Deciding the existence of a cherry-picking sequence is hard on two trees

Döcker, Janosch ; van Iersel, Leo; Kelk, Steven; Linz, Simone

DOI
10.1016/j.dam.2019.01.031

Publication date
2019

Document Version
Final published version

Published in
Discrete Applied Mathematics

Citation (APA)
Döcker, J., van Iersel, L., Kelk, S., & Linz, S. (2019). Deciding the existence of a cherry-picking sequence is hard on two trees. *Discrete Applied Mathematics*, *260*, 131-143. https://doi.org/10.1016/j.dam.2019.01.031

Discrete Applied Mathematics 260 (2019) 131–143

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Deciding the existence of a cherry-picking sequence is hard on two trees

Janosch Döcker [a], Leo van Iersel [b], Steven Kelk [c,*], Simone Linz [d]

[a] Department of Computer Science, University of Tübingen, Germany
[b] Delft Institute of Applied Mathematics, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands
[c] Department of Data Science and Knowledge Engineering (DKE), Maastricht University, The Netherlands
[d] School of Computer Science, University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 30 November 2017
Received in revised form 20 October 2018
Accepted 22 January 2019
Available online 13 February 2019

Keywords:
Phylogenetics
NP-hardness
Satisfiability
Phylogenetic networks
Elimination orders
Temporal networks

ABSTRACT

Here we show that deciding whether two rooted binary phylogenetic trees on the same set of taxa permit a *cherry-picking sequence*, a special type of elimination order on the taxa, is NP-complete. This improves on an earlier result which proved hardness for eight or more trees. Via a known equivalence between cherry-picking sequences and temporal phylogenetic networks, our result proves that it is NP-complete to determine the existence of a temporal phylogenetic network that contains topological embeddings of both trees. The hardness result also greatly strengthens previous inapproximability results for the minimum temporal-hybridization number problem. This is the optimization version of the problem where we wish to construct a temporal phylogenetic network that topologically embeds two given rooted binary phylogenetic trees and that has a minimum number of indegree-2 nodes, which represent events such as hybridization and horizontal gene transfer. We end on a positive note, pointing out that fixed parameter tractability results in this area are likely to ensure the continued relevance of the temporal phylogenetic network model.

1. Introduction

In the field of phylogenetics it is common to represent the evolution of a set of species X by a rooted phylogenetic tree; essentially a rooted, bifurcating tree whose leaves are bijectively labeled by X [21]. Driven by the realization that evolution is not always treelike there has been growing attention for the construction of phylogenetic *networks*, which generalize phylogenetic trees to directed acyclic graphs [1,11,14,22]. One well known optimization problem for phylogenetic networks is as follows: given a set of rooted phylogenetic trees $\mathcal{T}$ on the same set of taxa X, compute a phylogenetic network $N = (V, E)$ which *displays* (i.e. contains topological embeddings of) all the trees in $\mathcal{T}$, such that the *reticulation number* $|E| - (|V| - 1)$ is minimized. When N is restricted to being *binary* this is equivalent to minimizing the number of nodes of N with indegree-2. This optimization model is known as *minimum hybridization* and it has been extensively studied in the last decade (see e.g. [2,6,15,18,24]). More recently variations of minimum hybridization have been proposed which constrain the topology of N to be more biologically relevant. One such constraint is to demand that N is *temporal* [19]. Informally, a phylogenetic network N is temporal if (i) the nodes of N can be labeled with times, such that nodes of indegree-2 have contemporaneous parents, and time moves strictly forwards along treelike parts of the network; and (ii) each non-leaf vertex has a child whose

[*] Corresponding author.
E-mail addresses: janosch.doecker@uni-tuebingen.de (J. Döcker), L.J.J.vanIersel@tudelft.nl (L. van Iersel), steven.kelk@maastrichtuniversity.nl (S. Kelk), s.linz@auckland.ac.nz (S. Linz).

https://doi.org/10.1016/j.dam.2019.01.031
0166-218X/© 2019 Elsevier B.V. All rights reserved.

indegree is 1. Property (ii) by itself is referred to as *tree-child* in the literature [5]. It has been shown that when $|\mathcal{T}| = 2$ it is NP-hard to solve the minimum temporal-hybridization number problem to optimality [13]. To establish the result, the authors proved that the problem is in fact APX-hard, which implies that for some constant $c > 1$ it is not possible in polynomial time to approximate the optimum within a factor of c, unless P = NP [20].

A more fundamental question remained, however, open: is it possible in polynomial time to determine if *any* temporal phylogenetic network exists that displays the input trees, regardless of how large $|E| - (|V| - 1)$ is [12,23]? Here we settle this question by showing that, even for $|\mathcal{T}| = 2$, it is NP-complete to determine whether such a network exists. We prove this by using the *cherry-picking* characterization of temporal phylogenetic networks introduced in [12]. There it was shown that, given an arbitrarily large set $\mathcal{T}$ of rooted binary phylogenetic trees on X, there exists a temporal phylogenetic network that displays each tree in $\mathcal{T}$ precisely if $\mathcal{T}$ has a so-called cherry-picking sequence. Informally, a *cherry-picking sequence* on $\mathcal{T}$ is an elimination order on X that deletes one element of X at a time, where at each step only elements can be deleted which are in a cherry of every tree in $\mathcal{T}$ [12]. We show here that the seminal NP-complete problem 3-SAT [17] can be reduced to the question of whether two trees permit a cherry-picking sequence. This improves upon a recent result by two of the present authors which shows that, for $|\mathcal{T}| \geq 8$, it is NP-complete to determine whether $\mathcal{T}$ has a cherry-picking sequence [8]. Our hardness result is highly non-trivial and requires extensive gadgetry; to clarify we include an explicit example of the construction after the main proof.

As we discuss in the final section of the paper, this result has quite significant negative consequences: given that the decision problem is already hard, the minimum temporal-hybridization number problem is in some sense "effectively inapproximable", even for two trees. This greatly strengthens the earlier APX-hard inapproximability result. Nevertheless, as we subsequently point out, positive fixed parameter tractability (FPT) [7] results for the minimum temporal-hybridization number problem do already exist [12] and our results emphasize the importance of further developing such algorithms, since fixed parameter tractability forms the most promising remaining avenue towards practical exact methods.

2. Preliminaries

A *rooted binary phylogenetic tree* on a set of taxa X, where $|X| \geq 2$, is a rooted, connected, directed tree with a unique *root* (a vertex of indegree-0 and outdegree-2), where the leaves (vertices with indegree-1 and outdegree-0) are bijectively labeled by X, and where all interior vertices of the tree are indegree-1 and outdegree-2. If $|X| = 1$, we consider the single isolated node labeled by the unique element of X, to also be a rooted binary phylogenetic tree. Since all phylogenetic trees considered in this paper are rooted and binary, we henceforth write *tree* for brevity, and draw no distinction between the elements of X and the leaves they label. Let T be a tree, and let $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ be a set of trees. We use $X(T)$ to denote the taxa set of T and, similarly, we use $X(\mathcal{T})$ to denote the union of taxa sets over all elements in $\mathcal{T}$, i.e. $X(\mathcal{T}) = X(T_1) \cup X(T_2) \cup \cdots \cup X(T_m)$. Lastly, for two distinct elements x and y in X, we call $\{x, y\}$ a *cherry* of T if they have the same parent. A tree with a single cherry is referred to as a *caterpillar*.

Now, let T be a tree on X, and let $X' = \{x_1, x_2, \dots, x_k\}$ be an arbitrary set. We write $T|X'$ to denote the tree obtained from T by taking the minimum subtree spanning the elements of X' and repeatedly suppressing all vertices with indegree-1 and outdegree-1. (If v is a vertex with indegree-1 and outdegree-1, with incident edges (u, v) and (v, w), then *suppressing* v is achieved as follows: v and its two incident edges are deleted, and an edge (u, w) is added.)

Furthermore, we also write $T[-x_1, x_2, \dots, x_k]$ or $T[-X']$ for short to denote $T|(X - X')$. If $X \cap X' = \emptyset$, then $T|X'$ is the null tree and $T[-X']$ is T itself. For a set $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ of trees on subsets of X, we write $\mathcal{T}|X'$ (resp. $\mathcal{T}[-X']$) when referring to the set $\{T_1|X', T_2|X', \dots, T_m|X'\}$ (resp. $\{T_1[-X'], T_2[-X'], \dots, T_m[-X']\}$). Lastly, a rooted binary phylogenetic tree is *pendant* in T if it can be detached from T by deleting a single edge.

2.1. Cherry-picking sequence problem on trees with the same set of taxa

We say that a taxon $x \in X$ is *in* a cherry of T if there exists some $y \neq x$ such that $\{x, y\}$ is a cherry of T or T consists of a single leaf x. If x is in a cherry of T, we say that x is *picked* (or *pruned*) from T to denote the operation of replacing T with $T[-x]$. Given a set of trees $\mathcal{T}$, all on the same set of taxa X, we say that a taxon $x \in X$ is *available (for picking)* in $\mathcal{T}$ if x is in a cherry in each tree in $\mathcal{T}$. When this is the case, we say that x is *picked* (or *pruned*) from $\mathcal{T}$ to denote the operation of replacing $\mathcal{T}$ with $\mathcal{T}[-x]$.

Let $\mathcal{T}$ be a set of trees on the same set of taxa X. A *cherry-picking sequence* is an order on X, say $(x_1, x_2, \dots, x_{|X|})$, such that each x_i with $i \in \{1, 2, \dots, |X|\}$ is available in $\mathcal{T}[-x_1, x_2, \dots, x_{i-1}]$. Such a sequence is not guaranteed to exist; if it does, we say that $\mathcal{T}$ *permits* a cherry-picking sequence. It was shown in [8] that deciding whether such a sequence exists is NP-complete if $|\mathcal{T}| \geq 8$. Note that, if $|\mathcal{T}| = 1$, then $\mathcal{T}$ always has a cherry-picking sequence. To illustrate, a cherry-picking sequence for the two trees that are shown at the top of Fig. 1 is (f, g, d, b, a, c, e).

2.2. A more general cherry-picking sequence problem

Let $\mathcal{T}$ be a set of trees, and let $X = X(\mathcal{T})$. Suppose we consider the variant of the problem described in Section 2.1 in which the trees in $\mathcal{T}$ do not necessarily have the same set of taxa. In this case, some taxa may be missing from some trees. This requires us to generalize the concept of being *in* a cherry of a tree. We say that a taxon x is *in* a cherry of a tree T, if exactly one of the following conditions holds:

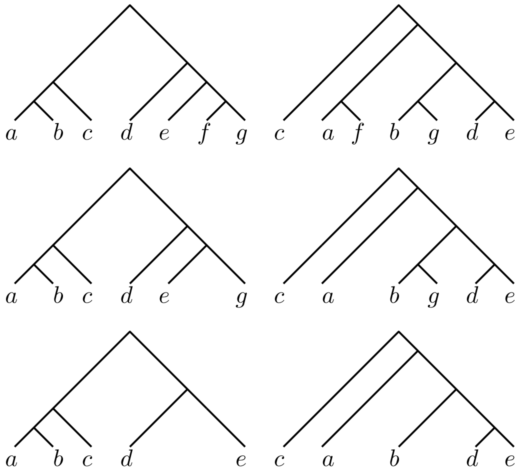

Fig. 1. A cherry-picking sequence for the two trees T and T' at the top is (f, g, d, b, a, c, e). The two trees in the middle have been obtained from T and T', respectively, by pruning f, and the two trees at the bottom have been obtained from T and T' by first pruning f and, subsequently, pruning g. While we can alternatively prune a and, subsequently, b, from T and T', note that no cherry-picking sequence exists for T and T' whose first two elements are a and b.

(1) $x \notin X(T)$ or
(2) $x \in X(T)$ and T has a cherry $\{x, y\}$, where x and y are distinct elements in $X(T)$.

(Note that, once again, this means that if x is the only taxon in T, then x is vacuously considered to be in a cherry of T.) It initially seems counter-intuitive to say, when condition 1 applies, that x is "in" a cherry of T. However, the idea behind this is that such trees do not constrain whether x can be picked; they "do not care". More formally, we say that a taxon x is *available* in $\mathcal{T}$ if it is in a cherry in each tree in $\mathcal{T}$. Similar to Section 2.1, we say that an order on X, say $(x_1, x_2 \ldots, x_{|X|})$ is a *cherry-picking sequence* of $\mathcal{T}$ if each x_i with $i \in \{1, 2, \ldots, |X|\}$ is available in $\mathcal{T}[-x_1, x_2, \ldots, x_{i-1}]$. If a tree becomes the null tree due to all its taxa being pruned away then this tree plays no further role. Moreover we note that, if all trees in $\mathcal{T}$ have the same set of taxa, then the more general definition of a cherry-picking sequence given in this subsection and that will be used throughout the rest of this paper coincides with that given in Section 2.1.

3. Main results

In this section, we establish the main result of this paper. We start with two lemmas.

Lemma 1. *Let $\mathcal{T}$ be a set of m trees on not necessarily the same set of taxa. Then we can construct in polynomial time a set $\mathcal{T}'$ of m trees all on the same set of taxa, such that $\mathcal{T}$ has a cherry-picking sequence if and only if $\mathcal{T}'$ does.*

Proof. Let $X = X(\mathcal{T})$, and let $Y = \{y_1, y_2, \ldots\}$ be the set of taxa that are missing from at least one input tree. Let $Y' = \{y'_1, y'_2, \ldots\}$ be a disjoint copy of this set. Every modified tree will have taxon set $X \cup Y' \cup \{\rho\}$. The idea is as follows. Let $T_{Y'}$ be an arbitrary rooted binary tree on Y'. For each input tree T_i, we start by joining T_i and ρ beneath a root, and then join this new tree and $T_{Y'}$ together beneath a root. Next, for each $y_j \in Y$ that is missing from T_i, we add y_j by subdividing the edge that feeds into y'_j and attaching y_j there (so y_j and y'_j become siblings). For an example, see Fig. 2. We call the set of trees constructed in this way $\mathcal{T}'$. The high-level idea is that if a tree T_i does not contain some taxon x, we attach x just above x' and thus ensure that, trivially, x is in a cherry in that tree (i.e. together with x'). So T_i does "not care" about x and will not obstruct it from being pruned.

First, assume that $\mathcal{T}$ has a cherry-picking sequence σ. (We show that $\mathcal{T}'$ has a cherry-picking sequence.) We start by applying exactly the same sequence of pruning operations to $\mathcal{T}'$. These picking operations will always be possible because, if a taxon $y \in Y$ is missing from a tree $T_i \in \mathcal{T}$, it will be in a cherry together with y' in the corresponding tree of $\mathcal{T}'$. After doing this, all the trees will be isomorphic and have the same set of taxa: $Y' \cup \{\rho\}$. At this point these remaining taxa can be pruned in bottom-up fashion (since two isomorphic trees always have a cherry-picking sequence). Hence $\mathcal{T}'$ has a cherry-picking sequence. Note that the taxon ρ is included to ensure that if, during σ, a tree T_i has been pruned down to a single taxon, this taxon can still be pruned in the corresponding tree of $\mathcal{T}'$ (because it is sibling to ρ).

In the other direction, let σ' be a cherry-picking sequence for $\mathcal{T}'$. Let σ be the sequence obtained by deleting all taxa from σ' that are not in X. Let x be an arbitrary element of X and let i be the position of x in σ'. Let $\ell'_1, \ell'_2, \ldots, \ell'_{i-1}$ be the prefix of σ' that has been pruned from $\mathcal{T}'$ prior to x, and let $\ell_1, \ell_2, \ldots, \ell_j$ (where $j \le i-1$) be the prefix of σ that has been pruned prior to x. We claim that, if x is available in $\mathcal{T}'[-\ell'_1, \ell'_2, \ldots, \ell'_{i-1}]$, then it is also available in $\mathcal{T}[-\ell_1, \ell_2, \ldots, \ell_j]$. To see this, let T be an arbitrary tree in $\mathcal{T}[-\ell_1, \ell_2, \ldots, \ell_j]$. If $x \notin X(T)$, then (by definition) x is in a cherry of T. If x is the only

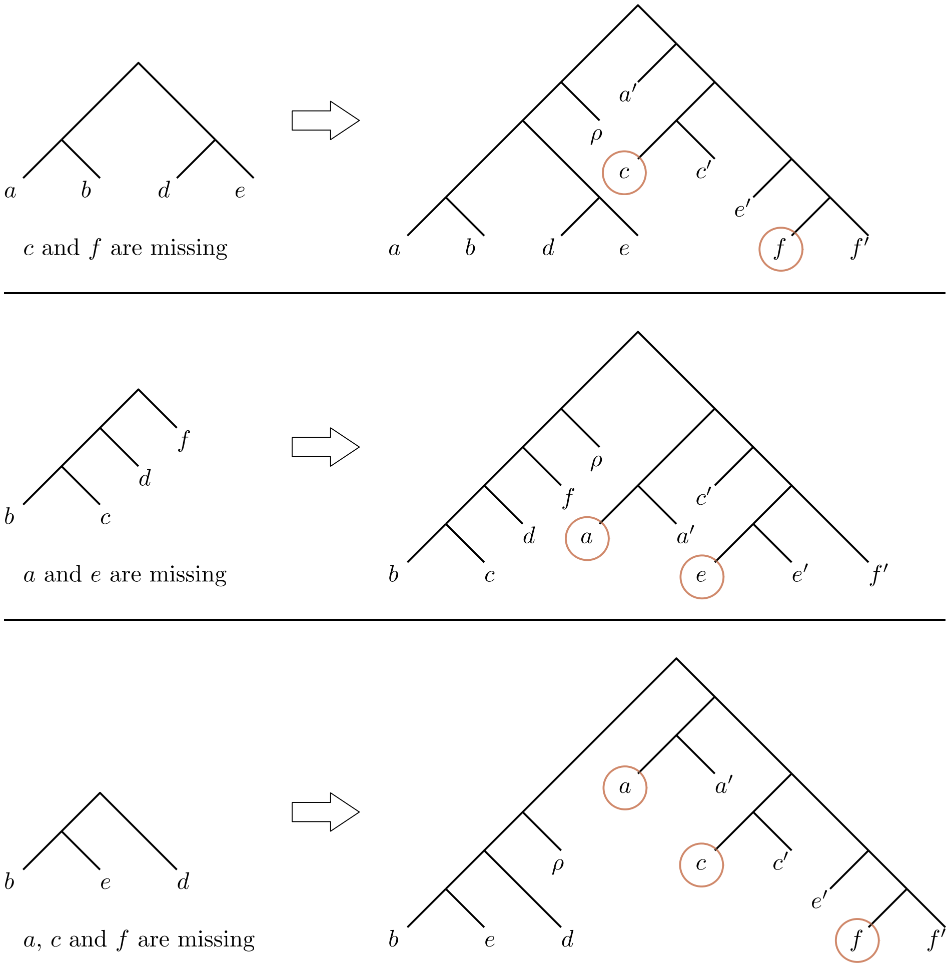

Fig. 2. The construction described in Lemma 1. Here Y, the set of taxa missing from at least one tree, is $\{a, c, e, f\}$. In each modified tree the artificially added members of Y are circled; note that they are always in cherries. A cherry-picking sequence for the original trees is e, b, c, d, a, f. A corresponding sequence for the modified trees is $e, b, c, d, a, f, f', e', c', a', \rho$.

taxon in T, then it is (also by definition) in a cherry. So the only case remaining is that $x \in X(T)$ and $|X(T)| \geq 2$. Let T' be the tree from $\mathcal{T}'[-\ell'_1, \ell'_2, \ldots, \ell'_{i-1}]$ that corresponds to T. The critical observation here is that, by construction, T occurs as a pendant subtree of T'. So if x was not in a cherry of T, then x would not be in a cherry of T' which gives a contradiction to the assumption that $\mathcal{T}'$ has a cherry-picking sequence. Hence, x is in a cherry of T. Due to the arbitrary choice of x and T, it follows that σ is a cherry-picking sequence for $\mathcal{T}$.

It remains to show that the reduction is polynomial time. Observe that, depending on the instance, the size of $\mathcal{T}$ can be dominated by $|X|$ or m. Each of the m trees in $\mathcal{T}'$ contains $|X| + |Y| + 1$ taxa, where $|Y| \leq |X|$, and the transformation itself involves straightforward operations, so overall the reduction takes poly$(|X|, m)$ time. □

Let $\mathcal{T}$ be a set of rooted binary trees, and let T_i and T_j be two trees in $\mathcal{T}$ such that $X(T_i) \cap X(T_j) = \emptyset$. Furthermore, let ρ_i and ρ_j be the root vertex of T_i and T_j, respectively. Obtain a new tree from T_i and T_j in the following way.

(1) Create a new vertex ρ and add new edges $e = (\rho, \rho_i)$ and $e' = (\rho, \rho_j)$.
(2) Subdivide e (resp. e') with a new vertex v (resp. v') and add a new edge (v, x) (resp. (v', y)), where x and y are two new taxa such that $\{x, y\} \cap X(\mathcal{T}) = \emptyset$.

We call the resulting rooted binary tree the *compound tree* of T_i and T_j. To illustrate, Fig. 3 depicts the compound tree of T_i and T_j.

The next lemma shows that, for a set $\mathcal{T}$ of rooted binary trees, the replacement of two trees in $\mathcal{T}$ with their compound tree preserves the existence and non-existence of a cherry-picking sequences for $\mathcal{T}$.

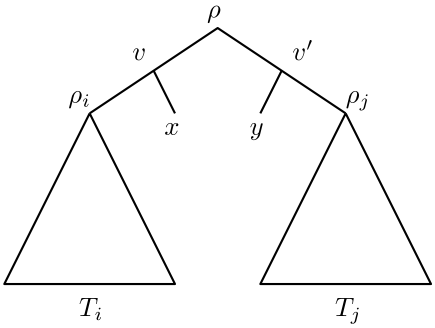

Fig. 3. The compound tree of two rooted binary trees T_i and T_j. The taxon x (resp. y) simply ensures that the last taxon pruned away in the T_i (resp. T_j) part is in a cherry with x (resp. y).

Lemma 2. *Let $\mathcal{T}$ be a set of rooted binary trees, and let T_i and T_j be two trees in $\mathcal{T}$ such that $X(T_i) \cap X(T_j) = \emptyset$. Let $T_{i,j}$ be the compound tree of T_i and T_j. Then $\mathcal{T}$ has a cherry-picking sequence if and only if $(\mathcal{T} - \{T_i, T_j\}) \cup \{T_{i,j}\}$ has a cherry-picking sequence.*

Proof. To ease reading, let $\mathcal{T}' = (\mathcal{T} - \{T_i, T_j\}) \cup \{T_{i,j}\}$. Furthermore, let $|X(\mathcal{T})| = n$, and let x and y be the unique two taxa in $X(T_{i,j})$ that do not label a leaf in T_i or T_j.

Suppose that $\sigma = (\ell_1, \ell_2, \dots, \ell_n)$ is a cherry-picking sequence for $\mathcal{T}$. Let i' be the maximum index of an element in σ such that $\ell_{i'} \in X(T_i)$ and, similarly, let j' be the maximum index of an element in σ such that $\ell_{j'} \in X(T_j)$. Then $\mathcal{T}[-\ell_1, \ell_2, \dots \ell_{i'-1}]$ contains a tree that is a single vertex labeled $\ell_{i'}$ and $\mathcal{T}[-\ell_1, \ell_2, \dots \ell_{j'-1}]$ contains a tree that is a single vertex labeled $\ell_{j'}$. Moreover, by the construction of $T_{i,j}$, the set $\mathcal{T}'[-\ell_1, \ell_2, \dots \ell_{i'-1}]$ contains a tree with cherry $\{\ell_{i'}, x\}$ and the set $\mathcal{T}'[-\ell_1, \ell_2, \dots \ell_{j'-1}]$ contains a tree with cherry $\{\ell_{j'}, y\}$. Since T_i and T_j are pendant subtrees in $T_{i,j}$ and σ is a cherry-picking sequence for $\mathcal{T}$, it now follows that

$$(\ell_1, \ell_2, \dots, \ell_n, x, y)$$

is a cherry-picking sequence for $\mathcal{T}'$.

Conversely, suppose that $\sigma' = (\ell_1, \ell_2, \dots, \ell_{n+2})$ is a cherry-picking sequence for $\mathcal{T}'$. Let $\{\ell_{i'}, \ell_{j'}\} = \{x, y\}$. Without loss of generality, we may assume that $i' < j'$. Then, as x and y are only contained in the leaf set of $T_{i,j}$, it is straightforward to check that

$$(\ell_1, \ell_2, \dots, \ell_{i'-1}, \ell_{i'+1}, \ell_{i'+2}, \dots, \ell_{j'-1}, \ell_{j'+1}, \ell_{j'+2}, \dots, \ell_{n+2})$$

is a cherry-picking sequence for $\mathcal{T}$. □

Now we establish the main result of this paper.

Theorem 1. *It is* NP*-complete to decide if two rooted binary phylogenetic trees T and T' on X have a cherry-picking sequence.*

Proof. Given an order $\sigma = (x_1, x_2, \dots, x_{|X|})$ on X, we can decide in polynomial time if, for each $i \in \{1, 2, \dots, |X|\}$, x_i is in a cherry in $T[-x_1, x_2, \dots, x_{i-1}]$ and $T'[-x_1, x_2, \dots, x_{i-1}]$. Hence, the problem of deciding if T and T' have a cherry-picking sequence is in NP. To establish the theorem, we use a reduction from 3-Sat. This is the variant of Satisfiability where each clause contains *exactly* three literals, and the logical expression is in conjunctive normal form, i.e.,

$$\bigwedge_{i=1}^{m} C_i = \bigwedge_{i=1}^{m} (l_{i,1} \vee l_{i,2} \vee l_{i,3}),$$

where $l_{i,j} \in \{v^{(k)}, \neg v^{(k)} \mid 1 \le k \le n\}$. The corresponding set of variables is denoted with

$$V := \{v^{(1)}, v^{(2)}, \dots, v^{(n)}\}.$$

We reduce from the NP-complete version of 3-Sat in which no variable occurs more than once in a given clause. Such restricted instances can easily be obtained by a standard transformation as described in [10]. In the remainder of this proof, n and m refer to the number of variables and clauses in a restricted 3-Sat instance, respectively.

Now, given an instance I of 3-Sat, we first construct a set $\mathcal{T}$ of $3n + 5m + 2$ trees with overlapping taxa sets and show that I has a satisfying truth assignment if and only if $\mathcal{T}$ has a cherry-picking sequence. We then repeatedly apply Lemma 2 in order to replace $\mathcal{T}$ with two trees and, finally, apply Lemma 1 to complete the proof of this theorem.

We start by describing the construction of $\mathcal{T}$ that makes use of the introduction of a set $\{b_1, b_2, \dots, b_{4n+3m}, b_X, b_Y, b_Z\}$ of blocking taxa. As we will see later, each such taxon can only be pruned from $\mathcal{T}$ after certain other taxa have been

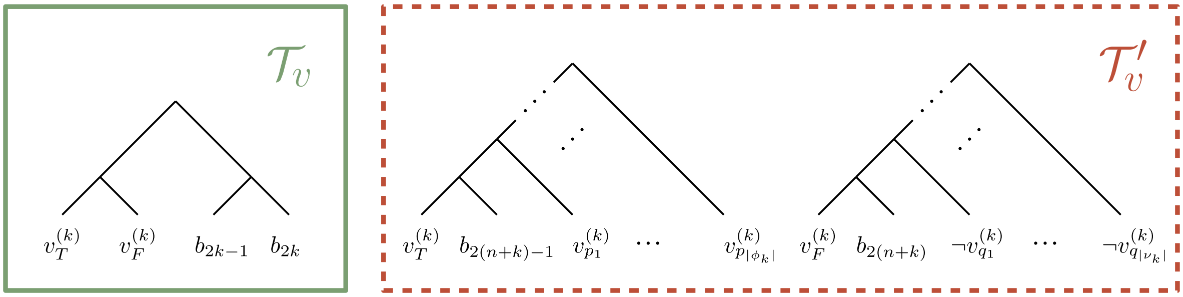

Fig. 4. Each variable $v^{(k)}$, is represented by a single tree in $\mathcal{T}_v$ and two trees in $\mathcal{T}'_v$.

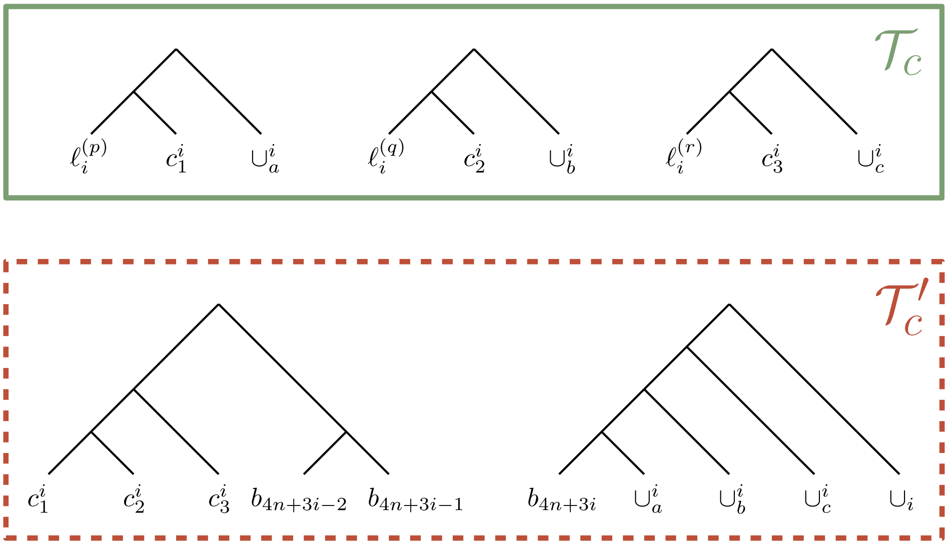

Fig. 5. Each clause C_i is represented by three trees in $\mathcal{T}_c$ and two trees in $\mathcal{T}'_c$.

pruned first and so the main function of the blocking taxa is to be unavailable for pruning which in turn constraints the number of possibilities to construct a cherry-picking sequence from $\mathcal{T}$. An explicit example of the construction of $\mathcal{T}$ is given subsequently to this proof.

Variable gadget. We construct two sets $\mathcal{T}_v$ and $\mathcal{T}'_v$ of trees. Each variable $v^{(k)}$ with $k \in \{1, 2, \ldots, n\}$ adds one tree on four taxa to $\mathcal{T}_v$ which is the tree shown in the solid box of Fig. 4. Each such tree has two blocking taxa and, intuitively, encodes whether $v^{(k)}$ is set to be true or false, depending on whether $v_T^{(k)}$ or $v_F^{(k)}$ is pruned first. Moreover, each variable $v^{(k)}$ adds two caterpillars to $\mathcal{T}'_v$. Relative to a fixed $v^{(k)}$, the precise construction of these caterpillars is based on the definition of two particular tuples. Let $\phi_k := (p_1, p_2, \ldots, p_{|\phi_k|})$ (resp. $\nu_k := (q_1, q_2, \ldots, q_{|\nu_k|})$) be the indices, in ascending order, of all the clauses in which $v^{(k)}$ appears unnegated (resp. negated). Since no clause contains any variable more than once, the elements in ϕ_k (resp. ν_k) are pairwise distinct.

Now the taxon set of one caterpillar contains $v_T^{(k)}$, a new blocking taxon and, for each element p_j in ϕ_k, a new taxon $v_{p_j}^{(k)}$, while the taxon set of the other caterpillar contains $v_F^{(k)}$, a new blocking taxon and, for each element q_j in ν_k, a new taxon $\neg v_{q_j}^{(k)}$. The precise ordering of the leaves in both caterpillars is shown in the dashed box of Fig. 4. It is easily checked that $|X(\mathcal{T}_v)| = 4n$ and, since each clause contains precisely three distinct variables, $|X(\mathcal{T}'_v)| = 4n + 3m$. Noting that the taxa sets of the trees in $X(\mathcal{T}_v)$ and $X(\mathcal{T}'_v)$ only overlap in $v_T^{(k)}$ and $v_F^{(k)}$, we have

$$|X(\mathcal{T}_v \cup \mathcal{T}'_v)| = 4n + 4n + 3m - 2n = 6n + 3m \qquad (1)$$

distinct taxa over all trees in $\mathcal{T}_v$ and $\mathcal{T}'_v$.

Clause gadget. We construct two sets $\mathcal{T}_c$ and $\mathcal{T}'_c$ of trees. For each $i \in \{1, 2, \ldots, m\}$, consider the clause $C_i = \ell^{(p)} \vee \ell^{(q)} \vee \ell^{(r)}$, where each $k \in \{p, q, r\}$ is an element in $\{1, 2, \ldots, n\}$ with $\ell^{(k)} \in \{v^{(k)}, \neg v^{(k)}\}$. Relative to C_i, we add three three-taxon trees to $\mathcal{T}_c$ which are shown in the solid box of Fig. 5. The first such tree has taxon set $\{\ell_i^{(p)}, c_1^i, \cup_a^i\}$ where $\ell_i^{(p)}$ is an element in $\{v_i^{(p)}, \neg v_i^{(p)}\}$. Note that $\ell_i^{(p)}$ labels a leaf of a tree in $\mathcal{T}'_v$ while the other two taxa do not label a leaf of a tree in $\mathcal{T}_v$ or $\mathcal{T}'_v$. The other two trees in $\mathcal{T}_c$ are constructed in an analogous way. Furthermore, for each C_i, we add two five-taxon trees to $\mathcal{T}'_c$ which

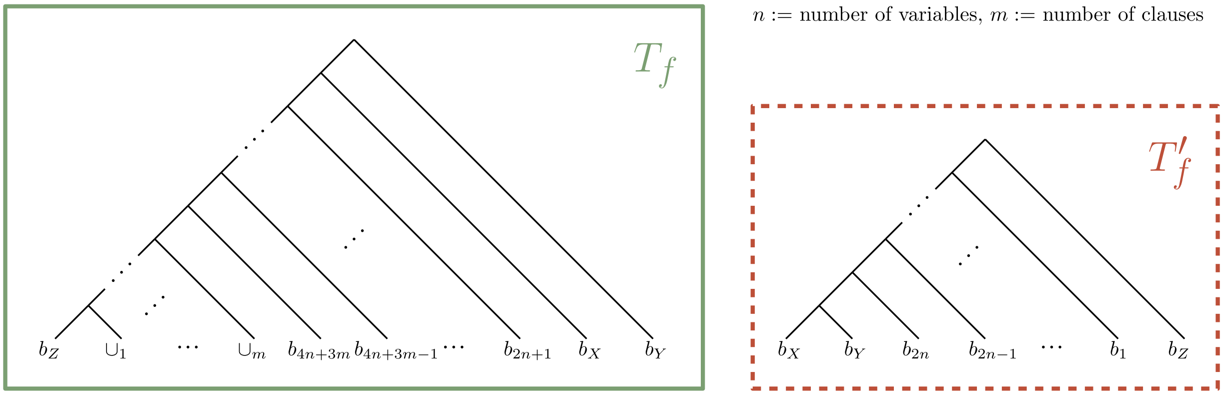

Fig. 6. The two trees T_f and T'_f in the construction of $\mathcal{T}$ from I.

are shown in the dashed box of Fig. 5. The taxa set of the first tree contains two new blocking taxa and the three previously encountered elements $\{c^i_1, c^i_2, c^i_3\}$, while the second tree contains one new blocking taxon, the new taxon $\cup_i$, and the three previously encountered elements $\{\cup^i_a, \cup^i_b, \cup^i_c\}$. Similar to the variable gadgets, we now count the number of taxa in trees in $\mathcal{T}_c$ and $\mathcal{T}'_c$. As no two trees in $\mathcal{T}_c$ or $\mathcal{T}'_c$ share a taxon, we have $|X(\mathcal{T}_c)| = 9m$ and $|X(\mathcal{T}'_c)| = 10m$. Moreover, since all taxa of trees in $\mathcal{T}'_c$, except for the blocking taxa and elements in $\{\cup_1, \cup_2, \dots, \cup_m\}$, are also taxa of trees in $\mathcal{T}_c$, we have

$$|X(\mathcal{T}_c \cup \mathcal{T}'_c)| = 9m + 10m - 6m = 13m. \tag{2}$$

Formula gadget. We complete the construction of $\mathcal{T}$ by constructing two caterpillars T_f and T'_f which are shown in the solid and dashed box of Fig. 6, and define

$$\mathcal{T} = \mathcal{T}_v \cup \mathcal{T}'_v \cup \mathcal{T}_c \cup \mathcal{T}'_c \cup \{T_f, T'_f\}.$$

Summarizing the construction, we have $|\mathcal{T}| = 3n + 5m + 2$. Moreover, by construction and Eqs. (1)–(2), it follows that $|X((\mathcal{T}_v \cup \mathcal{T}'_v) \cap (\mathcal{T}_c \cup \mathcal{T}'_c))| = 3m$. Now, since the three taxa b_X, b_Y, and b_Z, which are common to T_f and T'_f, are the only taxa of these two trees that are not contained in the taxa set of any other constructed tree, we have

$$|X(\mathcal{T})| = 6n + 3m + 13m - 3m + 3 = 6n + 13m + 3. \tag{3}$$

We next prove the following claim:

Claim 1. *I is satisfiable if and only if $\mathcal{T}$ has a cherry-picking sequence.*

First, suppose that I is satisfiable. Let $\beta : V \to \{T, F\}$ be a truth assignment for V such that each clause is satisfied. We next describe a sequence of pruning operation. Noting that each taxon in $X(\mathcal{T})$ is contained in the taxa sets of exactly two trees in $\mathcal{T}$ (a fact that we freely use throughout the rest of this proof), it is straightforward to verify that this sequence implies a cherry-picking sequence for $\mathcal{T}$.

Part 1: variable gadgets. For each variable $v^{(k)}$ with $k \in \{1, 2, \dots, n\}$ do the following. If $\beta(v^{(k)}) = T$ prune taxon $v^{(k)}_T$ from the two trees in $\mathcal{T}_v \cup \mathcal{T}'_v$ whose taxa sets contain $v^{(k)}_T$. On the other hand, if $\beta(v^{(k)}) = F$ prune taxon $v^{(k)}_F$ from the two trees in $\mathcal{T}_v \cup \mathcal{T}'_v$ whose taxa sets contain $v^{(k)}_F$. Taken together, these pruning steps delete a single leaf of each tree in $\mathcal{T}_v$ and a single leaf of half of the trees in $\mathcal{T}'_v$.

Part 2: clause gadgets. Consider the set of trees resulting from the pruning described in Part 1. For each $C_i = \ell^{(p)} \vee \ell^{(q)} \vee \ell^{(r)}$ with $i \in \{1, 2, \dots, m\}$, let L_i be a subset of $\{p, q, r\}$ such that $|L_i| = 2$ and, if $\ell^{(k)}_i$ is not satisfied by β, then $k \in L_i$. Setting $i = 1$, process the three literals in C_i from left to right in the following way.

(1) If $\ell^{(k)}_i$ is satisfied by β, prune $\ell^{(k)}_i$ from the tree in $\mathcal{T}_c$ whose taxa set contains $\ell^{(k)}_i$ and, noting that $\ell^{(k)}_i \in \{v^{(k)}_i, \neg v^{(k)}_i\}$, prune $\ell^{(k)}_i$ from the tree in $\mathcal{T}'_v$ whose taxa set contains $\ell^{(k)}_i$.

(2) If $k \in L_i$, prune c^i_s, where $s = 1$ if $k = p$, $s = 2$ if $k = q$, and $s = 3$ if $k = r$, from the two trees in $\mathcal{T}_c \cup \mathcal{T}'_c$ whose taxa sets contain c^i_s.

(3) Prune $\cup^i_t$, where $t = a$ if $k = p$, $t = b$ if $k = q$, and $t = c$ if $k = r$, from the two trees in $\mathcal{T}_c \cup \mathcal{T}'_c$ whose taxa sets contain $\cup^i_t$.

Now prune $\cup_i$ from the tree in $\mathcal{T}'_c$ whose taxa set contains $\cup_i$, and prune $\cup_i$ from T_f. If $i < m$, increment i by one and repeat this process with the next clause. Intuitively, by definition of L_i, the above process prunes exactly two elements in $\{c^i_1, c^i_2, c^i_3\}$. Since each clause is satisfied by β, this guarantees that we can prune each element in $\{\cup^i_a, \cup^i_b, \cup^i_c\}$ and, subsequently $\cup_i$.

Part 3: formula gadget and remaining taxa. Consider the set of trees resulting from the pruning described in Part 2. We prune the remaining taxa as follows.

(1) In order, prune each of

$$b_{4n+3m},\ b_{4n+3m-1},\ b_{4n+3m-2},\ \dots,\ b_{4n+3i},\ b_{4n+3i-1},\ b_{4n+3i-2},\ \dots,\ b_{4n+3},\ b_{4n+2},\ b_{4n+1}$$

from a tree $\mathcal{T}_c'$ whose taxa set contains the respective blocking taxa and from T_f. After all taxa have been pruned, each tree in $\mathcal{T}_c'$ is either the null tree or consists of a single vertex labeled c_s^i for some $s \in \{1, 2, 3\}$.

(2) For each $i \in \{1, 2, \dots, m\}$, prune the unique taxon c_s^i with $s \in \{1, 2, 3\}$ that has not been pruned in Part 2 from two trees in $\mathcal{T}_c \cup \mathcal{T}_c'$. Now, each tree in $\mathcal{T}_c \cup \mathcal{T}_c'$ that is not the null tree consists of a single vertex labeled $\ell_i^{(k)}$ for some $i \in \{1, 2, \dots, m\}$ and $k \in \{1, 2, \dots, n\}$.

(3) For each $k \in \{1, 2, \dots, n\}$, note that one of $\{b_{2(n+k)-1}, b_{2(n+k)}\}$ labels a leaf of a cherry in a tree in $\mathcal{T}_v'$ while the other labels the leaf of a tree in $\mathcal{T}_v'$ that consists of a single vertex. In order, prune each of

$$b_{4n},\ b_{4n-1},\ \dots,\ b_{2(n+k)},\ b_{2(n+k)-1},\ \dots,\ b_{2n+2},\ b_{2n+1}$$

from the tree in $\mathcal{T}_v'$ whose taxa set contains the respective blocking taxa and from T_f.

(4) In order, prune b_X and b_Y from T_f and T_f'.

(5) Consider the remaining trees in $\mathcal{T}_v$ and observe that each such tree consists of exactly three leaves, two of which are blocking taxa that form a cherry. In order, prune each of

$$b_{2n},\ b_{2n-1},\ \dots,\ b_{2k},\ b_{2k-1},\ \dots,\ b_2,\ b_1$$

from T_f' and the tree in $\mathcal{T}_v$ whose taxa set contains the respective blocking taxon.

(6) For each $k \in \{1, 2, \dots, n\}$, let $v_X^{(k)}$ be the unique element in $\{v_T^{(k)}, v_F^{(k)}\}$ that has not been pruned in Part 1. Prune $v_X^{(k)}$ from the two trees in $\mathcal{T}_v \cup \mathcal{T}_v'$ whose taxa sets contain $v_X^{(k)}$.

(7) For each $i \in \{1, 2, \dots, m\}$ in increasing order, consider each literal $\ell_i^{(k)}$ in $C_i = \ell^{(p)} \vee \ell^{(q)} \vee \ell^{(r)}$ with $k \in \{p, q, r\}$ that is not satisfied by β. By processing such literals from left to right in C_i, prune $\ell_i^{(k)}$ from the two trees in $\mathcal{T}_v' \cup \mathcal{T}_c$ whose taxa sets contain $\ell_i^{(k)}$. It is easily seen that the corresponding tree in $\mathcal{T}_v'$ either consists of a single vertex or contains a cherry with a leaf labeled $\ell_i^{(k)}$.

(8) Prune b_Z from T_f and T_f'.

Now, relative to the elements in $X(\mathcal{T})$, we prune $2n$ elements in Parts 1 and 3.6, all $4m$ elements in

$$\{\cup_1, \cup_a^i, \cup_b^i, \cup_c^i, \dots, \cup_m, \cup_a^m, \cup_b^m, \cup_c^m\}$$

in Part 2, and all $4n + 3m + 3$ blocking taxa in Parts 3.1, 3.3, 3.4, 3.5, and 3.8. Additionally, in Parts 2.1 and 3.7 we prune $3m$ taxa, and in Parts 2.2 and 3.2, we prune again $3m$ taxa. Summing up, we prune

$$6n + 13m + 3$$

taxa, which is equal to the number of elements in $X(\mathcal{T})$.

Second, suppose that $\mathcal{T}$ has a cherry-picking sequence $\sigma = (x_1, x_2, \dots, x_{|\sigma|})$. We write $x_i \prec x_j$ if and only if $i < j$ and $x_i \succ x_j$ if and only if $i > j$. Further, let

$$M := \{1, 2, \dots, m\},\ N := \{1, 2, \dots, n\},\ B := \{b_1, b_2, \dots, b_{4n+3m}, b_X, b_Y, b_Z\}.$$

We define a truth assignment $\beta\colon V \to \{T, F\}$ as follows

$$\beta\left(v^{(k)}\right) = \begin{cases} T & \text{if } \exists i \in M\colon v_T^{(k)} \prec \cup_i, \\ F & \text{else.} \end{cases}$$

In order to show that β satisfies each clause of I, we establish four necessary conditions that σ fulfills by construction.

(1) All taxa in $\{\cup_1, \cup_2, \dots, \cup_m\}$ are pruned earlier than any blocking taxon:

$$\forall i \in M\ \forall b \in B\colon \cup_i \prec b. \tag{4}$$

Argument: Observe that the arrangement of b_X, b_Y, b_Z in T_f and T_f' implies that all taxa in $\{\cup_1, \cup_2, \dots, \cup_m\}$ are pruned prior to any blocking taxon. Furthermore, we cannot prune any taxon in T_f' until we have pruned all taxa from T_f except for b_X, b_Y, and b_Z. We will freely use Condition 1 throughout the remainder of this proof.

(2) Let $C_i = \ell^{(p)} \vee \ell^{(q)} \vee \ell^{(r)}$ be a clause of I. At least one taxon in $\{\ell_i^{(p)}, \ell_i^{(q)}, \ell_i^{(r)}\}$ is pruned earlier than $\cup_i$. Stated more formally:

$$\forall i \in M\ \exists \ell_i^{(s_i)} \in \left\{\ell_i^{(p_i)}, \ell_i^{(q_i)}, \ell_i^{(r_i)}\right\}\colon \ell_i^{(s_i)} \prec \cup_i. \tag{5}$$

Argument: Consider the five trees in $\mathcal{T}_c \cup \mathcal{T}'_c$ representing C_i (see Fig. 5). In order to prune $\cup_i$, we have to prune all taxa in $\{\cup^i_a, \cup^i_b, \cup^i_c\}$ first. Since we can prune at most two taxa in $\{c^i_1, c^i_2, c^i_3\}$ prior to an element in $\{b_{4n+3i-2}, b_{4n+3i-1}\}$, pruning all taxa in $\{\cup^i_a, \cup^i_b, \cup^i_c\}$ is only possible if at least one taxon in $\{\ell^{(p)}_i, \ell^{(q)}_i, \ell^{(r)}_i\}$ has been pruned previously.

(3) Let $v^{(k)} \in V$ be any variable of I. Recall the definition of the tuples ϕ_k and ν_k that is used in the construction of the variable gadget. If there exists a $v^{(k)}_i$ with $v^{(k)}_i \prec \cup_i$ for some $i \in \phi_k$, then $v^{(k)}_T$ is also pruned earlier than $\cup_i$. Stated formally:

$$\forall k \in N \;\forall i \in \phi_k\colon \left(v^{(k)}_i \prec \cup_i \implies v^{(k)}_T \prec \cup_i\right). \tag{6}$$

Argument: Consider a variable $v^{(k)} \in V$ such that $v^{(k)}_i \prec \cup_i$ for some $i \in \phi_k$. Since there is no blocking taxon $b \in B$ with $b \prec \cup_i$, we have $\cup_i \prec b_{2(n+k)-1}$. Thus, $v^{(k)}_T$ is pruned from the associated caterpillar in $\mathcal{T}'_v$ that contains $v^{(k)}_i$ such that $v^{(k)}_T \prec v^{(k)}_i \prec \cup_i$ (see Fig. 4).

The following can be shown analogously. If there exists a $\neg v^{(k)}_i \prec \cup_i$ for some $i \in \nu_k$, then $v^{(k)}_F$ is also pruned earlier than $\cup_i$. Stated formally:

$$\forall k \in N \;\forall i \in \nu_k\colon \left(\neg v^{(k)}_i \prec \cup_i \implies v^{(k)}_F \prec \cup_i\right). \tag{7}$$

(4) Let $v^{(k)} \in V$ be any variable of I. If $v^{(k)}_T$ is pruned earlier than some taxon in $\{\cup_1, \cup_2, \ldots, \cup_m\}$, then $v^{(k)}_F$ is pruned later than all taxa in $\{\cup_1, \cup_2, \ldots, \cup_m\}$, i.e.,

$$\forall k \in N\colon \left(\left(\exists i \in M\colon v^{(k)}_T \prec \cup_i\right) \implies \left(\forall i \in M\colon v^{(k)}_F \succ \cup_i\right)\right). \tag{8}$$

Argument: Consider a variable $v^{(k)} \in V$ such that $v^{(k)}_T \prec \cup_i$ for some $i \in M$. Assume towards a contradiction that there is some $j \in M$ such that $v^{(k)}_F \prec \cup_j$. Then, one of the two blocking taxa b_{2k-1} and b_{2k} is pruned prior to $v^{(k)}_F$ (see Fig. 4). But this is not possible since there is no blocking taxon $b \in B$ with $b \prec \cup_j$.

As an immediate consequence of statement (8), we get the analogous statement for $v^{(k)}_F$, i.e.,

$$\forall k \in N\colon \left(\left(\exists i \in M\colon v^{(k)}_F \prec \cup_i\right) \implies \left(\forall i \in M\colon v^{(k)}_T \succ \cup_i\right)\right). \tag{9}$$

Now, we show that β indeed satisfies each clause of I. For each clause $C_i = \ell^{(p)} \vee \ell^{(q)} \vee \ell^{(r)}$, we have $\ell^{(s)}_i \prec \cup_i$ for some $\ell^{(s)}_i \in \left\{\ell^{(p)}_i, \ell^{(q)}_i, \ell^{(r)}_i\right\}$ (Condition 2). Since $\ell^{(s_i)}_i \in \left\{v^{(k)}_i, \neg v^{(k)}_i\right\}$ for some $k \in N$, we have $v^{(k)}_T \prec \cup_i$ if $\ell^{(s_i)}_i = v^{(k)}_i$ and $v^{(k)}_F \prec \cup_i$ if $\ell^{(s_i)}_i = \neg v^{(k)}_i$ (Condition 3). Hence, by setting $\beta(v^{(k)}) = T$ if $v^{(k)}_T \prec \cup_i$ and $\beta(v^{(k)}) = F$ if $v^{(k)}_F \prec \cup_i$, we satisfy at least one literal of each clause. Note that we can assign arbitrary truth values to variables $v^{(k)}$ with $v^{(k)}_T \succ \cup_i$ and $v^{(k)}_F \succ \cup_i$ for all $i \in M$. Here, we choose to set all these variables to F. The truth assignment β is consistent, since at least one taxon in $\left\{v^{(k)}_T, v^{(k)}_F\right\}$ is pruned later than all taxa in $\{\cup_1, \cup_2, \ldots, \cup_m\}$ (Condition 4). Hence, the truth assignment β is consistent and satisfies each clause of I.

Folding into two trees on the same set of taxa. The trees in $\mathcal{T}_v \cup \mathcal{T}_c \cup \{T_f\}$ and, similarly, the trees in $\mathcal{T}'_v \cup \mathcal{T}'_c \cup \{T'_f\}$ (see Fig. 4, 5, and 6) have mutually disjoint taxa sets. Hence, by $n + 3m$ applications of Lemma 2, we can construct a compound tree S for all trees in $\mathcal{T}_v \cup \mathcal{T}_c \cup \{T_f\}$ and, by $2n + 2m$ applications of Lemma 2, we can construct a compound tree S' for all trees in $\mathcal{T}'_v \cup \mathcal{T}'_c \cup \{T'_f\}$ such that $\mathcal{T}$ has a cherry-picking sequence if and only if S and S' have a cherry-picking sequence. Lastly, by applying Lemma 1, we obtain two trees T and T' from S and S', respectively, such that $X(T) = X(T')$, and S and S' have a cherry-picking sequence if and only if T and T' have such a sequence. It now follows that I is satisfiable if and only if T and T' have a cherry-picking sequence.

Number of taxa in the final instance. It remains to show that T and T' can be constructed in polynomial time. By Eq. (3), recall that $|X(\mathcal{T})| = 6n + 13m + 3$. Now, since we apply Lemma 2 a total of $3n + 5m$ times and each application introduces two new taxa, we have

$$|X(\{S, S'\})| = 6n + 13m + 3 + 2(3n + 5m) = 12n + 23m + 3.$$

Observe that each taxon in $X(\mathcal{T})$ labels a leaf of a unique tree in $\mathcal{T}_v \cup \mathcal{T}_c \cup \{T_f\}$ and a leaf of a unique tree in $\mathcal{T}'_v \cup \mathcal{T}'_c \cup \{T'_f\}$. It therefore follows that each taxon that is contained in exactly one of $X(S)$ and $X(S')$ has been introduced by an application of Lemma 2. Conversely, each application of this lemma introduces two taxa that are both contained in exactly one of $X(S)$ and $X(S')$. Hence, recalling that in obtaining T and T' from S and S', respectively, an additional leaf labeled ρ is introduced (see the third sentence in the proof of Lemma 1), we have

$$|X(T)| = |X(T')| = 12n + 23m + 3 + 2(3n + 5m) + 1 = 18n + 33m + 4.$$

It now follows, that the size of T and T' as well as the time it takes to construct these two trees are polynomial. This completes the proof of the theorem. □

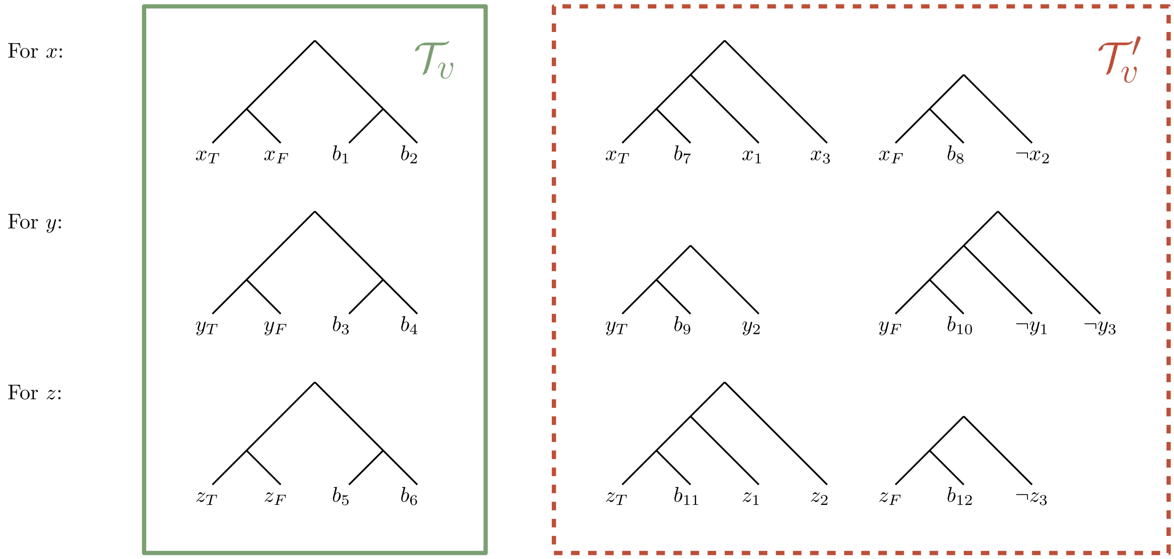

Fig. 7. The variable gadget for $(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z)$.

To illustrate the proof of Theorem 1, we now give an explicit example of a 3-SAT instance and show how it is reduced to a set of trees by following the construction that is described in the aforementioned proof. Let I be the following instance of 3-SAT

$$\underbrace{(x \vee \neg y \vee z)}_{C_1} \wedge \underbrace{(\neg x \vee y \vee z)}_{C_2} \wedge \underbrace{(x \vee \neg y \vee \neg z)}_{C_3}.$$

For the purpose of ordering the blocking taxa in the same way as described in the proof, we regard variable x as $v^{(1)}$, variable y as $v^{(2)}$, and variable z as $v^{(3)}$. Let $n = 3$ (resp. $m = 3$) be the number of variables (resp. clauses) in I. We construct a set $\mathcal{T}$ of $3n + 5m + 2 = 26$ trees. The 9 trees that represent the variable gadget $\mathcal{T}_v$ and $\mathcal{T}'_v$ are shown in Fig. 7, the 15 trees that represent the clause gadget $\mathcal{T}_c$ and $\mathcal{T}'_c$ are shown in Fig. 8 and the two trees that represent the formula gadget T_f and T'_f are shown in Fig. 9. Note that $|X(\mathcal{T})| = 3n + 13m + 3 = 60$. Clearly, I is satisfied for the truth assignment $\beta : \{x, y, z\} \to \{T, F\}$ with $\beta(x) = \beta(z) = T$, $\beta(y) = F$. To see that $\mathcal{T}$ also has a cherry-picking sequence of length 60, we follow the sequence of pruning operations that is described in Parts 1–3 in the first direction of the proof of Claim 1

$$\begin{aligned}&(x_T, y_F, z_T,\\ &\ x_1, c_1^1, \cup_a^1, \neg y_1, c_2^1, \cup_b^1, z_1, \cup_c^1, \cup_1, c_1^2, \cup_a^2, c_2^2, \cup_b^2, z_2, \cup_c^2, \cup_2, x_3, \cup_a^3, \neg y_3, c_2^3, \cup_b^3, c_3^3, \cup_c^3, \cup_3,\\ &\ b_{21}, b_{20}, \ldots, b_{13}, c_3^1, c_3^2, c_1^3, b_{12}, b_{11}, \ldots, b_7, b_X, b_Y, b_6, b_5, \ldots, b_1, x_F, y_T, z_F, \neg x_2, y_2, \neg z_3, b_Z),\end{aligned}$$

where line 1 corresponds to Part 1, line 2 corresponds to Part 2, and line 3 corresponds to Part 3.

4. Discussion

Given any set of input trees, there always exists some phylogenetic network displaying them. Roughly speaking, one can simply merge the input trees at the leaves and at the root. However, what happens when you restrict the network to have some additional, biologically motivated, properties? Then there might not always exist a network displaying the input trees. Moreover, deciding whether or not there exists such a network may be a difficult problem. Indeed, in this paper we have shown that even if the input consists of only two binary trees, it is already NP-complete to decide whether there exists any temporal phylogenetic network displaying them.

One could be tempted to look for approximation algorithms for the associated optimization problem: given a set of phylogenetic trees, find a temporal network that displays them and has smallest possible reticulation number, if such a network exists. Note, however, that an approximation algorithm is required to always output a valid solution, for any valid input. The problem formulation above (based on [13]) does not specify what a valid solution is when there does not exist a temporal network displaying the input trees. Nevertheless, whatever the output in that case is, it can be checked in polynomial time whether the output of the algorithm is a temporal network displaying the input trees. This is because temporal networks are tree-child, and checking whether a tree-child network displays a tree can be achieved in polynomial time [16]. Hence, any approximation algorithm for the problem could be used to decide in polynomial time whether there

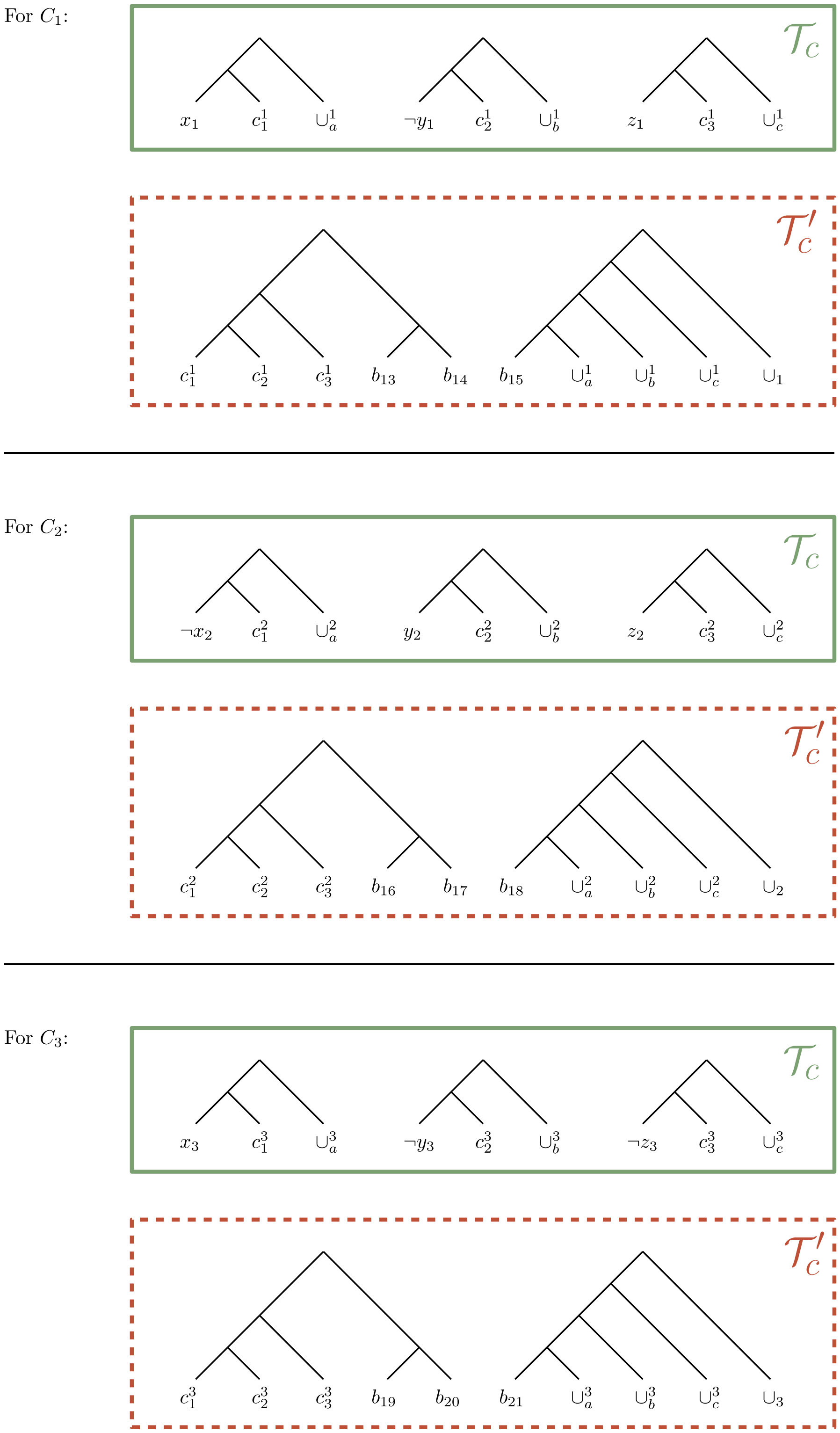
For C_1:
$\mathcal{T}_c$
x_1 c_1^1 $\cup_a^1$ $\neg y_1$ c_2^1 $\cup_b^1$ z_1 c_3^1 $\cup_c^1$
$\mathcal{T}_c'$
c_1^1 c_2^1 c_3^1 b_{13} b_{14} b_{15} $\cup_a^1$ $\cup_b^1$ $\cup_c^1$ $\cup_1$
For C_2:
$\mathcal{T}_c$
$\neg x_2$ c_1^2 $\cup_a^2$ y_2 c_2^2 $\cup_b^2$ z_2 c_3^2 $\cup_c^2$
$\mathcal{T}_c'$
c_1^2 c_2^2 c_3^2 b_{16} b_{17} b_{18} $\cup_a^2$ $\cup_b^2$ $\cup_c^2$ $\cup_2$
For C_3:
$\mathcal{T}_c$
x_3 c_1^3 $\cup_a^3$ $\neg y_3$ c_2^3 $\cup_b^3$ $\neg z_3$ c_3^3 $\cup_c^3$
$\mathcal{T}_c'$
c_1^3 c_2^3 c_3^3 b_{19} b_{20} b_{21} $\cup_a^3$ $\cup_b^3$ $\cup_c^3$ $\cup_3$

Fig. 8. The clause gadget for $(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z)$.

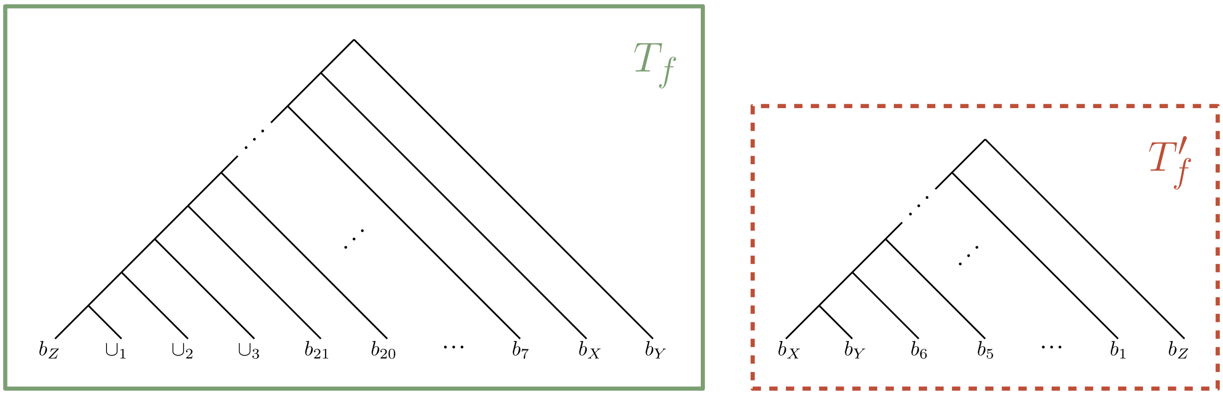

Fig. 9. The formula gadget for $(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z)$.

exists a temporal network displaying the input trees, which is not possible, unless P = NP, given the NP-completeness shown in this paper.

Therefore, a more promising direction is to consider fixed-parameter algorithms for the associated parameterized version of the problem. Given a set of phylogenetic trees and a parameter k, decide whether there exists a temporal network that displays the input trees and has reticulation number at most k. One then aims at algorithms solving this problem in $O(|X|^{O(1)}f(k))$ time, with f some function of k, preferably of the form c^k with c a small constant. Intuitively, such an FPT algorithm is only exponential in the reticulation number and not in the number of leaves. Indeed, even though it is NP-complete to decide whether there exists a temporal network with unlimited reticulation number, for small reticulation numbers this problem might be much easier. In fact, for instances of two binary trees a fixed-parameter algorithm is already known [12]. Important open problems include the question whether such algorithms exist for instances of more than two trees and whether algorithms can be developed that work well in practice.

It would also be interesting to consider other biologically motivated network classes. For example, binary tree-child (e.g. [5]) or tree-sibling networks (e.g. [3]). Could it be that one of the associated decision problems is nontrivial (for more than two input trees) but polynomial-time solvable? For other network classes, such as tree-based (e.g. [9]) or time-consistent (e.g. [4]) networks, it is known that there always exists a solution [25]. For such classes, it would be interesting to study the optimization version of the problem.

Acknowledgments

We thank Matúš Mihalák for useful discussions. Leo van Iersel was partly supported by the Netherlands Organization for Scientific Research (NWO), including Vidi grant 639.072.602, and partly by the 4TU Applied Mathematics Institute. Simone Linz was supported by the New Zealand Marsden Fund.

References

[1] E. Bapteste, L. van Iersel, A. Janke, S. Kelchner, S. Kelk, J. McInerney, D. Morrison, L. Nakhleh, M. Steel, L. Stougie, J. Whitfield, Networks: expanding evolutionary thinking, Trends Genet. 29 (8) (2013) 439–441.
[2] M. Bordewich, C. Semple, Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3) (2007) 458–466.
[3] G. Cardona, M. Llabrés, F. Rosselló, G. Valiente, A distance metric for a class of tree-sibling phylogenetic networks, Bioinformatics 24 (13) (2008) 1481–1488.
[4] G. Cardona, M. Llabrés, F. Rosselló, G. Valiente, Path lengths in tree-child time consistent hybridization networks, Inform. Sci. 180 (3) (2010) 366–383.
[5] G. Cardona, F. Rosselló, G. Valiente, Comparison of tree-child phylogenetic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (4) (2009) 552–569.
[6] Z.-Z. Chen, L. Wang, Hybridnet: a tool for constructing hybridization networks, Bioinformatics 26 (22) (2010) 2912–2913.
[7] M. Cygan, F. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, vol. 3., Springer, 2015.
[8] J. Döcker, S. Linz, On the existence of a cherry-picking sequence, Theoret. Comput. Sci. 714 (2018) 36–50.
[9] A. Francis, M. Steel, Which phylogenetic networks are merely trees with additional arcs? Syst. Biol. 64 (5) (2015) 768–777.
[10] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979.
[11] D. Gusfield, ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks, The MIT Press, 2014.
[12] P. Humphries, S. Linz, C. Semple, Cherry picking: a characterization of the temporal hybridization number for a set of phylogenies, Bull. Math. Biol. 75 (10) (2013) 1879–1890.
[13] P. Humphries, S. Linz, C. Semple, On the complexity of computing the temporal hybridization number for two phylogenies, Discrete Appl. Math. 161 (7) (2013) 871–880.
[14] D. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks: Concepts, Algorithms and Applications, Cambridge University Press, 2011.
[15] L. van Iersel, S. Kelk, C. Scornavacca, Kernelizations for the hybridization number problem on multiple nonbinary trees, J. Comput. System Sci. 82 (6) (2016) 1075–1089.
[16] L. van Iersel, C. Semple, M. Steel, Locating a tree in a phylogenetic network, Inform. Process. Lett. 110 (23) (2010) 1037–1043.

[17] R. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations (Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), Plenum, 1972, pp. 85–103.

[18] S. Kelk, L. van Iersel, S. Linz, N. Lekić, C. Scornavacca, Stougie. Cycle killer. L., Qu'est-ce que c'est? On the comparative approximability of hybridization number and directed feedback vertex set, SIAM J. Discrete Math. 26 (4) (2012) 1635–1656.

[19] B. Moret, L. Nakhleh, T. Warnow, C. Linder, A. Tholse, A. Padolina, J. Sun, R. Timme, Phylogenetic networks: modeling, reconstructibility, and accuracy, IEEE/ACM Trans. Comput. Biol. Bioinform. 1 (1) (2004) 13–23.

[20] C. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.

[21] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.

[22] S. Soucy, J. Huang, J. Gogarten, Horizontal gene transfer: building the web of life, Nature Rev. Genet. 16 (8) (2015) 472–482.

[23] M. Steel, Phylogeny: Discrete and Random Processes in Evolution, SIAM, 2016.

[24] C. Whidden, R. Beiko, N. Zeh, Fixed-parameter algorithms for maximum agreement forests, SIAM J. Comput. 42 (4) (2013) 1431–1466.

[25] L. Zhang, On tree-based phylogenetic networks, J. Comput. Biol. 23 (7) (2016) 553–565.