COMPUTING THE MINIMUM NUMBER OF HYBRIDIZATION EVENTS FOR A CONSISTENT EVOLUTIONARY HISTORY

MAGNUS BORDEWICH AND CHARLES SEMPLE

Abstract. It is now well-documented that the structure of evolutionary relationships between a set of present-day species is not necessarily tree-like. The reason for this is that reticulation events such as hybridizations mean that species are a mixture of genes from different ancestors. Since such events are relatively rare, a fundamental problem for biologists is to determine the smallest number of hybridization events required to explain a given (input) set of data in a single (hybrid) phylogeny. The main results of this paper show that computing this smallest number is APX-hard, and thus NP-hard, in the case the input is a collection of phylogenetic trees on sets of present-day species. This answers a problem which was raised at a recent conference (Phylogenetic Combinatorics and Applications, Uppsala University, 2004). As a consequence of these results, we also correct a previously published NP-hardness proof in the case the input is a collection of binary sequences, where each sequence represents the attributes of a particular present-day species. The APX-hardness of these problems mean that it is unlikely that there is an efficient algorithm for either computing the result exactly, or approximating it to any arbitrary degree of accuracy.

1. Introduction

Evolutionary trees, also called (rooted) phylogenetic trees, are used in evolutionary biology to represent the ancestral history of a collection of present-day species. However, evolution is not always tree-like because of reticulation events such as hybridizations and lateral gene transfers. Consequently, rooted acyclic digraphs, in which there is exactly one vertex that has in-degree zero and where the vertices of out-degree zero represent the present-day species, are being used to model reticulate evolution (see, for example, [2, 8, 14, 18]). In such digraphs, vertices with in-degree at least two represent reticulation events. In this paper, we generically call these vertices 'hybridization vertices' and these digraphs 'hybrid phylogenies'.

Hybridization events are relatively rare and so a fundamental problem for biologists studying the evolution of species whose past has included hybridization is

Date: 12 July 2006.

1991 *Mathematics Subject Classification.* 05C05; 92D15.

Key words and phrases. Rooted phylogenetic tree, reticulate evolution, hybrid phylogeny, phylogenetic network, agreement forest, rooted subtree prune and regraft.

The first author was supported by the New Zealand Institute of Mathematics and its Applications funded programme *Phylogenetic Genomics* and the second author was supported by the New Zealand Marsden Fund (UOC310). This work was done while the first author was a Postdoctoral Fellow at the University of Canterbury.

the following: given a collection of phylogenetic trees on sets of species that correctly represent the tree-like evolution of different parts of various species genomes, what is the smallest number of hybridization events required so that the all of the trees in this collection are simultaneously 'displayed' by a single hybrid phylogeny. This smallest number sets a lower bound on the degree of hybridization that has occurred in the evolution of the species under consideration. Posed in this way in [2], and in [14] but with an additional time constraint, this and similar problems have attracted recent interest (see, for example, [7, 8, 20]). The main results of this paper show that computing this smallest number is also APX-hard and thus, consequently, NP-hard. The latter means that, unless P=NP, there is some fixed positive constant c strictly bigger than 1 for which there is no polynomial-time algorithm such that, for all instances, the ratio between the size of the feasible solution outputted by the algorithm and the size of the optimal solution is always smaller than c. In fact, we show that the APX-hardness of computing this smallest number holds even for the simplest case in which the input collection consists of just two phylogenetic trees on the same set of species.

The paper is organized as follows. The next section contains some necessary preliminaries and a mathematical formalization of the above optimization problem for the simplest case (which we call Minimum Hybridization). Formal statements of the main results of this paper, as well as a short summary of the complexity classes and concepts used in these results are also included in this section. The proofs of the main results are given in Section 3. Section 4, contains some consequences of the work in Section 3 for the computational complexity of computing the so-called *rooted subtree prune and regraft distance* between a pair of phylogenetic trees. This measure of distance is closely associated with modelling reticulate evolution. Lastly, Section 5 contains a discussion of the problem *perfect phylogeny with recombination*, previously examined in [8] and [20]. We point out an error in the proof given in [20] that this problem is NP- and APX-hard, and use our earlier results to provide a correct proof. In general, the notation and terminology throughout this paper follows [17].

2. Preliminaries and Main Results

For a digraph D and a vertex v of D, we denote the in-degree and out-degree of v by $d^-(v)$ and $d^+(v)$, respectively. A *hybrid phylogeny* or *hybrid* (on X) is an ordered pair $\mathcal{H} = (D; \phi)$ consisting of

(i) a rooted acyclic digraph D in which the root has out-degree at least two and, for all vertices v with $d^+(v) = 1$, we have $d^-(v) \geq 2$, and
(ii) a bijective map ϕ from X into the set of vertices of D with out-degree zero.

For completeness, if $|X| = 1$, then the digraph consisting of an isolated vertex v and a map from X into $\{v\}$ is also defined to be a hybrid on X. The set X corresponds to the set of present-day species and is called the *label set* of $\mathcal{H}$ which is denoted by $\mathcal{L}(\mathcal{H})$. Vertices of in-degree at least two (called *hybridization vertices*) represent hybridization events and correspond to an exchange of genetic information between

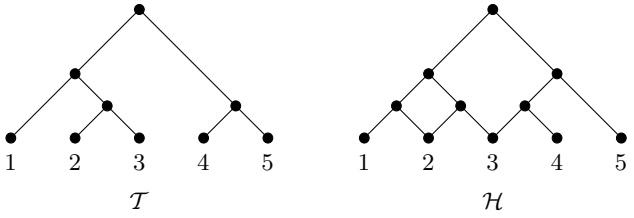

FIGURE 1. A rooted binary phylogenetic tree $\mathcal{T}$ and a hybrid $\mathcal{H}$ displaying $\mathcal{T}$.

hypothetical ancestors. The *hybridization number* of $\mathcal{H}$, denoted $h(\mathcal{H})$, is

$$h(\mathcal{H}) = \sum_{v \neq \rho} (d^-(v) - 1),$$

where ρ denotes the root of $\mathcal{H}$. Observe that $h(\mathcal{H}) \geq 0$, and $h(\mathcal{H}) = 0$ precisely if D is a rooted tree. Throughout this paper, we adopt the convention that hybrid phylogenies are always drawn with their arcs directed downwards and so omit the arrowheads. A hybrid phylogeny $\mathcal{H}$ with $h(\mathcal{H}) = 2$ is shown in Fig. 1.

A *rooted binary phylogenetic tree* is a special type of hybrid phylogeny in which the root has degree two and all other interior vertices have degree three, and (apart from the root) all vertices have in-degree one.

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree and let $\mathcal{H}$ be a hybrid phylogeny on X. We say that $\mathcal{H}$ *displays* $\mathcal{T}$ if $\mathcal{T}$ can be obtained from a rooted *subtree* of $\mathcal{H}$ by contracting degree-two vertices. In other words, $\mathcal{T}$ can be obtained from $\mathcal{H}$ by deleting first a subset of the edges of $\mathcal{H}$, and then deleting the isolated vertices, and contracting non-root degree-two vertices. For example, in Fig. 1, the hybrid $\mathcal{H}$ displays the rooted binary phylogenetic tree $\mathcal{T}$. For two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, we set

$$h(\mathcal{T}, \mathcal{T}') = \min\{h(\mathcal{H}) : \mathcal{H} \text{ is a hybrid on } X \text{ that displays } \mathcal{T} \text{ and } \mathcal{T}'\}.$$

The optimization problem MINIMUM HYBRIDIZATION is formally stated as follows.

MINIMUM HYBRIDIZATION
Instance: A finite set X, and two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Goal: Find a hybrid phylogeny $\mathcal{H}$ that displays $\mathcal{T}$ and $\mathcal{T}'$ with minimum hybridization number.
Measure: The value of $h(\mathcal{H})$.

The main results of this paper are Theorem 2.1 and Corollary 2.2.

Theorem 2.1. *The optimization problem* MINIMUM HYBRIDIZATION *is* APX*-hard. In particular, there is no polynomial-time approximation scheme for* MINIMUM HYBRIDIZATION *unless* P=NP.

It immediately follows from Theorem 2.1 that the analogous formalization of the (general) fundamental problem described in the introduction, where we are given an arbitrary size collection of rooted phylogenetic trees is APX-hard.

Corollary 2.2. *Unless* P=NP*, there is no polynomial-time approximation algorithm for* Minimum Hybridization *with an approximation ratio better than* $\frac{2113}{2112}$.

We end this section with a short summary of the complexity classes and concepts described in Theorem 2.1 and Corollary 2.2. For further details, we refer the reader to [15] and [1].

For optimization problems that are NP-hard, an important consideration is the possibility of polynomial-time approximation algorithms. In such an algorithm, one would like to guarantee for all instances that the ratio between the size of the feasible solution outputted by the algorithm and the size of an optimal solution is always smaller than some fixed constant. To treat minimization and maximization problems in the same way, we will assume that this ratio is always at least 1. The existence of polynomial-time approximation algorithms varies greatly amongst NP-hard problems. Indeed, there are some NP-hard problems π for which regardless of the size of this fixed constant, there is always such an algorithm. In this case, π is said to exhibit a *polynomial-time approximation scheme* (PTAS). Such problems include the problem of finding a maximum independent set in a planar graph. But then there are other NP-hard problems, such as the (general) travelling salesman problem, for which there exists no polynomial-time approximation algorithm (no matter how big the fixed constant is) unless P=NP.

The class APX (also known as MAX SNP) is the class of optimization problems for which there exists a polynomial-time approximation algorithm for some constant approximation ratio. Within this class, is the class of APX-complete problems. If an optimization problem is APX-complete, then it has no polynomial-time approximation scheme unless P=NP. Assuming that P $\neq$ NP, this implies that there is some fixed constant r strictly bigger than 1 for which there is no polynomial-time approximation algorithm with ratio r. To show that an optimization problem π_2 is APX-hard, it suffices to find an APX-complete problem π_1 and show that there is an 'L-reduction' from π_1 to π_2. Introduced by Papadimitriou and Yannakakis [15], the reason that this suffices is that L-reductions preserve approximability.

Let π_1 and π_2 be two optimization problems. An *L-reduction* from π_1 to π_2 is a pair of polynomial-time computable functions f and g, and a pair of positive constants α and β that satisfy the following properties:

(i) If I is an instance of π_1, then $f(I)$ is an instance of π_2 with

$$\mathrm{opt}(f(I)) \leq \alpha\, \mathrm{opt}(I),$$

where $\mathrm{opt}(I)$ and $\mathrm{opt}(f(I))$ denotes the size of an optimal solution to I and $f(I)$, respectively.

(ii) If S is a feasible solution of $f(I)$, then $g(S)$ is a feasible solution of I with

$$|\mathrm{opt}(I) - c(g(S))| \leq \beta\, |\mathrm{opt}(f(I)) - c(S)|,$$

where $c(g(S))$ and $c(S)$ is the size of $g(S)$ and S, respectively.

3. Proofs of Theorem 2.1 and Corollary 2.2

We prove Theorem 2.1 (and Corollary 2.2) in two steps. The first step is by establishing an L-reduction from Maximum 4-Dimensional Matching to a problem we call Maximum-Acyclic-Agreement Forest, while the second step is showing that there is an L-reduction from this latter problem to Minimum Hybridization.

Agreement Forests. Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree and let X' be a subset of X. The minimal rooted subtree of $\mathcal{T}$ that connects the vertices of $\mathcal{T}$ labelled by the elements of X' is denoted by $\mathcal{T}(X')$. Furthermore, the *restriction* of $\mathcal{T}$ to X', denoted by $\mathcal{T}|X'$, is the rooted binary phylogenetic tree that is obtained from $\mathcal{T}(X')$ by contracting any non-root vertices of degree two.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. For the purposes of the definition of an agreement forest, we regard the root of both $\mathcal{T}$ and $\mathcal{T}'$ as a vertex ρ at the end of a pendant edge adjoined to the original root. Furthermore, we also regard ρ as part of the label sets of $\mathcal{T}$ and $\mathcal{T}'$, thus we view both label sets as $X \cup \{\rho\}$. An *agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ is a collection $\{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$, where $\mathcal{T}_\rho$ is a rooted tree whose label set $\mathcal{L}_\rho$ includes ρ and $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ are rooted binary phylogenetic trees with label sets $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$, respectively, such that the following properties are satisfied:

(i) The label sets $\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ partition $X \cup \{\rho\}$.
(ii) For all $i \in \{\rho, 1, 2, \ldots, k\}$, $\mathcal{T}_i \cong \mathcal{T}|\mathcal{L}_i \cong \mathcal{T}'|\mathcal{L}_i$.
(iii) The trees in $\{\mathcal{T}(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ and $\{\mathcal{T}'(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex disjoint rooted subtrees of $\mathcal{T}$ and $\mathcal{T}'$, respectively.

It is easily seen that if $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, then, up to contracting non-root vertices of degree two, $\mathcal{F}$ can be obtained from each of $\mathcal{T}$ and $\mathcal{T}'$ by deleting $|\mathcal{F}| - 1$ edges. An agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ is a *maximum-agreement forest* if, amongst all agreement forests for $\mathcal{T}$ and $\mathcal{T}'$, it has the smallest number of components, in which case we denote the value of k by $m(\mathcal{T}, \mathcal{T}')$.

Intuitively, the deleted edges are those which disagree in $\mathcal{T}$ and $\mathcal{T}'$, and hence correspond to different paths of genetic inheritance, *i.e.* hybridization events. So the fewer edges deleted, the smaller the number of hybridization events. However, one additional condition is required to link agreement forests and the hybridization number formally. This condition excludes agreement forests in which any vertex in the associated hybrid phylogeny inherits genetic information from its own descendants.

Let $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ be an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Let $G_\mathcal{F}$ be the directed graph whose vertex set is $\mathcal{F}$ and for which $(\mathcal{T}_i, \mathcal{T}_j)$ is an arc precisely if $i \neq j$ and either

(I) the root of $\mathcal{T}(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}(\mathcal{L}_j)$, or
(II) the root of $\mathcal{T}'(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}'(\mathcal{L}_j)$.

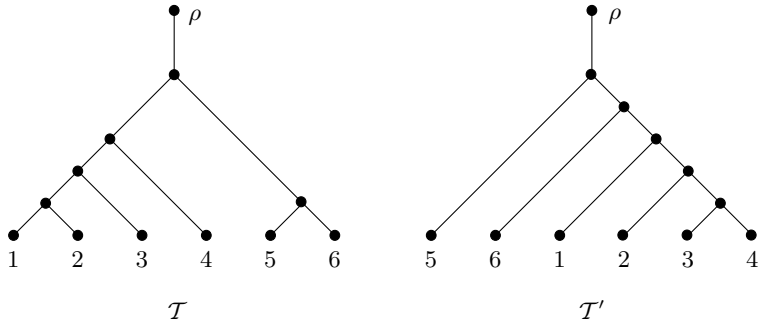

FIGURE 2. Two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$.

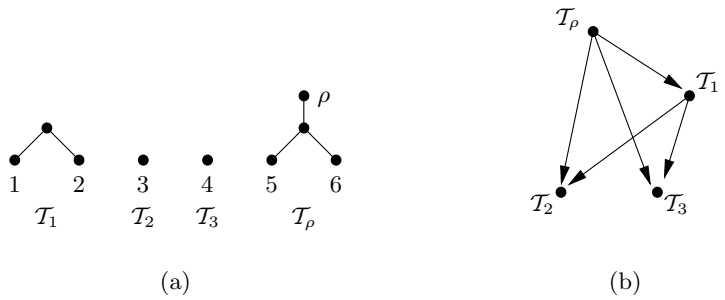

FIGURE 3. (a) A maximum-acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$. (b) The graph $G_{\mathcal{F}}$.

Since $\mathcal{F}$ is an agreement forest, the roots of $\mathcal{T}(\mathcal{L}_i)$ and $\mathcal{T}(\mathcal{L}_j)$, and the roots of $\mathcal{T}'(\mathcal{L}_i)$ and $\mathcal{T}'(\mathcal{L}_j)$ are not the same. We say that $\mathcal{F}$ is an *acyclic-agreement forest* if $G_{\mathcal{F}}$ is acyclic. (Note that, in [3], the adjective "good" is used instead of "acyclic" in the definition of an acyclic-agreement forest.) Furthermore, if $\mathcal{F}$ contains the smallest number of components over all acyclic-agreement forests for $\mathcal{T}$ and $\mathcal{T}'$, we say that $\mathcal{F}$ is a *maximum-acyclic-agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$, in which case we denote this value of k by $m_a(\mathcal{T}, \mathcal{T}')$. Observe that $m_a(\mathcal{T}, \mathcal{T}') = 0$ if and only if, up to isomorphism, $\mathcal{T}$ and $\mathcal{T}'$ are identical. To illustrate these definitions, Fig. 3(a) shows a maximum-acyclic-agreement forest $\mathcal{F}$ for the two rooted binary phylogenetic trees shown in Fig. 2, where we have adjoined to the root of each of $\mathcal{T}$ and $\mathcal{T}'$ a pendant edge as described above. The graph $G_{\mathcal{F}}$ is shown in Fig. 3(b).

The problem MAXIMUM-ACYCLIC-AGREEMENT FOREST is formally stated as follows.

MAXIMUM-ACYCLIC-AGREEMENT FOREST
Instance: A finite set X, and two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Goal: Find a maximum-acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$.
Measure: The number of components in $\mathcal{F}$ minus one.

For us, all of the work in proving Theorem 2.1 goes into establishing the L-reduction from Maximum 4-Dimensional Matching to Maximum-Acyclic-Agreement Forest because of the following theorem in [3, Theorem 2].

Theorem 3.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then*

(i) $h(\mathcal{T}, \mathcal{T}') = m_a(\mathcal{T}, \mathcal{T}')$.
(ii) *If $\mathcal{H}$ is a hybrid phylogeny that displays $\mathcal{T}$ and $\mathcal{T}'$, then there is a polynomial-time algorithm for converting $\mathcal{H}$ into an acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$. Furthermore,*

$$(|\mathcal{F}| - 1) - m_a(\mathcal{T}, \mathcal{T}') \leq h(\mathcal{H}) - h(\mathcal{T}, \mathcal{T}').$$

Remarks. Part (ii) in Theorem 3.1 is not explicitly stated in [3]. However, it is a consequence of the proof of [3, Theorem 2]. Intuitively, one takes $\mathcal{H}$ and systematically cuts off rooted subtrees whose root has in-degree at least two. By viewing the root of $\mathcal{H}$ as a vertex at the end of a pendant edge adjoined to the original root, we obtain an acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$, and so $|\mathcal{F}| - 1 \leq h(\mathcal{H})$. This construction also provides one direction of (i). For the other direction of (i), if $\mathcal{F}$ is an acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, then, taking an acyclic ordering of $G_{\mathcal{F}}$, one can construct a hybrid phylogeny beginning with the component of $\mathcal{F}$ containing the label ρ and systematically adjoining the rest of the components (respecting the ordering) with at most two new edges to the current hybrid phylogeny so that the resulting hybrid phylogeny displays the appropriate restrictions of $\mathcal{T}$ and $\mathcal{T}'$. The value of the hybridization number of the final hybrid phylogeny in this construction is at most $|\mathcal{F}| - 1$.

The next corollary is an immediate consequence of Theorem 3.1.

Corollary 3.2. *There is an L-reduction from* Maximum-Acyclic-Agreement Forest *to* Minimum Hybridization *with $\alpha = 1$ and $\beta = 1$.*

It follows from Corollary 3.2 that Minimum Hybridization is APX-hard if Maximum-Acyclic-Agreement Forest is APX-hard. With this in mind, we next show that there is an L-reduction from the following problem to Maximum-Acyclic-Agreement Forest.

Maximum B-Dimensional Matching (Max-BDM)
Instance: B disjoint sets $X_1, X_2, \ldots, X_B$. A subset Q of $X_1 \times X_2 \times \cdots \times X_B$.
Goal: Find a maximum-sized subset M of Q with the property that no two members of M agree in any coordinate.
Measure: The cardinality of M.

Kann [12] showed that Max-3DM is APX-complete, even when each element of $\cup_{i=1}^{B} X_i$ appears in at most 3 members of Q. Hazan *et al.* [9] proved explicit inapproximability ratios for Max-BDM, for $B \geq 4$. Chlebík and Chlebíková [6] gave tighter inapproximability ratios for Max-3DM and Max-4DM, and importantly their results hold even in the restricted case that each element of $\cup_{i=1}^{B} X_i$ appears in exactly 2 members of Q. We denote this restricted case by Max-BDM-2. We

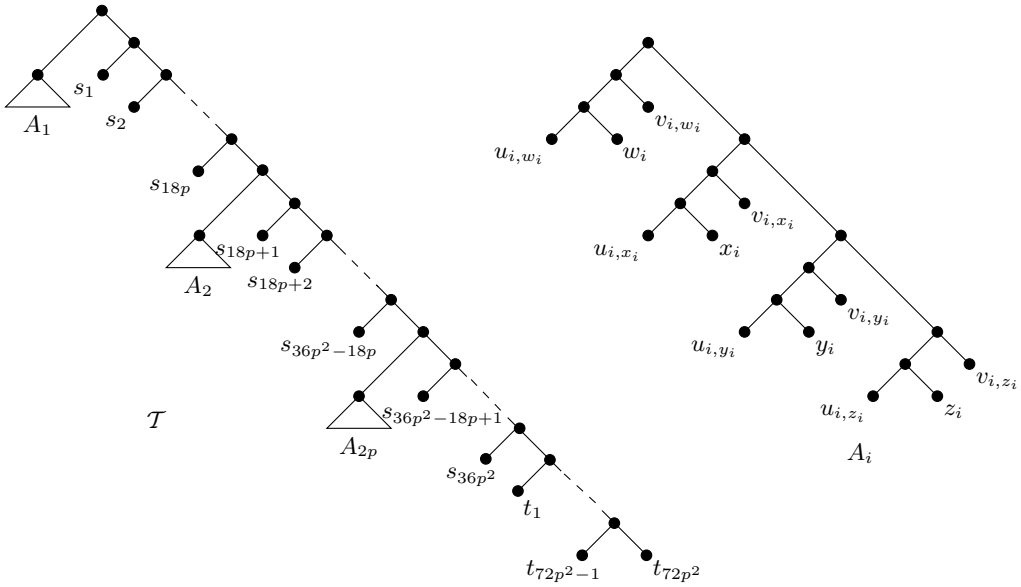

FIGURE 4. The tree $\mathcal{T}$ and its subtrees A_i.

will show that there is an L-reduction from MAX-4DM-2 to MAXIMUM-ACYCLIC-AGREEMENT FOREST.

Let W, X, Y, Z and $Q \subseteq W \times X \times Y \times Z$ be an instance I of MAX-4DM-2. Let $|W| = p$. Since each element of $W \cup X \cup Y \cup Z$ appears in exactly 2 members of Q, we have

$$p = |W| = |X| = |Y| = |Z| = |Q|/2.$$

Using the above instance of MAX-4DM-2, we now construct two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ with the same label sets. With some modifications, this construction follows the same construction as that used in [11] and [5] to show that a certain related problem is NP-hard but with MAX-4DM-2 replacing EXACT COVER BY 3-SETS (see Section 4 for further details).

Let $Q = \{(w_1, x_1, y_1, z_1), (w_2, x_2, y_2, z_2), \ldots, (w_{2p}, x_{2p}, y_{2p}, z_{2p})\}$. The tree $\mathcal{T}$ is shown in Fig. 4. Each subtree A_i, with $i = 1 \ldots 2p$, corresponds to exactly one tuple in Q. The tree $\mathcal{T}'$ is shown in Fig. 5. Each subtree B_r corresponds to an element r of $W \cup X \cup Y \cup Z$, where i and j identify the two members of Q in which r appears. The order of attaching the subtrees B_r for $r \in W \cup X \cup Y \cup Z$ to the spine of $\mathcal{T}'$ is not important. Each subtree C_i, with $i = 1 \ldots 2p$, corresponds to a tuple in Q.

The following lemma is central to the proof that MAXIMUM-ACYCLIC AGREEMENT FOREST is APX-hard. Although not used in this section, the second part of the lemma will be used in Section 4.

Lemma 3.3.

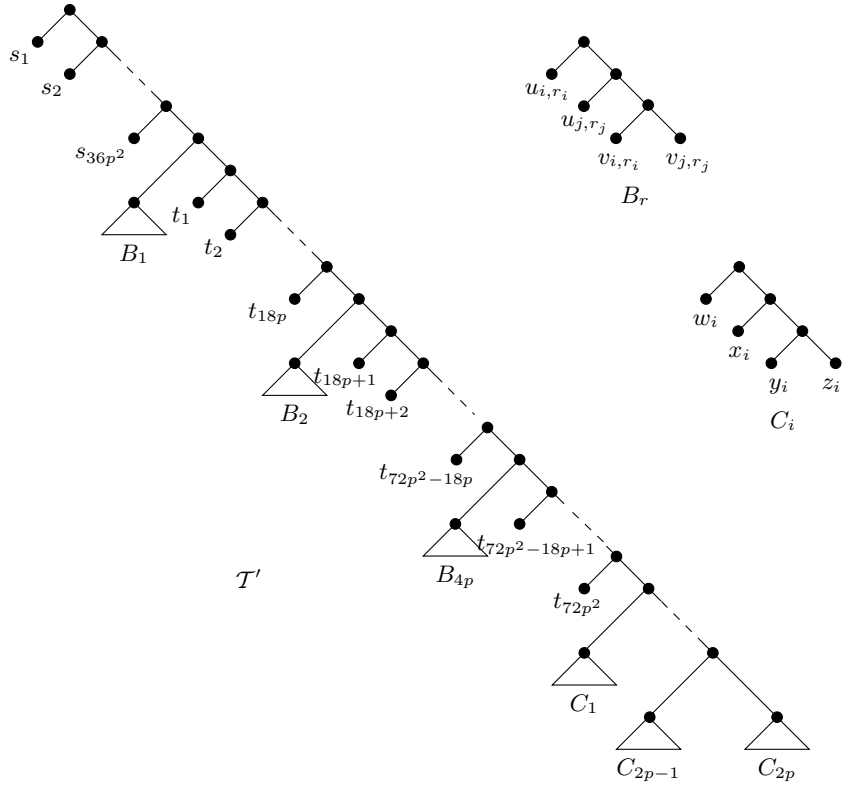

FIGURE 5. The tree $\mathcal{T}'$, and its subtrees B_r and C_i.

(i) *Q contains a 4-dimensional matching of size k if and only if there is an acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of size*

$$1 + 8k + 9(2p - k) = 18p - k + 1.$$

In particular, $m_a(\mathcal{T}, \mathcal{T}') = 18p - \mathrm{opt}(Q)$.

(ii) *If there is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of size*

$$1 + 8k + 9(2p - k) = 18p - k + 1,$$

then Q contains a 4-dimensional matching of size k. In particular, in combination with the necessary direction of (i), $m(\mathcal{T}, \mathcal{T}') = 18p - \mathrm{opt}(Q)$.

Proof. We first prove the necessary direction of (i). Suppose Q contains a 4-dimensional matching M of size k. We can obtain an acyclic-agreement forest $\mathcal{F}_M$ of size $18p - k + 1$ for $\mathcal{T}$ and $\mathcal{T}'$ by making the following edge deletions to $\mathcal{T}$ and then contracting any resulting non-root degree-two vertices:

(i) For each i, delete the edge attaching A_i to the rest of $\mathcal{T}$.

(ii) For each i, if A_i corresponds to a tuple in M, delete each of the pendant edges attaching w_i, x_i, y_i, and z_i, and then delete each of the edges attaching the subtrees containing u_{i,w_i} and v_{i,w_i}, u_{i,x_i} and v_{i,x_i}, and u_{i,y_i} and v_{i,y_i}. Thus, in this case, each A_i is broken into 8 components.

(iii) For each i, if A_i does not correspond to a tuple in M, then delete each of the pendant edges attaching the leaves u_{i,w_i}, v_{i,w_i}, u_{i,x_i}, v_{i,x_i}, u_{i,y_i}, v_{i,y_i}, u_{i,z_i}, and v_{i,z_i}. In this case, each A_i is broken into 9 components.

Clearly, $|\mathcal{F}_M| = 1+8k+9(2p-k) = 18p-k+1$. Furthermore, noting that each B_r corresponds to a particular element of $W \cup X \cup Y \cup Z$, we also have that $\mathcal{F}_M$ can be obtained from $\mathcal{T}'$ by making the following edge deletions and then contracting any resulting non-root degree-two vertices:

(i)$'$ For each i and r, delete the edge attaching B_r and C_i to the rest of $\mathcal{T}'$.

(ii)$'$ For each i, if C_i corresponds to a tuple in M, delete each of the pendant edges attaching w_i, x_i and y_i, so C_i is broken into 4 components. If C_i does not correspond to a tuple in M, it remains 1 component. Thus the cuttings in (ii)$'$ together with the cutting of each C_i in (i)$'$ contribute $4k + (2p-k)$ components.

(iii)$'$ For each $r \in W \cup X \cup Y \cup Z$, if r appears in a tuple in M, then it appears at most once, in which case without loss of generality we may assume r_i appears in some tuple in M, but r_j does not. Then in B_r delete each of the edges attaching u_{j,r_j} and v_{j,r_j}, so that B_r is broken into 3 components. If neither r_i nor r_j appears in a tuple in M, then in B_r delete each of the pendant edges attaching u_{i,r_i}, v_{i,r_i} and u_{j,r_j}, so that B_r is broken into 4 components. Hence the cuttings in (iii)$'$ together with the cutting of each B_r in (i)$'$ contribute $4k \cdot 3 + (4p-4k)4$ components.

In this case also, as expected, the total number of components is

$$1 + 4k + (2p-k) + 4k \cdot 3 + (4p-4k)4 = 1 + 8k + 9(2p-k).$$

Hence $\mathcal{F}_M$ is indeed an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. A routine check now shows that $\mathcal{F}_M$ is also an acyclic-agreement forest.

We next simultaneously prove (ii) and the sufficient direction of (i). Let $S = \{s_1, s_2, \ldots, s_{36p^2}, t_1, t_2, \ldots, t_{72p^2}\}$. Let $\mathcal{F}$ be an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of size at most $18p+1$. Note that $\mathcal{F}$ may or may not be acyclic. We first show that if $\mathcal{T}_j$ is a tree in $\mathcal{F}$ with label set $\mathcal{L}(\mathcal{T}_j)$, then if $\mathcal{L}(\mathcal{T}_j) \cap \mathcal{L}(A_i) \neq \emptyset$ it follows that $\mathcal{L}(\mathcal{T}_j) \subseteq \mathcal{L}(A_i)$, and if $\mathcal{L}(\mathcal{T}_j) \cap \mathcal{L}(B_r) \neq \emptyset$ it follows that $\mathcal{L}(\mathcal{T}_j) \subseteq \mathcal{L}(B_r)$.

Let $\mathcal{T}_j$ be a tree in $\mathcal{F}$, and first assume that for some i the set $\mathcal{L}(\mathcal{T}_j) \cap \mathcal{L}(A_i)$ is non-empty and contains at least one element x of $\mathcal{L}(\mathcal{T})$ not in $\mathcal{L}(A_i)$. Suppose that $x \in \mathcal{L}(A_{i'})$ for some $i' \neq i$. Then, since $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, there are at least $18p$ members of S that appear as singletons in $\mathcal{F}$ (those in the chain between A_i and $A_{i'}$). By comparing $\mathcal{T}$ and $\mathcal{T}'$, the label set of no component in $\mathcal{F}$ contains the entire label set of A_i (for any i), and so $\mathcal{F}$ contains at least $18p+2$ components; a contradiction. Now suppose that $x \in S$. If $x \in \{t_1, t_2, \ldots, t_{72p^2}\}$, then, as $\mathcal{F}$ is an agreement forest, each of the $18p$ elements in $\{s_{36p^2-18p+1}, \ldots, s_{36p^2}\}$ appear as singletons in $\mathcal{F}$. As the label set of no component

in $\mathcal{F}$ contains the entire label set of A_i, this implies that $\mathcal{F}$ contains at least $18p+2$ components; a contradiction. Therefore we may assume that $x \in \{s_1, s_2, \ldots, s_{36p^2}\}$. Using an argument similar to that just used, it is straightforward to deduce that $x \in \{s_{36p^2-18p+1}, \ldots, s_{36p^2}\}$. But then, by comparing $\mathcal{T}$ and $\mathcal{T}'$, the $18p-1$ other elements in $\{s_{36p^2-18p+1}, \ldots, s_{36p^2}\}$ appear as singletons in $\mathcal{F}$. Since the label set of no component of $\mathcal{F}$ contains a label in A_i and a label in $\{t_1, t_2, \ldots, t_{21pq}\}$ and since the label set of A_i is not a subset of the label set of a single component of $\mathcal{F}$, we again deduce that $\mathcal{F}$ contains at least $18p+2$ components; a contradiction. Effectively, this means that to obtain $\mathcal{F}$ from $\mathcal{T}$ each edge joining an A_i to the rest of $\mathcal{T}$ is deleted. Using this last fact, the result for $\mathcal{L}(\mathcal{T}_j) \cap \mathcal{L}(B_r) \neq \emptyset$ follows easily by similar reasoning.

Now suppose that $\mathcal{F}$ is an agreement forest of size $1+8k+9(2p-k) = 18p-k+1$. Fixing i, consider A_i. By the argument above, there is a subset of the components of $\mathcal{F}$ in which the union of the label sets is the label set of A_i. Since no component can contain labels from more than one B_r, a routine check shows that this subset must have at least 8 elements and, moreover, this subset has exactly 8 elements only if the partition of $\mathcal{L}(A_i)$ induced by the label sets is

$$\big\{\{w_i\}, \{x_i\}, \{y_i\}, \{z_i\}, \{u_{i,w_i}, v_{i,w_i}\}, \{u_{i,x_i}, v_{i,x_i}\}, \{u_{i,y_i}, v_{i,y_i}\}, \{u_{i,z_i}, v_{i,z_i}\}\big\}.$$

It now follows that each A_i contributes at least 8 components to $\mathcal{F}$. An important observation at this point is that regardless of the composition of $\mathcal{F}$, it is always an acyclic-agreement forest.

Since $\mathcal{F}$ has $1 + 8k + 9(2p - k)$ components, it follows from the last paragraph that at least k of the A_i's are 'partitioned' into 8 parts as described above. Let A_i and A_j be two such subtrees, and consider the associated tuples (w_i, x_i, y_i, z_i) and (w_j, x_j, y_j, z_j). Suppose that one of the components agree. Without loss of generality, we may assume that $x_i = x_j$. Since A_i and A_j are both partitioned into 8 parts, $\{u_{i,x_i}, v_{i,x_i}\}$ is the label set of one component of $\mathcal{F}$ and $\{u_{j,x_j}, v_{j,x_j}\}$ is the label set of another component of $\mathcal{F}$. But then, in $\mathcal{T}'$, the minimal subtree connecting u_{i,x_i} and v_{i,x_i} and the minimal subtree connecting u_{j,x_j} and v_{j,x_j} are not disjoint; a contradiction. Thus (w_i, x_i, y_i, z_i) and (w_j, x_j, y_j, z_j) have no coordinates in common. We conclude that Q contains a 4-dimensional matching of size k. This establishes both (ii) and the sufficient direction of (i). □

Theorem 3.4. *The optimization problem* Maximum-Acyclic-Agreement Forest *is* APX*-hard. In particular, unless* P=NP*, there is no polynomial-time approximation scheme for* Maximum-Acyclic-Agreement Forest.

Proof. To establish the result, we show that there is an L-reduction from Max-4DM-2 to Maximum-Acyclic-Agreement Forest. First note that by picking any m in Q and removing all other tuples which agree with m in at least one coordinate (thus removing at most 5 members including the one originally picked), and then picking another member from the resulting set and continuing this process, we observe that $\text{opt}(Q) \geq \frac{2p}{5}$; that is

$$2p \leq 5\text{opt}(Q). \tag{1}$$

Let I be an instance of Max-4DM-2, and let $f(I)$ be the function that maps I to $\mathcal{T}$ and $\mathcal{T}'$, an instance of Maximum-Acyclic-Agreement Forest as described prior to Lemma 3.3. Clearly, this mapping is computable in polynomial time in the size of I. Furthermore, by Lemma 3.3 and (1),

$$\begin{aligned} m_a(\mathcal{T},\mathcal{T}') &= 18p - \operatorname{opt}(Q) \\ &\leq 9(5\operatorname{opt}(Q)) - \operatorname{opt}(Q) \\ &= 44\operatorname{opt}(Q). \end{aligned}$$

It now follows that (i) in the definition of an L-reduction holds with $\alpha = 44$.

To see that (ii) holds, let $\mathcal{F}$ be an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of size $S_2 + 1 = 18p - k + 1$. Let g be the function that maps $\mathcal{F}$ to the feasible solution of I of size $S_1 = k$ as described at the end of the proof of Lemma 3.3. Again, g can be computed in polynomial time. Then $S_2 = 18p - S_1$, and so

$$\begin{aligned} & 18p - \operatorname{opt}(Q) = m_a(\mathcal{T},\mathcal{T}') \\ \Leftrightarrow \quad & 18p - S_1 - (18p - \operatorname{opt}(Q)) = S_2 - m_a(\mathcal{T},\mathcal{T}') \\ \Leftrightarrow \quad & \operatorname{opt}(Q) - S_1 = S_2 - m_a(\mathcal{T},\mathcal{T}'). \end{aligned}$$

It now follows that (ii) in the definition of an L-reduction also holds with $\beta = 1$. This completes the proof of the theorem. □

Theorem 2.1 immediately follows from Corollary 3.2 and Theorem 3.4. Moreover, because $\alpha = \beta = 1$ in the L-reduction from Maximum-Acyclic-Agreement Forest to Minimum Hybridization, Corollary 2.2 is an immediate consequence of Corollaries 3.2 and 3.5.

Chlebík and Chlebíková [6] recently showed that, unless P=NP, there is no polynomial-time approximation algorithm for Max-4DM-2 with an approximation ratio better than $\frac{48}{47}$. Using the L-reduction in the proof of Theorem 3.4 and, in particular, the values $\alpha = 44$ and $\beta = 1$, we get Corollary 3.5.

Corollary 3.5. *Unless* P=NP*, there is no polynomial-time approximation algorithm for* Maximum-Acyclic-Agreement Forest *with an approximation ratio better than* $\frac{2113}{2112}$.

Proof. Suppose that there is such an algorithm and suppose that P $\neq$ NP. Then using the notation and terminology in the proof of Theorem 3.4, we have

$$\begin{aligned} & \frac{S_2}{m_a(\mathcal{T},\mathcal{T}')} < \frac{2113}{2112} \\ \Leftrightarrow \quad & \frac{S_2 - m_a(\mathcal{T},\mathcal{T}')}{m_a(\mathcal{T},\mathcal{T}')} < \frac{2113}{2112} - 1 = \frac{1}{2112} \end{aligned}$$

But $m_a(\mathcal{T},\mathcal{T}') \leq 44\operatorname{opt}(Q)$, and so

$$\frac{1}{44\operatorname{opt}(Q)} \leq \frac{1}{m_a(\mathcal{T},\mathcal{T}')}.$$

Furthermore, $S_2 - m_a(\mathcal{T}, \mathcal{T}') = \text{opt}(Q) - S_1$. Therefore

$$\begin{aligned} & \frac{1}{44\,\text{opt}(Q)}(\text{opt}(Q) - S_1) < \frac{1}{2112} \\ \Leftrightarrow \quad & 1 - \frac{44}{2112} < \frac{S_1}{\text{opt}(Q)} \\ \Leftrightarrow \quad & \frac{47}{48} < \frac{S_1}{\text{opt}(Q)}. \end{aligned}$$

This last inequality implies that Max-4DM-2 has a polynomial-time approximation algorithm with an approximation ratio better than $\frac{48}{47}$, contradicting Chlebík and Chlebíkova's result. This completes the proof of the corollary. □

Remark. The proof of the L-reduction used could also be applied to give an L-reduction from Max-3DM-2 to Maximum-Acyclic-Agreement Forest with the corresponding values $\alpha' = 27$ and $\beta' = 1$. Although α' is much smaller than the α obtained for Max-4DM-2, the resulting inapproximability ratio is worse since the best known inapproximability result for Max-3DM-2 is only $\frac{95}{94}$ [6].

4. The Rooted Subtree Prune and Regraft Operation

Historically, one of the main tools for understanding and modelling reticulate evolution is a graph-theoretic operation called 'rooted subtree prune and regraft'. The reason for this is that a single rooted subtree prune and regraft operation can be used to model a single reticulation event (see [2, 10, 13, 14, 18]). Moreover, for a pair of rooted binary phylogenetic X-trees, the 'rooted subtree prune and regraft distance' between the two trees provides a lower bound to $h(\mathcal{T}, \mathcal{T}')$ (see [3, 19]). It is stated, but not verified, in [11] that computing this distance is APX-hard. In this section, we verify this result and also show that, unless P=NP, there is no polynomial-time approximation algorithm for computing this distance with an approximation ratio better than $\frac{2113}{2112}$. As we will soon see, it is no coincidence that this ratio is the same as that in Corollary 2.2.

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree. As in the definition of an agreement forest, for the purposes of the upcoming definition, we regard the root of $\mathcal{T}$ as a vertex ρ at the end of a pendant edge (called the *root edge*) adjoined to the original root. Let $e = \{u, v\}$ be an edge of $\mathcal{T}$ that is not the root edge, where u is the vertex that is in the path from the root of $\mathcal{T}$ to v. Let $\mathcal{T}'$ be the rooted binary phylogenetic tree obtained from $\mathcal{T}$ by deleting e and then adjoining a new edge f between v and the component C_u that contains u as follows. Create a new vertex u' which subdivides an edge in C_u, and adjoin f between u' and v, and then contract the degree-two vertex u. We say that $\mathcal{T}'$ has been obtained from $\mathcal{T}$ by a *rooted subtree prune and regraft* (rSPR) operation. We define the rSPR *distance* between two arbitrary rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ to be the minimum number of rooted subtree prune and regraft operations that is required to transform $\mathcal{T}$ into $\mathcal{T}'$. This distance is denoted by $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$. It is well-known that, for any such pair of trees, one can always obtain one from the other by a sequence of single rSPR

operations. Thus this distance is well-defined. We formally state the optimization problem of computing rSPR distance between $\mathcal{T}$ and $\mathcal{T}'$ as follows.

Minimum rSPR
Instance: A finite set X, and two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Goal: Find a minimum length sequence of single rSPR operations that transforms $\mathcal{T}$ into $\mathcal{T}'$.
Measure: The length of this sequence.

We remark here the following. Originally thought to be proved in [11], the NP-hardness of Minimum rSPR is established in [5] using the original reduction from "Exact Cover by 3-Sets (X3C)" and revising the definition of maximum-agreement forest given in [11] to that described in this paper. This reduction takes an instance of X3C and converts it into a pair of rooted binary phylogenetic trees with the same label sets for which the instance has an exact cover if and only if the two trees has an agreement forest of a certain size. The reduction used in the proof of Theorem 4.3 (see below) closely follows this original reduction with Max-4DM-2 replacing the closely related problem X3C.

Analogous to the APX-hardness proof of Minimum Hybridization, we prove the APX-hardness of Minimum rSPR in two steps. The first step is by showing that there is an L-reduction from Max-4DM-2 to Maximum-Agreement Forest.

Maximum-Agreement Forest
Instance: A finite set X, and two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Goal: Find a maximum-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$.
Measure: The number of components in $\mathcal{F}$ minus one.

Note that there is no reference to "acyclic" in this problem. The second step is by showing that there is an L-reduction from Maximum-Agreement Forest to Minimum rSPR. Because of Lemma 3.3(ii) and the necessary direction of Lemma 3.3(i) for agreement forests, the proofs used to establish Theorem 3.1 and Corollary 3.2 can be used to establish the first step and in particular the following theorem.

Theorem 4.1. *The optimization problem* Maximum-Agreement Forest *is* APX*-hard. Furthermore, unless* P=NP*, there is no polynomial-time approximation algorithm for* Maximum-Agreement Forest *with an approximation ratio better than* $\frac{2113}{2112}$.

Now for the second step. Like the value $h(\mathcal{T}, \mathcal{T}')$, the value $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ can be written in terms of agreement forests. Recall that $m(\mathcal{T}, \mathcal{T}')$ denotes the size of an agreement forest with the smallest number of components over all agreement forests for $\mathcal{T}$ and $\mathcal{T}'$ minus one. The following theorem is established in [5].

Theorem 4.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then*

(i) $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$.

(ii) *If we have a sequence of single* rSPR *operations that transforms* $\mathcal{T}$ *into* $\mathcal{T}'$*, then there is a polynomial-time algorithm for converting this sequence into an agreement forest* $\mathcal{F}$ *for* $\mathcal{T}$ *and* $\mathcal{T}'$*. Furthermore, if this sequence has length* s*, then*

$$(|\mathcal{F}| - 1) - m(\mathcal{T}, \mathcal{T}') \leq s - d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}').$$

Remark. Part (ii) in Theorem 4.2 is not explicitly stated in [5], but it is essentially a consequence of the inductive proof of [5, Theorem 2.1].

As a consequence of Theorem 4.2, there is an L-reduction from Maximum-Agreement Forest to Minimum rSPR with $\alpha = 1$ and $\beta = 1$. Together with Theorem 4.1, this implies the next theorem, the first part of which verifies a result that is stated without proof in [11].

Theorem 4.3. *The optimization problem* Minimum rSPR *is* APX*-hard. Furthermore, unless* P=NP*, there is no polynomial-time approximation algorithm for* Minimum rSPR *with an approximation ratio better than* $\frac{2113}{2112}$.

We end this section by considering what approximation ratios can be achieved in polynomial time for Minimum Hybridization and Minimum rSPR. Currently, we do not know of any polynomial-time approximation algorithm for Minimum Hybridization. However, based upon ideas in [11, 16], the current best polynomial-time approximation algorithm for Minimum rSPR is a 5-approximation algorithm by Bonet *et al.* [4]. Intuitively, this algorithm builds an agreement forest locally. One might hope that this algorithm extends to Minimum Hybridization, but, due to the additional global condition on a acyclic-agreement forest, it seems unlikely that such an approach will work.

5. Perfect Phylogenetic Networks with Recombination

Perfect phylogenetic network with recombination is a problem that has a very similar flavour to that of Minimum Hybridization, and has been studied by Gusfield *et al.* [8] and Wang *et al.* [20]. Like Minimum Hybridization, the goal of this problem is to compute the minimum number of hybridization events that is required to explain a given input, where in this case the input is a collection of binary sequences. It is shown in [20] that perfect phylogeny with recombination is NP- and APX-hard, however, an assertion in the NP-hardness proof is incorrect. In terms of the language used in this paper, this assertion states that if the rooted subtree prune and regraft distance of two rooted binary phylogenetic trees is k, then there is a hybrid phylogeny with k hybridization vertices each of in-degree two that displays both trees. In [3], explicit examples are given to show that this does not always hold. In this section, we verify the NP- and APX-hardness of the perfect phylogenetic network with recombination problem using the hardness results of Minimum Hybridization.

Although perfect phylogenetic network with recombination could be stated in terms of hybrid phylogenies, we formally state the problem in the language given in

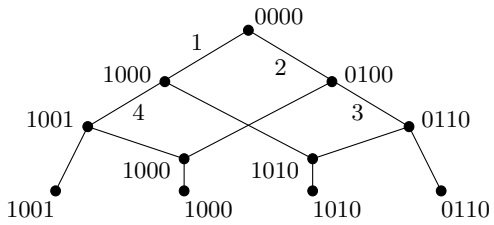

FIGURE 6. A phylogenetic network.

[8, 20]. An (n, m)-*phylogenetic network* $\mathcal{N}$ is a rooted acyclic digraph with exactly n vertices of out-degree zero in which each vertex other than the root has either one or two incoming edges, and each vertex of $\mathcal{N}$ is labelled with a binary sequence of length m. A vertex with two incoming edges is called a *recombination* vertex. Each integer in $\{1, 2, \ldots, m\}$ is assigned to exactly one edge of $\mathcal{N}$ that is not directed towards a recombination vertex. Beginning with the root which is labelled with the all-0 sequence, each of the binary sequences labelling the other vertices is based on the binary sequence of its parent and the incoming edge (in the case it is a non-recombination vertex) or its parents (in the case it is a recombination vertex). In particular, the sequences satisfy the following properties:

(I) If v is a non-recombination vertex with incoming edge e, then the sequence labelling v is obtained from the sequence labelling its parent by changing the i-th element from 0 to 1 for each integer i assigned to e. If no integer is assigned to e, then the sequence labelling v is the same as its parent.
(II) If v is a recombination vertex, then, for some positive integer p strictly between 1 and m (that is, $1 < p < m$), the sequence labelling v is the concatenation of the first p elements of the sequence labelling one of its parents and the last $m - p$ elements of its other parent.

As an example, a phylogenetic network is shown in Fig. 6. For each recombination vertex in this example, the first two elements in the associated sequence come from its 'left' parent and the second two elements come from its 'right' parent.

Let B be a collection of n binary sequences of length m. An (n, m)-phylogenetic network $\mathcal{N}$ *explains* B if the n vertices of out-degree zero are bijectively labelled with the elements of B. For example, the phylogenetic network in Fig. 6 explains the collection $\{1001, 1000, 1010, 0110\}$ of binary sequences.

Over all phylogenetic networks that explain B, we are interested in finding one with the smallest number of recombination vertices. We denote this smallest number by $r(B)$. The perfect phylogenetic network with recombination problem is formally stated as follows:

PERFECT PHYLOGENY WITH RECOMBINATION
Instance: A set B of n binary sequences of length m.
Goal: Find a (n, m)-phylogenetic network $\mathcal{N}$ that explains B with the minimum number of recombination vertices.
Measure: The number of recombination vertices in $\mathcal{N}$.

The motivation for Perfect Phylogeny with Recombination is similar to that for Minimum Hybridization except that, rather than having an input collection consisting of rooted binary phylogenetic trees, we now have an input collection consisting of binary sequences. Each sequence represents a present-day species and, in such a sequence, each coordinate represents some attribute (or character) of the species. A 1 usually indicates that the species under consideration has this particular attribute, while a 0 indicates that the species does not have this attribute. Observe that $0 \rightarrow 1$ is the only allowable transition. The reason for the wording "perfect phylogeny" is that the classical perfect phylogeny problem can be interpreted as the problem of deciding if there is a phylogenetic network with no recombination vertices that explains B.

As mentioned at the beginning of this section, the proof in [20] that establishes the NP-hardness of Perfect Phylogeny with Recombination uses an incorrect assertion. However, the result itself is correct as we next show.

To prove the NP-hardness of Perfect Phylogeny with Recombination, we use a reduction from Minimum Hybridization. We remark here that, even if the NP-hardness proof in [20] was correct, it appears that there is no simple reduction from Perfect Phylogeny with Recombination to Minimum Hybridization. Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, where $|X| = n$. For $\mathcal{T}$ and $\mathcal{T}'$, bijectively label the edges with the elements of $\mathcal{C} = \{\chi_1, \chi_2, \ldots, \chi_{2(n-1)}\}$ and $\mathcal{C}' = \{\chi'_1, \chi'_2, \ldots, \chi'_{2(n-1)}\}$, respectively. Note that both $\mathcal{T}$ and $\mathcal{T}'$ have $2(n-1)$ edges. Each of the elements in $\mathcal{C}$ and $\mathcal{C}'$ represent a binary character with states 0 and 1. For each vertex v and v' of $\mathcal{T}$ and $\mathcal{T}'$, respectively, we associate the binary sequence in which the i-th element is 1 if and only if χ_i (resp. χ'_i) labels an edge on the path from v to the root of $\mathcal{T}$ (resp. $\mathcal{T}'$). For each x in X, concatenate the sequence labelling x in $\mathcal{T}$ with the sequence labelling x in $\mathcal{T}'$. Let B be the resulting collection of n binary sequences of length $4(n-1)$. This construction is the same as that originally used in [20]. The following lemma is central to proving the NP-hardness (and APX-hardness) of Perfect Phylogeny with Recombination.

Lemma 5.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let B be a collection of binary sequences that is constructed from $\mathcal{T}$ and $\mathcal{T}'$ as above. Then*

$$r(B) = h(\mathcal{T}, \mathcal{T}').$$

Proof. We first show that $r(B) \leq h(\mathcal{T}, \mathcal{T}')$. Let $\mathcal{H}$ be a hybrid phylogeny on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and has the property that $h(\mathcal{H})$ is minimized. Let ρ denote the root of $\mathcal{H}$. Because of minimality and the fact that we have only two trees, each hybridization vertex of $\mathcal{H}$ has in-degree two. By deleting and contracting edges if necessary, we may assume that all the edges of $\mathcal{H}$ are used in some simultaneous displaying of $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, by refining vertices if necessary, we may also assume that if a vertex in $\mathcal{H}$ has in-degree two, then it has out-degree one. Now colour each vertex and edge of $\mathcal{H}$ green or red depending upon whether it is used by $\mathcal{T}$ or $\mathcal{T}'$, respectively, under the simultaneous displaying of $\mathcal{T}$ and $\mathcal{T}'$. Every vertex and edge is coloured with at least one colour. We will call a vertex or edge *monochromatic* if it is only coloured with one colour; otherwise we call it

bichromatic. We force the root of $\mathcal{H}$ to be bichromatic as follows. In the case that the root of one of the trees, $\mathcal{T}'$ say, is identified with a non-root vertex of $\mathcal{H}$, we will colour ρ and the edges of a directed path from ρ to this non-root vertex of $\mathcal{H}$ red, and view this path as part of $\mathcal{T}'$. The reason for this will be made clear soon. We next assign a binary sequence to each vertex of $\mathcal{H}$ based on this colouring.

As in the case of the sequences in B, the labelling comes in two parts. The root ρ is given the all-0 sequence. Consider the restriction of $\mathcal{H}$ to the green vertices and edges. For each green vertex $v \neq \rho$, assign it the first part of the sequence labelling the vertex of $\mathcal{T}$ corresponding to v. If v has degree two in this restriction, assign it the labelling of the first vertex 'above' it that has degree three or, in the case this vertex is the root, degree two. Now consider the restriction of $\mathcal{H}$ to the red vertices and edges. For each red vertex $v \neq \rho$, assign it the second part of the sequence labelling the vertex of $\mathcal{T}'$ corresponding to v. If v has degree two in this restriction, assign it the labelling of the first vertex 'above' it that has degree three or, in the case this vertex is the root, degree two. After this labelling, all of the bichromatic vertices of $\mathcal{H}$ have been assigned a sequence with both parts. If v is a monochromatic vertex of $\mathcal{H}$ coloured green, then the second part of its sequence label is the same as the second part of the sequence labelling the first bichromatic vertex that is met on the unique green path from v to ρ. Furthermore, if v is a monochromatic vertex of $\mathcal{H}$ coloured red, then the first part of its sequence label is the same as the first part of the sequence labelling the first bichromatic vertex that is met on the unique red path from v to ρ. Since ρ is bichromatic, this is well-defined.

This direction of the proof is completed by showing that $\mathcal{H}$ with this sequence labelling of the vertices is a phylogenetic network $\mathcal{N}$ that explains B. Clearly, there is a one-to-one correspondence between the elements of B and the vertices of $\mathcal{N}$ of out-degree zero. Furthermore, as $\mathcal{H}$ has the property that the out-degree of each hybridization vertex v is one, and the edges directed into v are different colours and monochromatic, the sequence assigned to v is of the type described in (II) of the definition of a phylogenetic network. Because of the way in which the elements in B are constructed and the way in which the sequences are assigned to the vertices of $\mathcal{H}$ from the sequences labelling the vertices of $\mathcal{T}$ and $\mathcal{T}'$, it is now easily seen that $\mathcal{N}$ is a phylogenetic network that explains B. Hence $r(B) \leq h(\mathcal{T}, \mathcal{T}')$.

To show that $r(B) \geq h(\mathcal{T}, \mathcal{T}')$, we can use Claim 2 in the second part of the proof of Theorem 1 in [20] which implies that if there is a phylogenetic network $\mathcal{N}$ that explains B and has k recombination vertices, then the underlying rooted acyclic digraph can be modified to give a rooted acyclic digraph that displays $\mathcal{T}$ and $\mathcal{T}'$, and has k recombination vertices, where each recombination vertex has in-degree two. In particular, there is a hybrid phylogeny $\mathcal{H}$ on X that displays $\mathcal{T}$ and $\mathcal{T}'$ with $h(\mathcal{H}) = k$. Thus $r(B) \geq h(\mathcal{T}, \mathcal{T}')$. □

The NP-hardness of Perfect Phylogeny with Recombination follows immediately from the next theorem.

Theorem 5.2. *The optimization problem* Perfect Phylogeny with Recombination *is* APX*-hard.*

Proof. Because of the strength of Lemma 5.1, the proof is straightforward. Let $\mathcal{T}$ and $\mathcal{T}'$ be an instance I of Minimum Hybridization, and let $f(I)$ be the function that maps $\mathcal{T}$ and $\mathcal{T}'$ to B, an instance of Perfect Phylogeny with Recombination as described prior to Lemma 5.1. Evidently, this mapping takes polynomial time in the size of $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, by Lemma 5.1, $r(B) = h(\mathcal{T}, \mathcal{T}')$ and so (i) in the definition of an L-reduction holds with $\alpha = 1$.

Now let $\mathcal{N}$ be a phylogenetic network that explains B with S_2 recombination vertices. Let g be the function that maps $\mathcal{N}$ to the feasible solution of $\mathcal{T}$ and $\mathcal{T}'$ of size $S_1 = S_2$ as described in the last paragraph of the proof of Lemma 5.1. Note that, as detailed in [20], this mapping can be computed in polynomial time. As $r(B) = h(\mathcal{T}, \mathcal{T}')$, it follows that

$$S_1 - h(\mathcal{T}, \mathcal{T}') = S_2 - r(B).$$

Thus (ii) in the definition of an L-reduction holds with $\beta = 1$. □

The proof of Corollary 5.3 is analogous to that used to prove Corollary 2.2. We omit the details.

Corollary 5.3. *Unless* P=NP*, there is no polynomial-time approximation algorithm for* Perfect Phylogeny with Recombination *with an approximation ratio better than* $\frac{2113}{2112}$.

Acknowledgments

We thank the referees for their helpful comments and, in particular, for pointing out an oversight in one of our original reductions and recognizing a slight sharpening of the inapproximability results.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, Complexity and Approximation (Springer, Berlin, 1999).

[2] M. Baroni, C. Semple, and M. Steel, A framework for representing reticulate evolution, Ann. Combin. 8 (2004) 391-408.

[3] M. Baroni, S. Grünewald, V. Moulton, and C. Semple, Bounding the number of hybridization events for a consistent evolutionary history, Journal of Mathematical Biology 51 (2005) 171-182.

[4] M. K. Bonet, K. St. John, R. Mahindru, and N. Amenta, Approximating subtree distances between phylogenies, Technical Report #669 (Centre de Recerca Matematica, Barcelona, 2006).

[5] M. Bordewich and C. Semple, On the computational complexity of the rooted subtree prune and regraft distance, Ann. Combin. 8 (2004) 409-423.

[6] M. Chlebík and J. Chlebíková, Inapproximability results for bounded variants of optimization problems, in: A. Lingas and B. J. Nilsson, eds, Fundamentals of Computation Theory, 14th International Symposium (FCT), Lecture Notes in Computer Science, Vol. 2751 (Springer-Verlag, 2003) 27-38.

[7] S. Grünewald, Private communication.

[8] D. Gusfield, S. Eddhu, and C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinformatics and Computational Biology 2 (2004) 173-213.

[9] E. Hazan, S. Safra and O. Schwartz, On the hardness of approximating k-dimensional matching, ECCC Report TR03-20, (2003).
[10] J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, Math. Biosci. 98 (1990) 185-200.
[11] J. Hein, T. Jing, L. Wang, K. Zhang, On the complexity of comparing evolutionary trees, Discrete Appl. Math. 71 (1996) 153-169.
[12] V. Kann, Maximum bounded 3-dimensional matching is MAX SNP-complete, Inform. Process. Lett. 37 (1991) 27-35.
[13] W. Maddison, Gene trees in species trees, Syst. Biol. 46 (1997) 523-536.
[14] L. Nakhleh, T. Warnow, C. Randal Linder, and K. St. John Reconstructing reticulate evolution in species - theory and practice, J. Computational Biology 12 (2005) 796-811.
[15] C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425-440.
[16] E. M. Rodrigues, M.-F. Sagot, and Y. Wakabayashi, Some approximation results for the maximum agreement forest problem, in: M. Goemans *et al.*, eds, Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques (APPROX and RANDOM), Lecture Notes in Computer Science, Vol. 2129 (Springer, Berlin, 2001) 159-169.
[17] C. Semple and M. Steel, Phylogenetics (Oxford University Press, 2003).
[18] Y. Song and J. Hein, Parsimonious reconstruction of sequence evolution and haplotyde blocks: finding the minimum number of recombination events, in: G. Benson and R. Page, eds, Algorithms in Bioinformatics (WABI), Lecture Notes in Bioinformatics, Vol. 2812 (Springer, Berlin, 2003) 287-302.
[19] Y. Song and J. Hein, Constructing minimal ancestral recombination graphs, J. Comp. Biol. 12 (2005) 147-169.
[20] L. Wang, K. Zhang, and L. Zhang, Perfect phylogenetic networks with recombination, J. Computational Biology 8 (2001) 69-78.

Department of Computer Science, Durham University, Durham DH1 3LE, United Kingdom

E-mail address: m.j.r.bordewich@durham.ac.uk

Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

E-mail address: c.semple@math.canterbury.ac.nz