglegrams of a given dimension on the unbalanced dataset.

amount of time – the slowest result took 150812s (just under 42 hours). This was to be expected as finding optimal solutions to an ILP is an NP-complete task, as is finding the optimal solution to cophylogeny reconstruction. CPLEX is expected to take exponentially more time to solve the proposed ILP as the problem size grows. This was verified by the results over the balanced dataset.

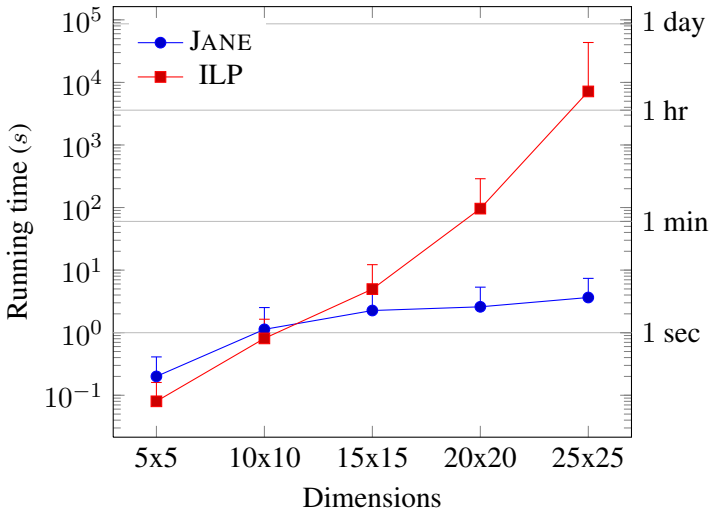


FIGURE 6.2. Plot of median (and upper quartile as errorbars) of running times for each tanglegram dimension in the large balanced dataset.

By comparing the results from the balanced and unbalanced dataset, we can see that raising the number of hosts h has a much greater impact than raising the number of parasite p (Fig. 6.3). This would also be expected – the most costly constraints to enforce are the temporal constraints which grow cubically with the number of host leaves. In contrast, parasite constraints only grow quadratically with the number of parasite leaves.

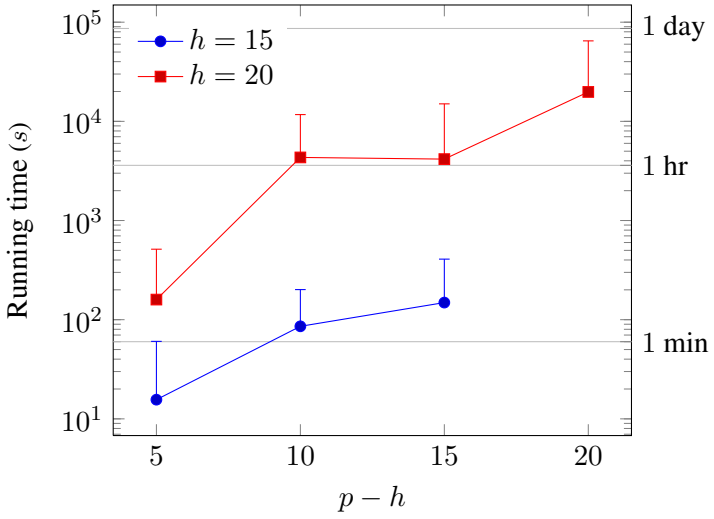


FIGURE 6.3. Plot of median (and upper quartile as errorbars) of running times for the ILP implementation for tanglegrams of fixed h and increasing p in the large unbalanced dataset.

The larger instances in the unbalanced dataset also revealed several cases where the solutions reported by JANE had higher objective costs than the exact optimal solutions found by the ILP implementation – 6 out of the 70 testcases in the unbalanced dataset (highlighted rows in Table C.3). This shows that JANE is not guaranteed to find the optimal solution and can fail on modest sized inputs (eg. 15×30).

JANE uses a probabilistic metaheuristic, so rerunning JANE multiple times or changing the algorithm parameters may allow JANE to find the optimal solution. However, this shows that the results of JANE can never be guaranteed to be optimal.

6.3.2 Real World Datasets

Only the implementation of our proposed ILP and JANE were run over the real world datasets. Both these datasets contained tanglegram sizes that exceed the practical limits of the Libeskind-Hadas and Charleston [28] ILP formulation.

6.3.2.1 Butterflies Dataset

The results of both methods on the butterflies dataset can be found in Table C.4.

The butterflies dataset contained relatively small tanglegrams, with both JANE and the proposed ILP method taking less than two seconds in all instances. None-the-less, this was still beyond the limits of the Libeskind-Hadas and Charleston [28] ILP implementation.

Across all tanglegrams in this dataset, the implementation of the proposed ILP actually performed consistently faster than JANE.

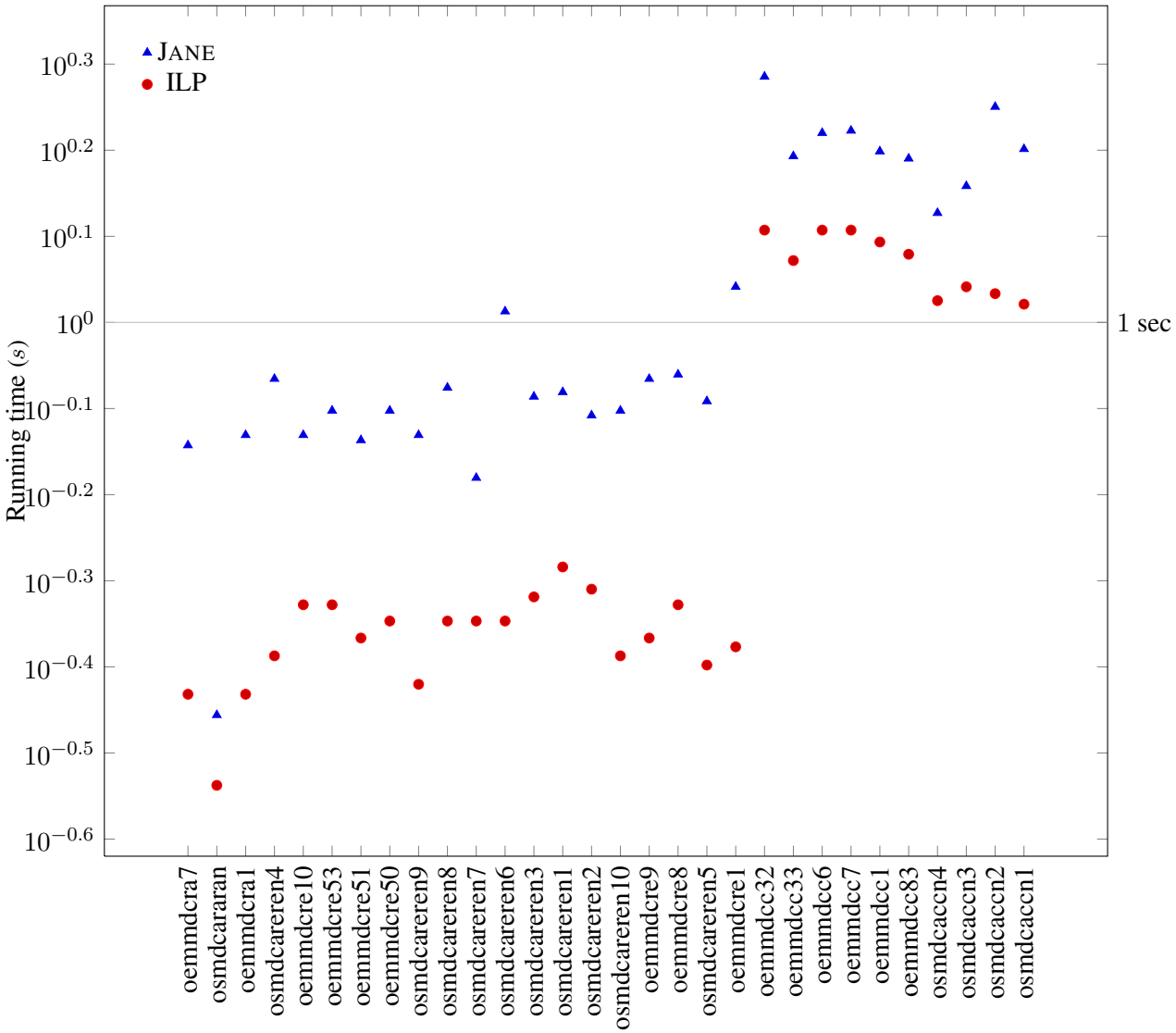


FIGURE 6.4. Plot of running times for each input instance in the butterflies dataset.

6.3.2.2 Assorted Dataset

The results of both methods on the assorted dataset can be found in Table C.5.

The ILP method found the optimal solution in all the tanglegrams of the dataset within the memory limits of the testbed computer. The slowest tanglegrams to process were the large Rodents-Hantavirus [44] and Finches-Vidua Brood Parasites [51] tanglegram, requiring 10 and 11 hours respectively.

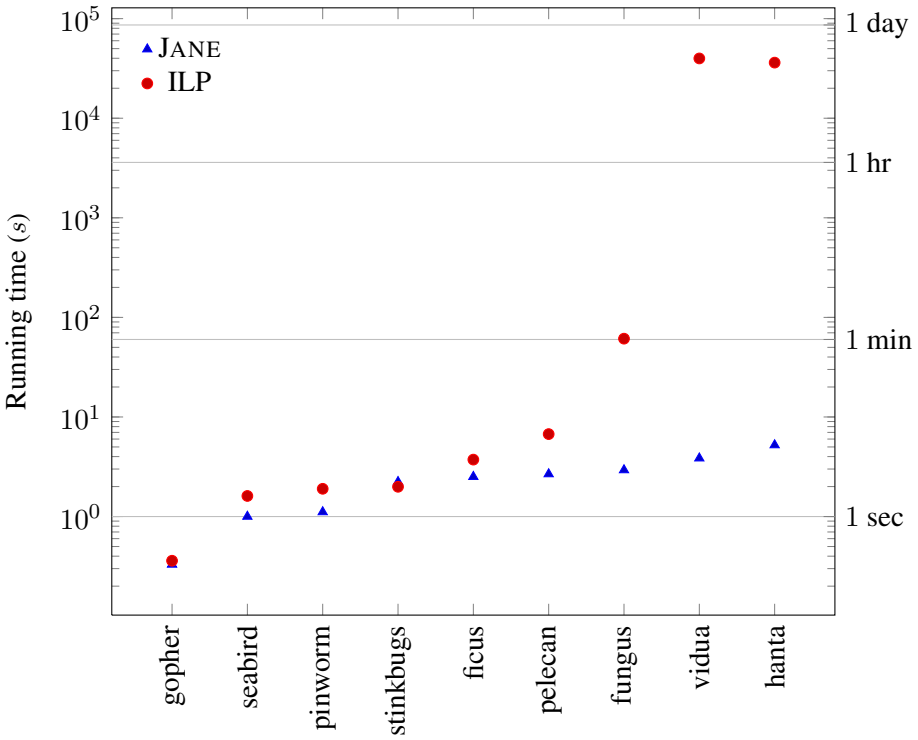


FIGURE 6.5. Plot of running times for each input instance in the assorted real world dataset.

This shows that, although exact and hence an exponential time approach to the reconstruction problem, the proposed ILP method is a viable alternative to heuristics methods currently employed. Whilst far slower, results are still obtainable within a reasonable amount of time and, unlike heuristic methods, are guaranteed to be optimal.

CHAPTER 7

Conclusion

7.1 Future Work

7.1.1 Pushing the bounds of intractability further

We have shown that our proposed ILP is a practical method for finding optimal cophylogeny reconstructions for current real world size datasets. However, due to the intractable nature of the reconstruction problem, the implementation quickly reaches a limit. Whilst practical by today's standards, our current implementation cannot feasibly solve large problem sizes which may arise in future research.

We have produced only a single implementation of the ILP using one of the many existing solvers. Our implementation also left the parameters of the solver unchanged. The next step would be to closely analyse the formulation to assist the solver with additional problem-specific knowledge – for instance, providing lazy cut constraints. The ILP should also be implemented across more solvers to take full advantage of the various development directions of different solvers.

Parallelisation is another approach to improving the practical limits of our ILP approach. Whilst CPLEX is multi-threaded, a large portion of the run time is sequential. Our ILP formulation is actually a zero-one integer linear program and can be rewritten as a pseudo-boolean optimisation problem. This will allow new implementations to use pseudo-boolean optimisers, a rapidly developing field with a current focus on parallelisation.

We have only used our ILP formulation to create an exact solution method. A further direction of research is to consider the integer relaxation of the ILP. This may lead to the development of bounded approximation algorithms – methods which can quickly find reconstructions with a guaranteed quality of solution. By nature, the ILP formulation is a purely mathematical model. This provides a wide scope for

the analytical demonstration of approximation bounds. Such methods would provide a middle-ground to the current choice between slow but exact solutions, and fast but non-guaranteed methods.

7.1.2 Expanding the model of reconstruction

Our ILP is based on a very restricted model of cophylogeny reconstruction. Several strict assumptions are made, and whilst these assumptions are common in current research, a future direction of research is in solving the cophylogeny reconstruction problem without such assumptions.

The most recent release of the JANE heuristic as of writing, JANE 3, has been adapted to accept a limited type of widespread parasites [2]. This may provide a foundation for remodelling our proposed ILP formulation to allow such variations.

A further assumption that might be weakened is the requirement that the input phylogenies are trees. Evolution processes can lead to more generalised directed acyclic graphs. Whilst Charleston [8] discussed a method to solve cophylogeny reconstruction on reticulate graphs, little progress has been made since. This is another model expansion that can be incorporated into the ILP.

7.2 Conclusions

The major contributions of the research presented in this thesis are the development of a mathematical model of the cophylogeny reconstruction problem, and from this model, a practical exact method of solving the problem. In the process, we have produced several useful analytical results on temporal compatibility – a crucial but often ignored aspect of cophylogeny reconstruction. From these results, we are able to produce exact mathematical definitions for temporal compatibility, leading to an ILP formulation of the reconstruction problem. Our concrete mathematical formulation of the problem creates a sound framework that can be used as a basis for further research on cophylogeny reconstruction. In particular, we expect that new optimisation techniques and approximation methods will be easier to develop as a result.

We have developed a reference implementation base on the proposed ILP formulation. Our results show that, unlike existing exact methods, our formulation remains practical for real world sized datasets. In addition, the results demonstrate that current heuristics cannot reliably converge on the optimal solution. In contrast, we have been able to prove analytically that our proposed ILP is guaranteed to find the optimal feasible reconstruction. Our implementation can be run on modest commodity hardware and still produce optimal results on real life datasets within a reasonable time frame. This is the first method to practically identify optimal solutions for the cophylogeny reconstruction problem and will facilitate the development of more accurate models of evolutionary systems in a wide range of disciplines, both within and beyond biology.

APPENDIX A

Corrections to Previous ILP Formulation

This section addresses errors identified in Libeskind-Hadas and Charleston [28].

A.1 Missing Constraints

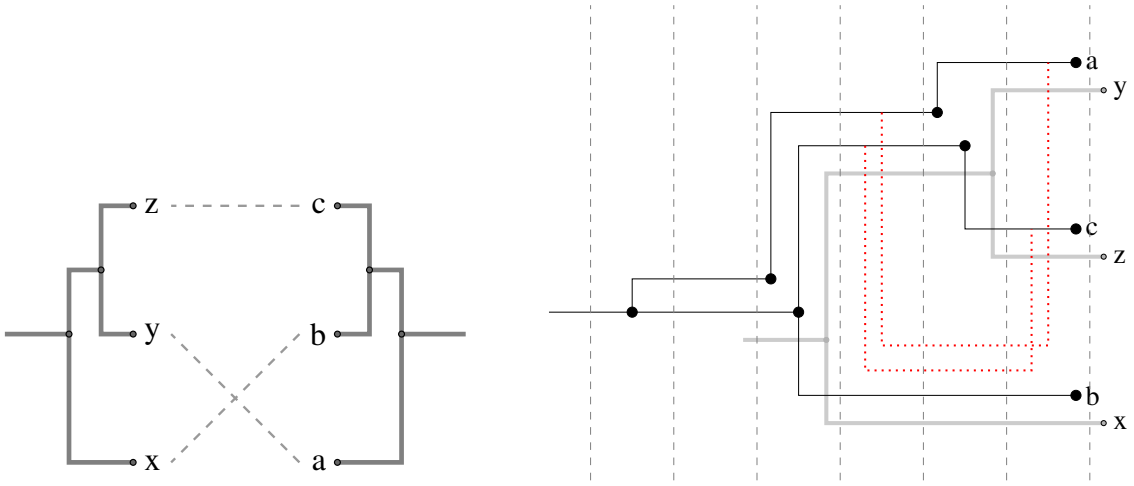
Constraints were described to ensure that host activity and parasite activity were contiguous (Constraint 3 and 6). However, the host-parasite associations were not required to be contiguous – that is, there was no analogue contiguous condition on the π variable. Without such a condition, full host switches would be considered possible, contradicting traceability (see Section 2.2.1).

This isn't just a theoretical concern. Consider the tanglegram in Fig. A.1a. An optimal solution for a cost scheme incurring a relatively high host switch cost is shown in Fig. A.1b. However, without a contiguity requirement on π, a lower cost alternate solution (normally considered infeasible) can be constructed.

We can add the following constraint to correct this oversight:

> **Constraint 15 (Mapping Contiguity):** If a parasite is active on a host at time t, then it must have been active on the host or the host's parent at time $t-1$ or it wasn't active at time $t-1$.

$$\pi_{i,j,t} \rightarrow \pi_{i,j,t-1} \vee \pi_{i,j',t-1} \vee \overline{p_{i,t-1}}$$



(A) Input tanglegram

(B) Optimal solution, and (red) lower cost infeasible solution allowed without mapping contiguity constraint.

FIGURE A.1. Example tanglegram with infeasible solution postulated by the Libeskind-Hadas and Charleston ILP formulation.

A.2 Incorrect Constraints

The constraints labelled *Cospeciation* (Constraint 11) and *Duplication* (Constraint 12) contain logical errors.

A.2.1 Constraint 11

Constraint 11 states [28]:

$$c_{i,j,t} \leftrightarrow \pi_{i,j,t} \wedge ps_{i,t} \wedge hs_{j,t} \wedge$$
$$[(\pi_{leftchild(i),leftchild(j),t+1} \wedge \pi_{rightchild(i),rightchild(j),t+1}) \vee$$
$$(\pi_{leftchild(i),rightchild(j),t+1} \wedge \pi_{rightchild(i),leftchild(j),t+1})]$$

This tries to capture two aspects of cospeciation in a single condition:

(1) Firstly, it tries to define a cospeciation: a cospeciation occurs if and only if a parasite is active on a host at some time, and both host and parasite speciate at this time.

(2) Secondly, it tries to enforce the requirement that after a cospeciation, the children of the parasite must be active on both children of the host.

However, the second of these aspects is not an *if and only if* condition. It is a condition that a cospeciation event imposes on the feasibility of the reconstruction, and not a condition the reconstruction imposes on the definition of a cospeciation.

As it stands, Constraint 11 allows for reconstructions that postulate cospeciations where both children of the parasite map to only a single child of the host. The condition imposed only means that such an event would not be counted as a cospeciation, but is still feasible.

We can fix this error by separating Constraint 11 into two constraints:

Constraint 11 (Cospeciation): Parasite i and host j cospeciate at time t if parasite i is active on host j at time t, and both parasite i and host j speciate at time t.

$$c_{i,j,t} \leftrightarrow \pi_{i,j,t} \wedge ps_{i,t} \wedge hs_{j,t}$$

Constraint 16 (Cospeciation Result): If parasite i cospeciates on host j at time t, then the children of i must be active on both children of j at time $t+1$.

$$c_{i,j,t} \leftrightarrow [\left(\pi_{leftchild(i),leftchild(j),t+1} \wedge \pi_{rightchild(i),rightchild(j),t+1}\right) \vee \left(\pi_{leftchild(i),rightchild(j),t+1} \wedge \pi_{rightchild(i),leftchild(j),t+1}\right)]$$

A.2.2 Constraint 12

Constraint 12 states [28]:

$$d_{i,t} \leftrightarrow ps_{i,t} \wedge \pi_{i,j,t} \wedge \overline{hs_j, t} \wedge \left(\pi_{leftchild(i),j,t+1} \vee \pi_{rightchild(i),j,t+1}\right)$$

Like Constraint 11, this tries to capture both the definition of a duplication event and also the traceability requirement on the children that result from a duplication event.

Again, this constraint does not prevent an infeasible duplication from occurring in a solution – it only prevents such an infeasible duplication from being recognised and counted.

We suggest splitting the constraint:

Constraint 12 (Duplication): Parasite i duplicates on host j at time t if parasite i is active on host j at time t, parasite i speciates at time t, and host j does not speciate at time t.

$$d_{i,t} \leftrightarrow ps_{i,t} \wedge \pi_{i,j,t} \wedge \overline{hs_j, t}$$

Constraint 17 (Duplication Result): If parasite i duplicates on host j at time t, then at least one of the children of i must be active j at time $t+1$.

$$d_{i,t} \rightarrow \left(\pi_{leftchild(i),j,t+1} \vee \pi_{rightchild(i),j,t+1}\right)$$

APPENDIX B

Full ILP Listing

For a description of variables, see Chapter 5.

minimise

$$Cost_{dup}\left(|P| - 1 - 2\sum \mathcal{C}_{p,h}\right) + Cost_{switch}\sum \mathcal{X}_{p,h_1,h_2} + Cost_{loss}\left(-|P| + \sum \Phi_{p,h}\right)$$

subject to

$$\begin{aligned}
\ll_{h_1,h_2} + \ll_{h_2,h_1} &= 1 && \forall h1 \neq h2 \in H \\
\ll_{h,h} &= 0 && \forall h \in H \\
-\ll_{h_1,h_2} - \ll_{h_2,h_3} + \ll_{h_1,h_3} &\geq -1 && \forall h_1 \neq h_2 \neq h_3 \in H \\
\ll_{h_1,h_2} &= 1 && \forall\,(h_1,h_2) \in \preceq_H \\
-\Phi_{p_1,h_1} - \Phi_{p_2,h_2} + \ll_{h_1\prime,h_2} &\geq -1 && \forall\,(p_1,p_2) \in \preceq_P, h_1, h_2 \in H \\
-\mathcal{X}_{p,h_1,h_2} + \ll_{h_2\prime,h_1} &\geq 0 && \forall p \in P, (h_1,h_2) \in \not\sim_H \\
-\Phi_{p_1,h_1} - \Phi_{p_2,h_2} + \ll_{h_1\prime,h_2} &\geq -1 && \forall p_1 \in P, h_1, h_2 \in H, \left({p_1}^+, p_2\right) \in \preceq_P \\
\Phi_{p,\varphi(p)} &= 1 && \forall p \in P_{\mathcal{L}} \\
-\Phi_{p,h_1} - \Phi_{p,h_2} &\geq -1 && \forall p \in P, (h_1,h_2) \in \not\sim_H \\
-\Phi_{p_0,h} + \Phi_{p_0,h'} - \Phi_{p_0,u} &\geq -1 && \forall\,(u,h) \in \preceq_H \\
-\mathcal{X}_{p,h_1,h_2} + \Phi_{p',h_1} &\geq 0 && \forall p \in P, (h_1,h_2) \in \not\sim_H \\
-\mathcal{X}_{p,h_1,h_2} + \Phi_{p,h_2} &\geq 0 && \forall p \in P, (h_1,h_2) \in \not\sim_H \\
-\mathcal{X}_{p,h_1,h_2} + \Phi_{p^+,h_1} &\geq 0 && \forall p \in P, (h_1,h_2) \in \not\sim_H \\
-\mathcal{C}_{p,h} + \Phi_{p,h} &\geq 0 \\
-\mathcal{C}_{p,h} + \Phi_{p_L,h_L} + \Phi_{p_R,h_L} &\geq 0 \\
-\mathcal{C}_{p,h} + \Phi_{p_L,h_R} + \Phi_{p_R,h_R} &\geq 0 && \forall p \in P - P_{\mathcal{L}}, h \in H - H_{\mathcal{L}}
\end{aligned}$$

$$-\Phi_{p',h} - \Phi_{p,h} + \Phi_{p^+,h} + \sum_{u:(h,u)\in\nsim_H} \mathcal{X}_{p^+,h,u} \geq -1 \qquad \forall p \in P, h \in H$$

$$-\Phi_{p,h} - \Phi_{p^+,h} + \Phi_{p,h'} + \Phi_{p',h} \geq -1 \qquad \forall p \in P, h \in H$$

$$\begin{aligned} -\Phi_{p,h} + \Phi_{p,h_L} + \Phi_{p,h_R} + \Phi_{p_L,h} + \Phi_{p_R,h} \\ +\Phi_{p_L,h_L} + \Phi_{p_L,h_R} + \Phi_{p_L,h_R} + \Phi_{p_L,h_L} \geq 0 \end{aligned} \qquad \forall p \in P, h \in H$$

$$-\Phi_{p,h} + \Phi_{p,h'} + \Phi_{p',h} + \Phi_{p',h'} + \sum_{u:(u,h)\in\nsim_H} \mathcal{X}_{p,u,h} \geq 0 \qquad \forall p \in P, h \in H$$

$$\ll_{h_1,h_2} \in \{0,1\} \qquad \forall h_1, h_2 \in H$$

$$\Phi_{p,h} \in \{0,1\} \qquad \forall p \in P, h \in H$$

$$\mathcal{X}_{p,h_1,h_2} \in \{0,1\} \qquad \forall p \in P, (h_1,h_2) \in \nsim_H$$

$$\mathcal{C}_{p,h} \in \{0,1\} \qquad \forall p \in P, h \in H$$

APPENDIX C

Full Result Tables

C.1 Column Descriptions

Dataset ID: In *real world* datasets, this is a simple descriptive identifier for the tanglegram. For *synthetic* datasets, this string contains the parameters that uniquely generate the tanglegram (including the dimensions of the tanglegram and a seed for the random number generator).

Dimension: The dimension of the tanglegram in the form $h \times p$ where h is the number of host leaves and p the number of parasite leaves.

ILP Obj.: The objective cost of the optimal reconstruction, as found by the ILP proposed in this thesis (see Sections 5.2 and 6.1.2).

ILP Time: The number of seconds (in terms of a wall clock) the reference implementation of the proposed ILP took to find the optimal solution.

Prev. ILP Obj.: The objective cost of the optimal reconstruction, as found by the Libeskind-Hadas and Charleston [28] ILP (see Section 6.1.3).

Prev. ILP Time: The number of seconds (in terms of a wall clock) the reference implementation of the Libeskind-Hadas and Charleston [28] ILP took to find the optimal solution. Where cells are empty, the implementation did not finish within two weeks, or exceeded the system memory of the testbed computer. Where the column is missing, the implementation did not succeed on any instances in the given dataset.

JANE Obj.: The objective cost of the best reconstruction found by the JANE 3 [2] implementation of the heuristic described in Conow et al. [15].

JANE Time: The number of seconds (in terms of a wall clock) the reference JANE 3 took to converge on a locally optimal solution.

A more detailed description of the benchmark process and the testbed computer is in Section 6.2.

C.2 Synthetic Datasets

C.2.1 Dataset – Small

Dataset ID	Dimension ($h \times p$)	Prev. ILP Obj.	Prev. ILP Time (s)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
4.4.Q_c5SXgbWsEU	4x4	5	6.69	5	0.05	5	0.16
4.4.PXZENCl06Szx	4x4	3	5.95	3	0.04	3	0.16
4.4.0E1C5gF8qvgP	4x4	5	11.94	5	0.04	5	0.16
4.4.0iFxsrUzlyvb	4x4	6	31.49	6	0.04	6	0.15
4.4.tHyYYRpRijH-	4x4	5	5.53	5	0.04	5	0.15
4.4.EE78GzmTW75w	4x4	3	7.28	3	0.05	3	0.17
4.4.1hypVzBQ3HPd	4x4	3	8.76	3	0.06	3	0.16
4.4.v_V4AwkI_lAs	4x4	3	4.51	3	0.06	3	0.17
4.4.NOU9w3bUhey8	4x4	6	26.71	6	0.04	6	0.18
4.4.QV1op-6TsloT	4x4	5	16.24	5	0.05	5	0.17

cont'd on next page

Dataset ID	Dimension ($h \times p$)	Prev. ILP Obj.	Prev. ILP Time (s)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
5.5.vf0FHaoNMA1i	5x5	9	620.72	9	0.08	9	0.20
5.5.BOeUl-ZWkUvG	5x5	6	227.38	6	0.06	6	0.16
5.5.we_kZurkTQHb	5x5	9	1,142.04	9	0.09	9	0.21
5.5.QHD2Vi3F90Kb	5x5	9	224.60	9	0.05	9	0.19
5.5.iaJwG3F9hdro	5x5	6	216.34	6	0.05	6	0.20
5.5.XjUobVWyJ09M	5x5	8	153.62	8	0.07	8	0.21
5.5.IA2JYfPzCNcg	5x5	9	281.69	9	0.06	9	0.21
5.5.GU5rgwasSryQ	5x5	9	260.11	9	0.06	9	0.20
5.5.rq83QsNlysFQ	5x5	8	668.64	8	0.07	8	0.18
5.5.UxjkZi8MReQo	5x5	5	298.66	5	0.08	5	0.18
6.6.4HwRBDUuyjFZ	6x6	6	47,460.67	6	0.12	6	0.22
6.6.sbXtMj9BWf3a	6x6	11	144,135.56	11	0.11	11	0.22
6.6.-HKiB1JsSg5B	6x6	12	15,866.61	12	0.17	12	0.24
6.6.pWh1Epco9W5J	6x6	10	108,078.08	10	0.11	10	0.24
6.6.tU3KoaFIKEnC	6x6	–	–	8	0.11	8	0.22
6.6.66UqWvAJz3GE	6x6	9	92,373.72	9	0.14	9	0.23
6.6.R0oXfVSEQn97	6x6	9	36,630.70	9	0.10	9	0.20
6.6.jBqJDQ648z0R	6x6	12	85,495.87	12	0.14	12	0.24
6.6.WFU7K7cfjKn_	6x6	–	–	9	0.14	9	0.25
6.6.wE7Z74D8zIfk	6x6	–	–	8	0.12	8	0.22

cont'd on next page

Dataset ID	Dimension ($h \times p$)	Prev. ILP Obj.	Prev. ILP Time (s)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
7.7.Y_Cxpk5-TvB5	7x7	–	–	9	0.14	9	0.27
7.7.FwuyxEa9TstB	7x7	–	–	12	0.19	12	0.27
7.7.bH1yFPnerIO0	7x7	–	–	12	0.24	12	0.29
7.7.QDUoQ2oH3Z5z	7x7	–	–	14	0.20	14	0.28
7.7.y-U8_Voy4LEL	7x7	–	–	12	0.20	12	0.29
7.7.sd37KBU4XOXW	7x7	–	–	12	0.19	12	0.29
7.7.qB_7GA3KkDrG	7x7	–	–	14	0.21	14	0.29
7.7.dKsZTijIEhTH	7x7	–	–	15	0.21	15	0.27
7.7.wWwlyzxsVeU5	7x7	–	–	14	0.19	14	0.27
7.7.FnWZySY7J0Q1	7x7	–	–	11	0.21	11	0.28
8.8.1k0dl38AradE	8x8	–	–	17	0.30	17	0.39
8.8.Jc42G4h9g-_g	8x8	–	–	14	0.27	14	0.38
8.8.4r6UNOXFB8-R	8x8	–	–	14	0.28	14	0.36
8.8.Qc6xSfgaT_YI	8x8	–	–	15	0.23	15	0.31
8.8.D_43STkJYiCh	8x8	–	–	12	0.30	12	0.81
8.8.oJzmZuYUndhw	8x8	–	–	12	0.28	12	0.40
8.8.faQ1XFwxJLBB	8x8	–	–	17	0.50	17	0.37
8.8.7lMHoLCxi8V9	8x8	–	–	14	0.28	14	0.91
8.8.OuJMrb7K2al2	8x8	–	–	17	0.36	17	0.37
8.8.OI0Of60D9SFZ	8x8	–	–	15	0.30	15	0.76

cont'd on next page

Dataset ID	Dimension ($h \times p$)	Prev. ILP Obj.	Prev. ILP Time (s)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
9.9.seV8FviChqx2	9x9	–	–	19	0.58	19	0.42
9.9.yQnHsVW58g1A	9x9	–	–	15	0.47	15	0.71
9.9.IPnfRTCEiBBi	9x9	–	–	17	0.45	17	0.71
9.9.WVM_atKtndrD	9x9	–	–	20	0.56	20	0.44
9.9.C4cAcYtuT0Wq	9x9	–	–	19	0.61	19	0.43
9.9.swLNjnK71hdD	9x9	–	–	15	0.47	15	1.11
9.9.C8OBVl3lO-kT	9x9	–	–	20	0.60	20	0.60
9.9.kKQ3n5ne9yRJ	9x9	–	–	18	0.47	18	0.82
9.9.uweNQu7TFGA8	9x9	–	–	17	0.46	17	0.83
9.9.l0MmxHMWkPyh	9x9	–	–	18	0.45	18	1.00

TABLE C.1. Full results for *small tanglegrams* dataset.

C.2.2 Dataset – Balanced

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
5.5.VYCEQExBSnPd	5x5	6	0.04	6	0.22
5.5.ur5J5eYrBU_S	5x5	5	0.09	5	0.21
5.5.sgqoguLyXeMt	5x5	8	0.07	8	0.19
5.5.JUXxJrK9QJCA	5x5	9	0.08	9	0.21
5.5.TOwvqULaOuYK	5x5	6	0.08	6	0.17
5.5.UOkNFwlvQICU	5x5	9	0.05	9	0.18
5.5.JcN-Axv2qmv4	5x5	6	0.07	6	0.18
5.5.gR9knCa9OEp8	5x5	8	0.09	8	0.20
5.5.3VBmEaZtSOPw	5x5	9	0.08	9	0.18
5.5.oxyPgeDa8a1f	5x5	9	0.06	9	0.21
10.10.ROcWSxP_VEft	10x10	20	0.83	20	0.58
10.10.O7McIW7gWLTt	10x10	20	0.71	20	1.13
10.10.aHNPxom6ob0y	10x10	19	0.81	19	0.84
10.10.bZ9bY5x41QEh	10x10	18	0.71	18	1.32
10.10.z_orjpuDSXAA	10x10	21	0.87	21	0.82
10.10.mK1Xp1I2TdyP	10x10	21	0.79	21	1.39
10.10.g4FsViSKOO1w	10x10	18	0.66	18	1.43
10.10.t1p-ld78mh_y	10x10	20	0.81	20	0.61
10.10.sRe27oG6coq5	10x10	20	0.86	20	0.82
10.10.TzUuddROlqqr	10x10	21	0.83	21	1.51
15.15.I285J5TgUhCl	15x15	33	7.25	33	2.50
15.15.jprDiEt6MYCI	15x15	29	4.57	29	1.76
15.15.lirYwb1Bx69C	15x15	32	4.39	32	2.08
15.15.XqhSSt1lMU6I	15x15	32	6.21	32	1.91
15.15.X42ArEnpwdkg	15x15	33	4.41	33	2.32
15.15.J4Hx0LKcVTDE	15x15	26	2.64	26	2.07
15.15.pV20YH7H0zhB	15x15	30	3.34	30	1.90
15.15.BQnGkxS-kD6u	15x15	37	9.25	37	2.39

cont'd on next page

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
15.15.FrEwQuHUAUXd	15x15	36	7.97	36	2.35
15.15.IuX7B68a0rK_	15x15	31	4.97	31	2.26
20.20.dyT7G1TFUopb	20x20	48	192.64	48	2.57
20.20.HvBJkCRbwjh7	20x20	45	92.65	45	2.59
20.20.YTV8XM–uteb	20x20	48	346.77	48	2.75
20.20.UHb_lENCW3h8	20x20	51	264.17	51	2.70
20.20._ztj4hXEr2bW	20x20	47	74.77	47	2.55
20.20.FmMweLM9ZOi7	20x20	47	142.01	47	2.15
20.20.7GEwhjw8fD9g	20x20	49	45.94	49	3.03
20.20._ndjIo8hccQS	20x20	47	96.13	47	2.50
20.20.vyuC0ooIgHwV	20x20	46	22.84	46	3.40
20.20.tEbbzRB_MHRf	20x20	47	19.01	47	2.57
25.25.Z_OP9NziASht	25x25	64	1,998.12	64	3.69
25.25.nidCLV7q4Sr4	25x25	60	7,176.22	60	3.65
25.25.8a4D2AkVI8yl	25x25	64	34,345.07	64	3.65
25.25.ATmlq9ZiWbVU	25x25	64	36,255.30	64	3.77
25.25.nUZMYgHTEtEx	25x25	66	150,812.13	66	3.63
25.25.N4s_8hAAErTK	25x25	63	665.45	63	3.61
25.25.2fbwJQEWhYpI	25x25	56	166.98	56	4.08
25.25.LmU2CP-6wcad	25x25	59	403.68	59	3.73
25.25.5qnDG-icjw1L	25x25	64	5,590.40	64	3.51
25.25.uBbisz2oe4bS	25x25	61	68,227.21	61	3.48

TABLE C.2. Full results for *balanced tanglegrams* dataset.

C.2.3 Dataset – Unbalanced

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
15.20.dIHLO2qJZmiJ	15x20	38	4.55	38	2.45
15.20.zHF3DcOliH7E	15x20	45	7.75	45	2.17
15.20.xLkVxBAXdX0c	15x20	45	11.76	45	2.77
15.20.iD8Fv2_X0ev8	15x20	47	44.85	47	2.23
15.20.s-G3XaWNrMqD	15x20	45	8.94	45	2.38
15.20.A_RhErTkN7fV	15x20	52	303.04	52	2.45
15.20.uF4Pdxeg-76q	15x20	50	108.25	50	2.42
15.20.vrYN12Ost7cs	15x20	41	15.61	41	2.30
15.20.-byJD4GdszNT	15x20	48	16.30	48	2.27
15.20.-2F0XXZ9lL4q	15x20	47	12.53	47	2.89
15.25.Chaq5YIRFw-Z	15x25	59	115.38	59	2.88
15.25.Kvl9dQqbuGsP	15x25	61	88.42	61	2.98
15.25.J73tomIKoTS7	15x25	61	133.64	61	2.34
15.25.9ciKIEMhmdg-	15x25	59	18.47	59	2.14
15.25.laRcsapwf7MH	15x25	61	54.10	61	2.77
15.25.dNpT78UBze1P	15x25	52	10.21	52	2.88
15.25.5tZXrb9zHOxR	15x25	62	46.79	62	2.34
15.25.O8bjBvyM5J-k	15x25	53	48.41	53	2.65
15.25.2TkzrLIazc58	15x25	58	85.74	58	2.37
15.25.myVV0j-v_AI3	15x25	61	155.93	61	2.59
15.30.-o8HhB4jZ2vj	15x30	71	515.44	71	2.56
15.30.fL4m4ghoBE3m	15x30	69	24.79	69	2.85
15.30.eKyAovGLw2Mf	15x30	69	57.07	69	2.87
15.30.YrxyjFZwa7Sz	15x30	69	46.91	69	2.37
15.30.Mvk_aVGRCpLp	15x30	68	190.75	69	2.42
15.30.M0obRvRwJixH	15x30	67	311.96	67	2.62
15.30.yM3AX_hBr-mr	15x30	64	141.81	64	2.67
15.30.VnoMgaSsWXIX	15x30	71	259.31	71	2.88

cont'd on next page

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
15.30.6-JooQ1gs9Pe	15x30	67	13.96	67	3.05
15.30.wg9cZ91_It4p	15x30	72	148.76	74	2.73
20.25.1-EikzHDLk7d	20x25	59	114.33	59	3.00
20.25.xKA1QBHmt1vX	20x25	62	645.63	63	3.17
20.25.zZWqfDj-ToGy	20x25	58	23.28	58	3.18
20.25.gsBGybkTSkWd	20x25	56	43.01	56	3.03
20.25.vxOcuGhFJM8W	20x25	55	12.12	55	2.80
20.25.ji25qa-OmLbO	20x25	63	1,202.01	63	3.26
20.25.NmsD9zCs5q5z	20x25	61	354.34	61	3.30
20.25.ZHimFkjd3tfd	20x25	60	159.52	60	2.96
20.25.I2J_qNxYBzWo	20x25	60	148.25	60	3.34
20.25.06ebukD-ujce	20x25	59	207.09	60	3.22
20.30.hfVpUXdHYeCw	20x30	72	1,546.90	72	3.12
20.30.4EUNnPKI5dD4	20x30	73	6,073.64	73	3.62
20.30.Bby-H_HbA7mY	20x30	76	15,078.78	76	3.34
20.30.YVh9VSAqRLph	20x30	70	349.33	70	2.84
20.30.MUDvBew9rxUz	20x30	73	4,326.93	73	3.58
20.30.FjWIfDF2R2Sx	20x30	73	910.83	73	3.28
20.30.kOR3qm8KIuTd	20x30	74	1,718.75	74	3.55
20.30.9wL5j3xADJo5	20x30	76	7,374.95	76	3.57
20.30.AsCeVvzpsx-W	20x30	73	22,229.99	73	3.50
20.30.F511HQof9KHW	20x30	73	3,984.77	73	3.56
20.35.kE_nKwyqxsVn	20x35	83	1,179.54	83	3.81
20.35.cHpBY-7WGGT3	20x35	86	19,082.84	86	3.76
20.35.-0SDKvZxIvgu	20x35	83	3,652.56	83	3.91
20.35.UWfiG2LIuJ3K	20x35	83	4,161.58	83	3.65
20.35.dP2M3Tj-Pmiy	20x35	80	2,783.10	80	3.44
20.35.ucb4TAjGa95-	20x35	88	9,702.95	88	3.86
20.35.doYWGNmQm4mK	20x35	89	15,107.91	91	3.81

cont'd on next page

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
20.35.WBTdYpEXAbwa	20x35	84	610.98	84	3.55
20.35.nQPY3BiT89Df	20x35	76	42.55	76	3.55
20.35.1jSkLnDoHX1q	20x35	88	10,851.58	89	4.29
20.40.4Z4hncX5A6wV	20x40	98	21,422.59	98	3.78
20.40.cgGhiVQu3Ga_	20x40	100	45,008.41	100	4.38
20.40.-pXrUKuyKHnI	20x40	95	19,797.15	95	4.10
20.40.J1FldJd2Ktxb	20x40	94	16,715.82	94	3.64
20.40.HYf4Xf0PqPs8	20x40	92	3,730.19	92	4.08
20.40.Pz3K-1LzfAbu	20x40	87	8,332.68	87	3.94
20.40.9N2IMplupoXj	20x40	101	56,606.13	101	4.12
20.40.BiZNaJN2kxKk	20x40	90	799.29	90	4.79
20.40.cW27IL_LSfA4	20x40	98	66,215.58	98	3.57
20.40._3SHz31k8bIn	20x40	99	2,134.76	99	3.88

TABLE C.3. Full results for *unbalanced tanglegrams* dataset. Highlighted: instances with non-optimal JANE results.

C.3 Real World Datasets

C.3.1 Dataset – Butterflies

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
oemmdcra7	8x8	15	0.37	15	0.72
osmdcararan	8x8	10	0.29	10	0.35
oemmdcra1	8x8	15	0.37	15	0.74
osmdcareren4	9x9	11	0.41	11	0.86
oemmdcre10	9x9	14	0.47	14	0.74
oemmdcre53	9x9	14	0.47	14	0.79
oemmdcre51	9x9	14	0.43	14	0.73
oemmdcre50	9x9	11	0.45	11	0.79
osmdcareren9	9x9	11	0.38	11	0.74
osmdcareren8	9x9	14	0.45	14	0.84
osmdcareren7	9x9	14	0.45	14	0.66
osmdcareren6	9x9	14	0.45	14	1.03
osmdcareren3	9x9	14	0.48	14	0.82
osmdcareren1	9x9	14	0.52	14	0.83
osmdcareren2	9x9	14	0.49	14	0.78
osmdcareren10	9x9	10	0.41	10	0.79
oemmdcre9	9x9	11	0.43	11	0.86
oemmdcre8	9x9	11	0.47	11	0.87
osmdcareren5	9x9	10	0.40	10	0.81
oemmdcre1	9x9	11	0.42	11	1.10
oemmdcc32	12x12	17	1.28	17	1.93
oemmdcc33	12x12	15	1.18	15	1.56
oemmdcc6	12x12	17	1.28	17	1.66
oemmdcc7	12x12	17	1.28	17	1.67
oemmdcc1	12x12	18	1.24	18	1.58
oemmdcc83	12x12	15	1.20	15	1.55
osmdcaccn4	12x12	10	1.06	10	1.34

cont'd on next page

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
osmdcaccn3	12x12	13	1.10	13	1.44
osmdcaccn2	12x12	10	1.08	10	1.78
osmdcaccn1	12x12	13	1.05	13	1.59

TABLE C.4. Full results for *butterflies* dataset.

C.3.2 Dataset – Assorted

Dataset ID	Dimension ($h \times p$)	ILP Obj.	ILP Time (s)	JANE Obj.	JANE Time (s)
gopher	8x10	11	0.36	11	0.33
seabird	11x14	20	1.61	20	1.00
pinworm	13x13	20	1.90	20	1.11
stinkbugs	14x14	8	1.99	8	2.23
ficus	16x16	20	3.73	20	2.51
pelecan	18x18	30	6.73	30	2.67
fungus	20x24	51	61.13	51	2.93
vidua	33x21	44	39,848.36	44	3.85
hanta	34x42	70	36,173.02	70	5.22

TABLE C.5. Full results for *assorted* dataset.

Bibliography

[1] Atkinson, Q. D. (2011). Phonemic Diversity Supports a Serial Founder Effect Model of Language Expansion from Africa. *Science*, 332(6027):346–349.

[2] Black, K. (2011). JANE 3. http://www.cs.hmc.edu/~hadas/jane/index.html.

[3] Brooks, D. R. (1981). Hennig's Parasitological Method: A Proposed Solution. *Systematic Zoology*, 30(3):pp. 229–249.

[4] Brooks, D. R. and McLennan, D. A. (1991). *Phylogeny, Ecology, and Behavior: A Research Program in Comparative Biology*. University Of Chicago Press, 1 edition.

[5] Brooks, D. R., Van Veller, M. G. P., and McLennan, D. A. (2002). How to do BPA, really. *Journal of Biogeography*, 28(3):345–358.

[6] Charleston, M. A. (1998). Jungles: a new solution to the host/parasite phylogeny reconciliation problem. *Mathematical Biosciences*, 149(2):191–223.

[7] Charleston, M. A. (2002). Principles of cophylogenetic maps. *Biological Evolution and Statistical Physics*, pages 122–147.

[8] Charleston, M. A. (2003). Recent results in cophylogeny mapping. *Advances in parasitology*, 54:303–30.

[9] Charleston, M. A. (2009). A New Likelihood Method for Cophylogenetic Analysis. Technical Report 636, The University of Sydney.

[10] Charleston, M. A. (2011a). personal communication.

[11] Charleston, M. A. (2011b). TREEMAP 3. http://sydney.edu.au/engineering/it/~mcharles/software/treemap/treemap3.html.

[12] Charleston, M. A. and Galvani, A. (2006). A cophylogenetic perspective of host-pathogen evolution. In *Disease evolution: Models, concepts and data analyses*, pages 146–160. American Mathematical Society.

[13] Charleston, M. A. and Page, R. D. M. (2002). TREEMAP 2. http://sydney.edu.au/engineering/it/~mcharles/software/treemap/treemap.html.

[14] Conow, C., Fielder, D., and Ovadia, Y. (2010a). JANE. http://www.cs.hmc.edu/~hadas/jane/Jane1/index.html.

[15] Conow, C., Fielder, D., Ovadia, Y., and Libeskind-Hadas, R. (2010b). Jane: a new tool for the cophylogeny reconstruction problem. *Algorithms for molecular biology : AMB*, 5(1):16.

[16] Cousins, B., Peebles, J., Schramm, T., and Yodpinyanee, A. (2010). JANE 2. http://www.cs.hmc.edu/~hadas/jane/index.html.

[17] Dunn, M., Greenhill, S. J., Levinson, S. C., and Gray, R. D. (2011). Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, to appear.

[18] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

[19] Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E., and Matsuda, G. (1979). Fitting the Gene Lineage into its Species Lineage, a Parsimony Strategy Illustrated by Cladograms Constructed from Globin Sequences. *Systematic Zoology*, 28(2):pp. 132–163.

[20] Gray, R. D. and Jordan, F. M. (2000). Language trees support the express-train sequence of Austronesian expansion. *Nature*, 405(6790):1052–5.

[21] Hafner, M. S. and Nadler, S. A. (1988). Phylogenetic trees support the coevolution of parasites and their hosts. *Nature*, 332(6161):258–9.

[22] Hougen-Eitzman, D. and Rausher, M. D. (1994). Interactions between Herbivorous Insects and Plant-Insect Coevolution. *The American Naturalist*, 143(4):pp. 677–697.

[23] Hoyal Cuthill, J. F. and Charleston, M. A. (2011). in preparation.

[24] Hughes, J., Kennedy, M., Johnson, K. P., Palma, R. L., and Page, R. D. M. (2007). Multiple cophylogenetic analyses reveal frequent cospeciation between pelecaniform birds and Pectinopygus lice. *Systematic biology*, 56(2):232–51.

[25] Hugot, J.-P. (2003). New Evidence for Hystricognath Rodent Monophyly from the Phylogeny of their Pinworms. In *Tangled Trees: Phylogeny, Cospeciation and Coevolution*, pages 144—-173. University of Chicago Press.

[26] Junick, S., Merkle, D., Middendorf, M., and Legat, R. (2005). TARZAN. http://pacosy.informatik.uni-leipzig.de/51-0-Tarzan.html.

[27] Kikuchi, Y., Hosokawa, T., Nikoh, N., Meng, X.-Y., Kamagata, Y., and Fukatsu, T. (2009). Host-symbiont co-speciation and reductive genome evolution in gut symbiotic bacteria of acanthosomatid stinkbugs. *BMC biology*, 7(1):2.

[28] Libeskind-Hadas, R. and Charleston, M. A. (2008). An Integer Linear Programming Formulation of the Cophylogeny Reconstruction Problem. Technical Report 629, The University of Sydney.

[29] Libeskind-Hadas, R. and Charleston, M. A. (2009). On the computational complexity of the reticulate cophylogeny reconstruction problem. *Journal of Computational Biology : a Journal of Computational Molecular Cell Biology*, 16(1):105–17.

[30] Merkle, D. and Middendorf, M. (2005). Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory in Biosciences*, 123(4):277–99.

[31] Merkle, D., Middendorf, M., and Wieseke, N. (2010a). A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC bioinformatics*, 11 Suppl 1:S60.

[32] Merkle, D., Middendorf, M., and Wieseke, N. (2010b). CORE-PA. http://pacosy.informatik.uni-leipzig.de/49-1-CoRe-PA.html.

[33] Nelson, G. J. and Platnick, N. I. (1981). *Systematics and Biogeography: Cladistics and Vicariance*. Columbia University Press.

[34] Nöllenburg, M., Holten, D., Völker, M., and Wolff, A. (2008). Drawing Binary Tanglegrams: An Experimental Evaluation. *Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments ALENEX*, pages 106–119.

[35] Ovadia, Y., Fielder, D., Conow, C., and Libeskind-Hadas, R. (2011). The co phylogeny reconstruction problem is NP-complete. *Journal of Computational Biology : a Journal of Computational Molecular Cell Biology*, 18(1):59–65.

[36] Page, R. D. M. (1990). Component Analysis: A Valiant Failure? *Cladistics*, 6(2):119–136.

[37] Page, R. D. M. (1993). Parasites, phylogeny and cospeciation. *International Journal for Parasitology*, 23(4):499–506.

[38] Page, R. D. M. (1994a). Maps Between Trees and Cladistic Analysis of Historical Associations among Genes,Organisms, and Areas. *Systematic Biology*, 43(1):58–77.

[39] Page, R. D. M. (1994b). Parallel Phylogenies: Reconstructing the History of Host-Parasite Assemblages. *Cladistics*, 10(2):155–173.

[40] Page, R. D. M. (1995). TREEMAP. http://taxonomy.zoology.gla.ac.uk/rod/treemap.html.

[41] Page, R. D. M. and Charleston, M. A. (2002). TreeMap versus BPA (again): a response to Dowling. *Taxonomy*, 02(02):1–26.

[42] Paterson, A. M., Palma, R., and Gray, R. D. (2003). Drowning on arrival, missing the boat, and x-events: How likely are sorting events? In *Tangled Trees: Phylogeny, Cospeciation and Coevolution*, pages 287–309. University Of Chicago Press, Chicago.

[43] Paterson, A. M., Wallis, G. P., Wallis, L. J., and Gray, R. D. (2000). Seabird and Louse Coevolution: Complex Histories Revealed by 12S rRNA Sequences and Reconciliation Analyses. *Systematic Biology*, 49(3):383–399.

[44] Ramsden, C., Holmes, E. C., and Charleston, M. A. (2009). Hantavirus evolution in relation to its rodent and insectivore hosts: no evidence for codivergence. *Molecular biology and evolution*, 26(1):143–53.

[45] Refrégier, G., Le Gac, M., Jabbour, F., Widmer, A., Shykoff, J. A., Yockteng, R., Hood, M. E., and Giraud, T. (2008). Cophylogeny of the anther smut fungi and their caryophyllaceous hosts: prevalence of host shifts and importance of delimiting parasite species for inferring cospeciation. *BMC evolutionary biology*, 8(1):100.

[46] Ronquist, F. (1995). Reconstructing the history of host-parasite associations using generalised parsimony. *Cladistics*, 11(1):73–89.

[47] Ronquist, F. (1998). Three-Dimensional Cost-Matrix Optimization and Maximum Cospeciation. *Cladistics*, 14(2):167–172.

[48] Ronquist, F. (2003). Parsimony analysis of coevolving species associations. In *Tangled Trees: Phylogeny, Cospeciation and Coevolution*, pages 22–64. University of Chicago Press.

[49] Ronquist, F. and Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–4.

[50] Ronquist, F. and Nylin, S. (1990). Process and Pattern in the Evolution of Species Associations. *Systematic Zoology*, 39(4):323.

[51] Sorenson, M. D., Balakrishnan, C. N., and Payne, R. B. (2004). Clade-Limited Colonization in Brood Parasitic Finches (Vidua spp.). *Systematic Biology*, 53(1):140–153.

[52] Tehrani, J. J., Collard, M., and Shennan, S. J. (2010). The cophylogeny of populations and cultures: reconstructing the evolution of Iranian tribal craft traditions using trees and jungles. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 365(1559):3865–74.

[53] Weiblen, G. D. and Bush, G. L. (2002). Speciation in fig pollinators and parasites. *Molecular ecology*, 11(8):1573–8.

[54] Wiley, E. O. (1988). Parsimony Analysis and Vicariance Biogeography. *Systematic Zoology*, 37(3):pp. 271–290.

[55] Woolhouse, M. E. J., Webster, J. P., Domingo, E., Charlesworth, B., and Levin, B. R. (2002). Biological and biomedical implications of the co-evolution of pathogens and their hosts. *Nature genetics*, 32(4):569–77.