Bulletin of Mathematical Biology manuscript No.
(will be inserted by the editor)

Cherry picking: a characterization of the temporal hybridization number for a set of phylogenies

Peter J. Humphries · Simone Linz · Charles Semple

Received: date / Accepted: date

Abstract Recently, we have shown that calculating the minimum-temporal-hybridization number for a set $\mathcal{P}$ of rooted binary phylogenetic trees is NP-hard and have characterized this minimum number when $\mathcal{P}$ consists of exactly two trees. In this paper, we give the first characterization of the problem for $\mathcal{P}$ being arbitrarily

S.L. was supported by a Marie Curie International Outgoing Fellowship within the 7$^{\text{th}}$ European Community Framework Programme. C.S. was supported by the New Zealand Marsden Fund and the Allan Wilson Centre for Molecular Ecology and Evolution.

Peter J. Humphries
Department of Mathematics and Physics, North Carolina Central University, Durham, NC, USA
E-mail: pjhumphries@gmail.com

Simone Linz
Center for Bioinformatics, University of Tübingen, Tübingen, Germany
E-mail: linz@informatik.uni-tuebingen.de
Present address: Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

Charles Semple
Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand
E-mail: charles.semple@canterbury.ac.nz

large. The characterization is in terms of cherries and the existence of a particular type of sequence. Furthermore, in an online appendix to the paper, we show that this new characterization can be used to show that computing the minimum-temporal hybridization number for two trees is fixed-parameter tractable.

Keywords cherry · fixed-parameter tractability · phylogenetic network · phylogenetic tree · temporal network

1 Introduction

While the construction of phylogenetic networks from phylogenetic trees has recently attracted much attention from a mathematical and computational viewpoint (e.g. [5, 9, 14, 15, 18]), the biological meaningfulness of the resulting networks is often questioned. Indeed, the reconstruction of rooted phylogenetic networks is still not widely used by biologists and the development of tools that are biologically significant and computationally tractable remains challenging [13]. In this paper, we investigate rooted phylogenetic networks—called *temporal* networks—that satisfy the following two time constraints. First, speciation events occur successively and, second, reticulation events, such as horizontal gene transfer, hybridization or recombination, occur instantaneously. Hence, the three species that are involved in a reticulation event, i.e. the new species resulting from this event and its two distinct parents, must coexist in time. While these constraints are biologically well-motivated, they are often neglected in theoretical work. Yet, if two vertices (resp. species) in a phylogenetic network $\mathcal{N}$ do not coexist in time, then their genomes can only be combined to create a new species when explained by an additional evolutionary event that is not present in $\mathcal{N}$. In the following, we consider hybridization as representative of reticulation and, therefore, refer to a reticulation event as a hybridization event.

Emphasizing that a set of rooted phylogenetic trees can always simultaneously be embedded into a rooted phylogenetic network, but not necessarily into a temporal network, we state the following decision problem. Suppose that we are given a positive integer k and a set $\mathcal{P}$ of rooted binary phylogenetic trees that correctly represent the evolutionary history of a set of present-day species for $|\mathcal{P}|$ distinct genetic markers, where $|\mathcal{P}|$ is the number of elements (i.e. trees) in $\mathcal{P}$. Can the elements of $\mathcal{P}$ be merged into a temporal network with fewer than k hybridization events? We call this problem MINIMUM-TEMPORAL HYBRIDIZATION (resp. 2-MINIMUM-TEMPORAL HYBRIDIZATION for the restricted case when $|\mathcal{P}| = 2$). Furthermore, if k is the smallest value for which MINIMUM-TEMPORAL HYBRIDIZATION returns the answer 'yes', then $k - 1$ is said to be the *minimum-temporal-hybridization number* for $\mathcal{P}$. This minimum number indicates the significance with which hybridization has influenced the evolutionary history of the species under consideration. In an earlier paper [12], we showed that the corresponding optimization version of 2-MINIMUM-TEMPORAL HYBRIDIZATION is APX-hard and thus also NP-hard. Unless P=NP, it is therefore unlikely that an efficient and exact algorithm exists to solve MINIMUM-TEMPORAL HYBRIDIZATION.

In this paper, we give a novel characterization of the minimum-temporal-hybridization number for an arbitrary sized set $\mathcal{P}$ of rooted binary phylogenetic trees. This characterization is based on a particular type of sequence of the leaves of the trees in $\mathcal{P}$ and is unrelated to the popular concept of agreement forests that is frequently employed to quantify hybridization (see below). We call such a sequence a *cherry-picking sequence* for $\mathcal{P}$. Without time constraints, MINIMUM-TEMPORAL HYBRIDIZATION becomes the well-known problem HYBRIDIZATION NUMBER [1,8, 9,17,18,20] for which an analogous characterization remains elusive despite the attempts of a number of independent collaborations within the last ten years. The reason for wanting to establish such a characterization is that, for the instance of HYBRIDIZATION NUMBER when $|\mathcal{P}| = 2$, there exists a characterization in terms of agreement forests and it is precisely this characterization that yields the exact algorithms [5,8,9,18,20] for this particular instance. However, this characterization does not appear to extend to when $|\mathcal{P}| > 2$. An analogous situation occurs for Minimum-Temporal Hybridization. In [12], we established an agreement forest characterization for the minimum-temporal hybridization number for two rooted binary phylogenetic trees, but, again, this characterization does not appear to generalize to an arbitrary sized set of rooted binary phylogenetic trees. The only available algorithms for Hybridization Number for when $|\mathcal{P}| > 2$ are described in [7, 19] but these are simply heuristics that compute lower and upper bounds on the exact solution.

Fixed-parameter algorithms have proven to be a useful tool for solving many NP-hard problems exactly. Roughly speaking, a decision problem is *fixed-parameter tractable* if there exists an algorithm that solves it in time $O(f(k)p(n))$, where f is some function, k is a parameter of the input, p is some fixed polynomial, and n is the size of the input. Thus, if k is reasonably small, the problem may be tractable even for large n. For fixed-parameter algorithms in the context of phylogenetics, we refer the interested reader, for example, to [9–11]. In the online appendix to this paper, we show that 2-Minimum-Temporal Hybridization is fixed-parameter tractable. We present a fixed-parameter algorithm—called 2-MinTempHybrid—that exactly computes the minimum-temporal-hybridization number for two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ or determines that they cannot be merged into a temporal network. In particular, by making use of the aforementioned characterization of the minimum-temporal-hybridization number in terms of a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ and using two previously established reduction rules [5], we reduce the size (i.e. the number of leaves) of $\mathcal{T}$ and $\mathcal{T}'$ so that it is linear in the minimum-temporal-hybridization number for $\mathcal{T}$ and $\mathcal{T}'$.

The remainder of the paper is organized as follows. The next section contains notation and terminology that is used throughout the rest of the paper. In Section 3, we establish the above-mentioned characterization of the minimum-temporal-hybridization number in terms of cherry-picking sequences for an ar-

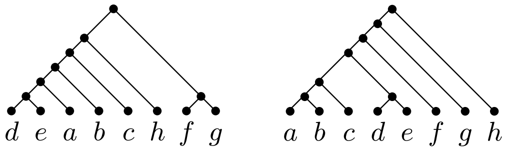

Fig. 1 Two trees $\mathcal{T}$ and $\mathcal{T}'$ on $X = \{a, b, \ldots, h\}$.

bitrarily large set of rooted binary phylogenetic trees. Section 4 briefly outlines an approach to establish fixed-parameter tractability for 2-Minimum-Temporal Hybridization in terms of cherry-picking sequences. Full details (including formal proofs) of this result are given in the online appendix to this paper. Lastly, Section 5 finishes off with a brief conclusion.

2 Preliminaries

This section provides notation and terminology that is used throughout the rest of the paper.

2.1 Phylogenetic trees

Throughout this paper, we use $[k]$ to refer to the set $\{1, 2, \ldots, k\}$ and X to denote a finite set with $|X| = n$. A *rooted binary phylogenetic X-tree* $\mathcal{T}$ is a rooted tree with leaf set X and, apart from the root which has degree two, all interior vertices have degree three. To ease reading, we refer to a rooted binary phylogenetic X-tree as a *tree on X*. Furthermore, in the remainder of this section and Section 3, we use $\mathcal{P}$ to denote a set of trees on X, where $|\mathcal{P}| = m$. The set X is often said to be the *label set* of $\mathcal{T}$ and is denoted by $\mathcal{L}(\mathcal{T})$. Furthermore, a pair of leaves $\{a, b\}$ of $\mathcal{T}$ is called a *cherry* if a and b are leaves that are adjacent to a common vertex of $\mathcal{T}$. Figure 1 shows two trees $\mathcal{T}$ and $\mathcal{T}'$ on $\mathcal{L}(\mathcal{T}) = \mathcal{L}(\mathcal{T}') = \{a, b, \ldots, h\}$, where $\mathcal{T}$ has the two cherries $\{d, e\}$ and $\{f, g\}$

Now, let $\mathcal{T}$ be a tree on X, and let $X' = \{x_1, x_2, \ldots, x_k\}$ be a subset of X. The minimal rooted subtree of $\mathcal{T}$ that connects all vertices in X' is denoted by $\mathcal{T}(X')$. Furthermore, the rooted tree obtained from $\mathcal{T}(X')$ by contracting all non-root degree-2 vertices is the *restriction of $\mathcal{T}$ to X'* and is denoted by $\mathcal{T}|X'$. We also write $\mathcal{T}[-x_1, x_2, \ldots, x_k]$ or $\mathcal{T}[-X']$ for short to denote $\mathcal{T}|(X - X')$. For a set $\mathcal{P} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$ of trees on X, we sometimes write $\mathcal{P}|X'$ (resp. $\mathcal{P}[-X']$) when referring to the set $\{\mathcal{T}_1|X', \mathcal{T}_2|X', \ldots, \mathcal{T}_m|X'\}$ (resp. $\{\mathcal{T}_1[-X'], \mathcal{T}_2[-X'], \ldots, \mathcal{T}_m[-X']\}$). Lastly, a rooted phylogenetic tree is *pendant* in $\mathcal{P}$ if it can be detached from each tree in $\mathcal{P}$ by deleting a single edge.

2.2 Temporal networks

A *network* $\mathcal{N}$ on a finite set X is a rooted acyclic digraph with the following properties:

(i) the *root* has in-degree 0 and out-degree at least 2;

(ii) X is the set of *leaves* of the network, that is, the vertices with out-degree 0 and in-degree 1;

(iii) all remaining vertices are *interior vertices*, and each such vertex either has in-degree 1 and out-degree at least 2 or is a *hybridization vertex* that has in-degree at least 2 and out-degree 1;

(iv) arcs ending in a hybridization vertex are *hybridization arcs*, while all other arcs in the network are *tree arcs*; and

(v) every interior vertex has at least one outgoing tree arc.

We note that the above definition of a network coincides with that of a so-called tree-child network which was introduced by Cardona et al. [6].

Now, let $\mathcal{N}$ be a network on X, and let $\mathcal{P}$ be a set of trees on X' with $X' \subseteq X$. We say that $\mathcal{N}$ *displays* $\mathcal{P}$ if each tree in $\mathcal{P}$ can be obtained from $\mathcal{N}$ by a sequence of arc and vertex deletions, and degree-2 vertex contractions.

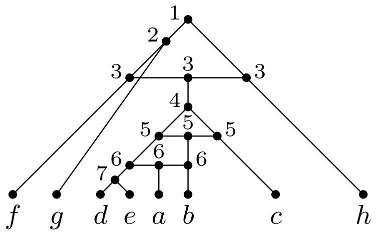

Fig. 2 A temporal network $\mathcal{N}$. For each internal vertex v a temporal labeling $t(v)$ is given next to v. Note that $\mathcal{N}$ is a minimum-temporal network for the two trees shown in Figure 1.

Again, let $\mathcal{N}$ be a network on X, and let V be the set of vertices of $\mathcal{N}$. Let $t : V \to \mathbb{R}^+$ be a map such that, for all $u, v \in V$, we have $t(u) = t(v)$ whenever (u, v) is a hybridization arc, and $t(u) < t(v)$ whenever (u, v) is a tree arc. Then t is a *temporal labeling* of $\mathcal{N}$, in which case $\mathcal{N}$ is said to be *temporal.* An example of a temporal network is shown in Figure 2, where arcs directed horizontally are hybridization arcs while arcs directed downwards are tree arcs.

For a temporal network $\mathcal{N}$, the *temporal-hybridization number* of $\mathcal{N}$ is defined as

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

where $d^-(v)$ denotes the in-degree of v. Furthermore, for a set $\mathcal{P}$ of trees on X, we define the *minimum-temporal-hybridization number* as follows:

$$h_t(\mathcal{P}) = \min\{h_t(\mathcal{N}) : \mathcal{N} \text{ is a temporal network on } X \text{ that displays } \mathcal{P}\}.$$

A temporal network $\mathcal{N}$ on X that displays a set $\mathcal{P}$ of trees on X and has the property $h_t(\mathcal{N}) = h_t(\mathcal{P})$ is said to be a *minimum-temporal network* for $\mathcal{P}$. Note that there are instances of $\mathcal{P}$ for which no such temporal network displays $\mathcal{P}$ [12]. Furthermore, we will sometimes write $h_t(\mathcal{T}, \mathcal{T}')$ instead of $h_t(\mathcal{P})$, in which case $\mathcal{P}$ contains exactly two trees $\mathcal{T}$ and $\mathcal{T}'$ on X. In such a case, the in-degree of any vertex of a minimum-temporal network $\mathcal{N}$ for $\mathcal{P}$ is at most 2 and, so, $h_t(\mathcal{N})$ is simply the number of hybridization vertices of $\mathcal{N}$. In contrast to Lemma 2 of [14], we emphasize that, due to property (v) in the definition of a network, a temporal-hybridization network $\mathcal{N}$ that has a hybridization vertex whose in-degree is greater than 2 cannot be transformed into a temporal-hybridization network $\mathcal{N}'$ such that each hybridization vertex of $\mathcal{N}'$ has in-degree 2 and $h_t(\mathcal{N}) = h_t(\mathcal{N}')$.

We next state the decision problem Minimum-Temporal Hybridization that was informally described in the introduction of this paper.

Minimum-Temporal Hybridization

Instance. A set $\mathcal{P}$ of trees on X and a positive integer k.

Question. Does there exist a temporal network on X that displays $\mathcal{P}$, and has fewer than k hybridization vertices?

In the remainder of this paper, we sometimes consider Minimum-Temporal Hybridization for when $m = 2$ in which case we refer to it as 2-Minimum-Temporal Hybridization. It was shown in [12] that 2-Minimum-Temporal Hybridization is NP-hard and, thus, Minimum-Temporal Hybridization remains computationally hard for an arbitrary sized set $\mathcal{P}$.

Remark. As mentioned in the introduction, by weakening the question of Minimum-Temporal Hybridization so that it asks whether or not there exists a network on X that displays $\mathcal{P}$ and has fewer than k hybridization vertices, we derive a second decision problem—called Hybridization Number—that was introduced by Bordewich and Semple [5] and since then has been investigated in many subsequent studies for when $|\mathcal{P}| = 2$. Analogously to the definition of $h_t(\mathcal{T}, \mathcal{T}')$, we define the *minimum-hybridization number* for two trees $\mathcal{T}$ and $\mathcal{T}'$ on X, denoted by $h(\mathcal{T}, \mathcal{T}')$, as the minimum number of hybridization vertices of a so-called hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$. In comparison with networks as defined at the beginning of this subsection, a hybridization network is more general and, in particular, may or may not be temporal. However, by slightly modifying the proof of [2, Theorem 2], one can show that, if there exists a hybridization network $\mathcal{N}$ that displays $\mathcal{T}$ and $\mathcal{T}'$, and $h(\mathcal{N}) = k$, then there also exists a tree-child network $\mathcal{N}'$ that displays $\mathcal{T}$ and $\mathcal{T}'$, and $h(\mathcal{N}') \leq k$.

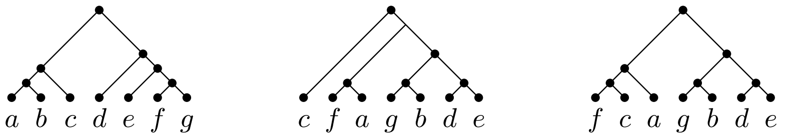

Fig. 3 Three trees on X for which $\sigma = (f, g, d, b, a, e, c)$ is a cherry-picking sequence with weight $w(\sigma) = 5$.

For the reader familiar with the notion of agreement forests, we end this section by loosely describing a characterization for when two trees $\mathcal{T}$ and $\mathcal{T}'$ on X admit a temporal network and, if so, how the size of a so-called "maximum-temporal-agreement forest" for $\mathcal{T}$ and $\mathcal{T}'$ is related to the minimum-temporal-hybridization number for $\mathcal{T}$ and $\mathcal{T}'$. Roughly speaking, a maximum-temporal-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ of smallest size whose subtrees satisfy certain time constraints. Humphries et al. [12] showed that there exists a temporal network on X that displays $\mathcal{T}$ and $\mathcal{T}'$ if and only if there exists a so-called "temporal-agreement forest" for $\mathcal{T}$ and $\mathcal{T}'$, in which case,

$$h_t(\mathcal{T}, \mathcal{T}') = |\mathcal{F}| - 1,$$

where $|\mathcal{F}|$ is the size of a maximum-temporal-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$. For details, see [12].

3 Cherry-Picking Sequences

In this section, we establish a novel characterization of $h_t(\mathcal{P})$ for a set $\mathcal{P}$ of trees on X and show how it can be used to compute this minimum.

Let $\sigma = (x_1, x_2, \ldots, x_p)$ be an arbitrary sequence. For $i, j \in [p]$ with $i \leq j$, we call the sequence $(x_i, x_{i+1}, \ldots, x_j)$ a *substring* of σ. We say that an ordering of the elements in X, say $(x_1, x_2, \ldots, x_n)$, is a *cherry-picking sequence* for $\mathcal{P}$ precisely if each x_i with $i \in [n-1]$ labels a leaf of a cherry in each tree that

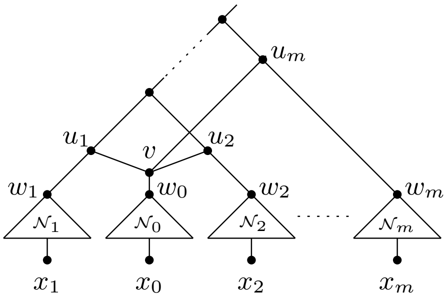

Fig. 4 Setup of $\mathcal{N}$ in the proof of Lemma 1.

is contained in $\mathcal{P}[-x_1, x_2, \ldots, x_{i-1}]$. For example, Figure 3 shows three trees for which (f, g, d, b, a, e, c) is a cherry-picking sequence.

The first theorem in this section, which requires the following lemma, characterizes whether or not there exists a temporal network that displays a set of trees on X.

Lemma 1 *Let $\mathcal{P}$ be a set of trees on X, and suppose that there is a temporal network $\mathcal{N}$ on X that displays $\mathcal{P}$. Then there exists an element of X that labels a leaf of a cherry in each tree of $\mathcal{P}$.*

Proof Let $\mathcal{N}$ be a temporal network on X that displays $\mathcal{P}$ and whose temporal labeling is t. Let v be a hybridization vertex for which the assigned temporal labeling is maximized. Furthermore, let (v, w_0) be the unique tree arc of $\mathcal{N}$ that is incident with v, and, for each $j \in [m]$, let (u_j, v) be a hybridization arc of $\mathcal{N}$ that is used to display $\mathcal{T}_j \in \mathcal{P}$ in $\mathcal{N}$. Note that, for $j, j' \in [m]$, we may have $u_j = u_{j'}$. Now, for each $j \in [m]$, obtain the subnetwork $\mathcal{N}_j$ with root w_j by deleting the tree arc (u_j, w_j), and obtain the subnetwork $\mathcal{N}_0$ with root w_0 by deleting the tree arc (v, w_0) (see Figure 4). For any $i \in \{0, 1, 2, \ldots, m\}$, note that, by property (v) in the definition of a network and the maximality of $t(v)$, the aforementioned tree arc exists and each $\mathcal{N}_i$ is a rooted binary phylogenetic tree that is pendant in $\mathcal{P}$. Now, if there exists an $\mathcal{N}_i$ in the multiset $\{\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_m\}$ that contains at least two leaves, the claim clearly follows. Therefore, we may assume that each $\mathcal{N}_i$ consists of a single leaf labeled x_i. As (u_j, v) is used to display $\mathcal{T}_j$ in $\mathcal{N}$, it follows that $\{x_0, x_j\}$ is a cherry of $\mathcal{T}_j$ for each j. Hence, x_0 is a leaf of a cherry in each tree of $\mathcal{P}$. This completes the proof of the lemma. □

Theorem 1 *Let $\mathcal{P}$ be a set of trees on X. There exists a temporal network that displays $\mathcal{P}$ if and only if there exists a cherry-picking sequence for $\mathcal{P}$.*

Proof First, suppose that there exists a temporal network $\mathcal{N}$ that displays $\mathcal{P}$. To show that there exists a cherry-picking sequence for $\mathcal{P}$, we use induction on n. If $X = \{x_1\}$, then (x_1) is a cherry-picking sequence for $\mathcal{P}$. Now, assume that $n \geq 2$ and that the result holds for all sets of trees on X' with $|X'| < n$ for which a temporal network exists that displays them. By Lemma 1, there exists an element $x_1 \in X$ that labels a leaf of a cherry in each element of $\mathcal{P}$. Let $\mathcal{P}' = \{\mathcal{T}_j[-x_1] : \mathcal{T}_j \in \mathcal{P}\}$. Furthermore, let $\mathcal{N}'$ be the network obtained from $\mathcal{N}$ by deleting the vertex labeled x_1 and suppressing any resulting degree-2 vertex. By property (v) in the definition of a network, $\mathcal{N}'$ is indeed a network. Since $\mathcal{N}$ is a temporal network that displays $\mathcal{P}$, it is easily seen that $\mathcal{N}'$ is a temporal network that displays $\mathcal{P}'$. Thus, by the induction assumption, there exists a cherry-picking sequence $(x_2, x_3 \ldots, x_n)$ for $\mathcal{P}'$. Since x_1 labels a leaf of a cherry in each element of $\mathcal{P}$, it is then easily checked that $(x_1, x_2, x_3, \ldots, x_n)$ is a cherry-picking sequence for $\mathcal{P}$.

Second, suppose that $\mathcal{P}$ has a cherry-picking sequence. We show by induction on n that there exists a temporal network $\mathcal{N}$ that displays $\mathcal{P}$. If $X = \{x_1\}$, then (x_1) is a cherry-picking sequence for $\mathcal{P}$ and a single vertex labeled x_1 is a temporal network that displays $\mathcal{P}$. Now assume that $n \geq 2$ and that the result holds for all sets of trees on X' with $|X'| < n$ for which a cherry-picking sequence exists. Let $(x_1, x_2, \ldots, x_n)$ be a cherry-picking sequence for $\mathcal{P}$. Furthermore, let $\mathcal{P}' = \{\mathcal{T}_j[-x_1] : \mathcal{T}_j \in \mathcal{P}\}$. Evidently, $(x_2, x_3, \ldots, x_n)$ is a cherry-picking sequence for $\mathcal{P}'$. It now follows from the induction assumption that there exists a temporal network

$\mathcal{N}'$ that displays $\mathcal{P}'$. For each $j \in [m]$, let $\{x_1, z_j\}$ be the cherry of $\mathcal{T}_j \in \mathcal{P}$ that contains x_1, and let $e_j = (u_j, z_j)$ be the arc in $\mathcal{N}'$ that is directed into the leaf z_j. Note that for $j, j' \in [m]$, we may have $z_j = z_{j'}$. Now, let $S = \{(u_j, z_j) : j \in [m]\}$. If S contains precisely one element, say $e = (u_i, x_i)$, then obtain a network $\mathcal{N}$ from $\mathcal{N}'$ by subdividing e with a new vertex v and adjoining v to a new vertex labeled x_1 with a new arc. By setting $t(u_i) < t(v) < t(x_i)$, it follows that $\mathcal{N}$ is a temporal network that displays $\mathcal{T}$ and $\mathcal{T}'$. Otherwise, if $|S| > 1$, then obtain a network $\mathcal{N}$ from $\mathcal{N}'$ by creating a new vertex v and, for each $e_i \in S$ with $e_i = (u_i, x_i)$, subdividing e_i with a new vertex v_i and adjoining v_i and v with a new arc. Lastly, adjoin v to a new vertex labeled x_1 with a new arc. Again, by construction, $\mathcal{N}$ displays $\mathcal{P}$. Furthermore, set

$$t(v) = t(v_1) = t(v_2) = \cdots = t(v_{|S|})$$

so that

$$\max\{t(u_1), \ldots, t(u_{|S|})\} < t(v) < \min\{t(x_1), \ldots, t(x_{|S|})\}.$$

As the temporal labelings of the leaves $x_1, x_2, \ldots, x_{|S|}$ of $\mathcal{N}$ can arbitrarily be adjusted, such a temporal assignment exists, and it follows that $\mathcal{N}$ is also temporal.

□

Now, let $\sigma = (x_1, x_2, \ldots, x_n)$ be a cherry-picking sequence for $\mathcal{P}$. For each $i \in [n-1]$, let

$$c_i(\sigma) = \left| \bigcup_{j=1}^{m} \{\{x_i, z_j\}\} \right| - 1,$$

where $\{x_i, z_j\}$ is the cherry of $\mathcal{T}_j \in \mathcal{P}[-x_1, x_2, \ldots, x_{i-1}]$ that contains x_i, and let $c_n(\sigma) = 0$. We refer to $c_i(\sigma)$ as the *cherry count associated with* x_i. Note that $c_{n-1}(\sigma) = c_n(\sigma)$. Furthermore, we refer to

$$w(\sigma) = \sum_{i=1}^{n} c_i(\sigma)$$

as the *weight* of σ and call σ a *minimum-cherry-picking sequence* for $\mathcal{P}$ if $w(\sigma)$ is minimized over all cherry-picking sequences for $\mathcal{P}$. We denote this minimum number by $s(\mathcal{P})$. Referring back to Figure 3 the cherry-picking sequence $\sigma =$

(f, g, d, b, a, e, c) for the three trees shown in this figure has the following cherry counts: $c_3(\sigma) = c_6(\sigma) = c_7(\sigma) = 0$, $c_2(\sigma) = c_4(\sigma) = c_5(\sigma) = 1$, and $c_1(\sigma) = 2$. Thus, the weight of σ is $w(\sigma) = 5$.

The next theorem characterizes the temporal-hybridization number for a set $\mathcal{P}$ of trees if $\mathcal{P}$ admits a temporal network.

Theorem 2 *Let $\mathcal{P}$ be a set of trees on X such that there exists a temporal network that displays $\mathcal{P}$. Then $h_t(\mathcal{P}) = s(\mathcal{P})$.*

Proof We first show that $h_t(\mathcal{P}) \geq s(\mathcal{P})$. Let $\mathcal{N}$ be a temporal network that displays $\mathcal{P}$. The proof is by induction on $h_t(\mathcal{N})$. If $h_t(\mathcal{N}) = 0$, then all trees in $\mathcal{P}$ are isomorphic and the result clearly follows. Now assume that $h_t(\mathcal{N}) = k > 0$ and that the result holds for all sets of trees that can be displayed by a temporal network whose temporal-hybridization number is less than k. Let v be a hybridization vertex of $\mathcal{N}$ whose temporal labeling is maximized over all such vertices in $\mathcal{N}$. Furthermore, let $\mathcal{L}_v$ be the subset of X that precisely contains the elements that are descendants of v in $\mathcal{N}$. Now, delete v in $\mathcal{N}$ and its incident arcs and, subsequently, suppress all resulting vertices with in-degree and out-degree 1. Let $\mathcal{N}'$ be the component that results from these operations and contains the original root of $\mathcal{N}$. Furthermore, let $\mathcal{N}''$ be the other resulting component. It is easily checked that $\mathcal{N}'$ is a temporal network for $\mathcal{P}' = \mathcal{P}[-\mathcal{L}_v]$ with $h_t(\mathcal{N}') = h_t(\mathcal{N}) - (d^-(v) - 1)$, and $\mathcal{N}''$ is a temporal network for $\mathcal{P}'' = \mathcal{P}|\mathcal{L}_v$ with $h_t(\mathcal{N}'') = 0$. Since $d^-(v) \geq 2$, it now follows by the induction assumption that $s(\mathcal{P}') \leq h_t(\mathcal{P}') \leq h_t(\mathcal{N}')$ and, trivially, that $s(\mathcal{P}'') \leq h_t(\mathcal{P}'') \leq h_t(\mathcal{N}'')$.

By Theorem 1, there exist cherry-picking sequences for $\mathcal{P}$, $\mathcal{P}'$, and $\mathcal{P}''$. Let $\sigma' = (x_1, x_2, \ldots, x_l)$ be a minimum-cherry-picking sequence for $\mathcal{P}'$, and let $\sigma'' = (y_1, y_2, \ldots, y_{l'})$ be a minimum-cherry-picking sequence for $\mathcal{P}''$. It now follows that

$$\sigma = (y_1, y_2, \ldots, y_{l'-1}, x_1, x_2, \ldots, x_{i-1}, y_{l'}, x_i, x_{i+1}, \ldots, x_l)$$

is a cherry-picking sequence for $\mathcal{P}$, where $i \in [l]$ such that $y_{l'}$ labels a leaf of a cherry in each tree of $\mathcal{P}[-y_1, y_2, \ldots, y_{l'-1}, x_1, x_2, \ldots, x_{i-1}]$. Furthermore, as $\mathcal{N}$ is temporal and displays $\mathcal{P}$, we have $c_{(l'-1)+(i-1)+1}(\sigma) \leq d^-(v) - 1$ while each other element of σ has the same cherry count in σ as in σ' and σ'', respectively. Thus,

$$h_t(\mathcal{N}) = h_t(\mathcal{N}') + d^-(v) - 1 \geq s(\mathcal{P}') + c_i(\sigma) \geq s(\mathcal{P}).$$

Choosing $\mathcal{N}$ to be a minimum-temporal-hybridization network for $\mathcal{P}$ establishes the first direction of the proof and, hence, $h_t(\mathcal{P}) \geq s(\mathcal{P})$.

To complete the proof, let $\sigma = (x_1, x_2, \ldots, x_n)$ be a cherry-picking sequence for $\mathcal{P}$. We show by induction on $w(\sigma)$ that $h_t(\mathcal{P}) \leq s(\mathcal{P})$. If $w(\sigma) = 0$, then it is easily seen that all trees in $\mathcal{P}$ are isomorphic and so the result holds. Now assume that $w(\sigma) = k > 0$ and that the result holds for all sets of trees that have a cherry-picking sequence whose weight is less than k. Let $i \in [n]$ be the smallest index of an element in σ such that $c_i(\sigma) \geq 1$. Furthermore, let $\mathcal{P}' = \mathcal{P}[-x_1, x_2, \ldots, x_i]$. As σ is a cherry-picking sequence for $\mathcal{P}$, it follows that $\sigma' = (x_{i+1}, x_{i+2}, \ldots, x_n)$ is such a sequence for $\mathcal{P}'$ and thus $w(\sigma') = w(\sigma) - c_i(\sigma)$. It now follows by the induction assumption that $h_t(\mathcal{P}') \leq s(\mathcal{P}') \leq w(\sigma')$.

Let $\mathcal{N}'$ be a minimum-temporal network that displays $\mathcal{P}'$. Note that $h_t(\mathcal{N}') \leq w(\sigma')$. Furthermore, by the definition of the cherry count $c_i(\sigma)$, there exist precisely $c_i(\sigma) + 1$ distinct cherries containing x_i among the trees in $\mathcal{P}[-x_1, x_2, \ldots, x_{i-1}]$. For each $l \in [c_i(\sigma)+1]$, let $\{x_i, z_l\}$ be such a cherry. We next construct a temporal network that displays $\mathcal{P}$ and whose temporal-hybridization number is $w(\sigma)$. First, obtain $\mathcal{N}''$ from $\mathcal{N}'$ by creating new vertices v and x_i that are joined via a new arc (v, x_i), and then, for each $l \in [c_i(\sigma)+1]$, subdividing the arc (w_l, z_l) incident with the leaf labeled z_l with a new vertex u_l and adding the arc (u_l, v). Since $c_i(\sigma) + 1$ distinct pendant arcs have been subdivided to obtain $\mathcal{N}''$ from $\mathcal{N}'$, it follows that, by setting $t(v) = t(u_1) = t(u_2) = \ldots = t(u_{c_i(\sigma)+1})$ so that

$$\max_{l \in [c_i(\sigma)+1]} \{t(w_l)\} < t(v) < \min_{l \in [c_i(\sigma)+1]} \{t(z_l)\},$$

$\mathcal{N}''$ is a temporal network that displays $\mathcal{P}[-x_1, x_2, \ldots, x_{i-1}]$. Note that $d^-(v) = c_i(\sigma) + 1$. Second, obtain a temporal network from $\mathcal{N}''$ by attaching a vertex labeled x_{i-1} to a new vertex that subdivides a pendant arc of $\mathcal{N}''$ via a new arc such that the resulting network displays $\mathcal{P}[-x_1, x_2, \ldots, x_{i-2}]$. Since $c_{i-1}(\sigma) = 0$, this is always possible. Continuing in this way and in order for the vertices labeled $x_{i-2}, x_{i-3}, \ldots, x_1$, we eventually obtain a temporal network $\mathcal{N}$ that displays $\mathcal{P}$. Moreover, we have

$$w(\sigma) = w(\sigma') + c_i(\sigma) \geq h_t(\mathcal{N}') + d^-(v) - 1 \geq h_t(\mathcal{P}).$$

Now, by choosing σ to be a minimum-cherry-picking sequence for $\mathcal{P}$, it follows that $h_t(\mathcal{P}) \leq s(\mathcal{P})$, thereby establishing the theorem. □

4 Fixed-parameter tractability of 2-Minimum-Temporal Hybridization

In this section, we give a brief overview on how to establish a fixed-parameter algorithm for 2-Minimum-Temporal Hybridization. For the details of this algorithm and two extensions of it that are likely to have a positive impact on practical running times, we refer the interested reader to the online appendix to this paper, which can be accessed from the journal's website. The approach taken for establishing fixed-parameter tractability of 2-Minimum-Temporal Hybridization follows that of Bordewich and Semple [5], who showed that Hybridization Number is fixed-parameter tractable for when $|\mathcal{P}| = 2$. Building on this result, an improved kernel size has recently been described in [15]. It is possible to mimic the result of [5]—by using the agreement forest characterization established in [12]—to obtain a fixed-parameter algorithm for 2-Minimum-Temporal Hybridization by considering a new outcome that arises in one of the key lemmas [5, Lemma 3.1.2]. However, rather than using agreement forests, it is also possible to establish a fixed-parameter algorithm by using the more intuitive and potentially more efficient (e.g. when the input trees only have a small number of cherries) characterization in terms of cherry-picking sequences. In particular, it can be shown that reducing two trees $\mathcal{T}$ and $\mathcal{T}'$ on the same label set by repeated applications of the previously employed subtree reduction and the weighted version of the chain reduction [5] result in two trees $\mathcal{S}$ and $\mathcal{S}'$ whose label set size is linear in $h_t(\mathcal{T}, \mathcal{T}')$. Furthermore, without going into detail, the "weight" of a minimum-cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$ is the same as the "weight" of a minimum-cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. In summary, these results can be used to establish a fixed-parameter algorithm for 2-Minimum-Temporal Hybridization that is parameterized by $h_t(\mathcal{T}, \mathcal{T}')$. Unfortunately, despite the characterization of the minimum-temporal-hybridization number for an arbitrary sized set of rooted binary phylogenetic trees in terms of cherry-picking sequences, it remains an open problem to establish a fixed-parameter algorithm for Minimum-Temporal Hybridization.

We end this section by noting why it appears to be not possible to upgrade the approach taken by Bordewich and Semple [5, Lemma 3.3] to an arbitrarily large set $\mathcal{P}$ of trees on X in order to show that Hybridization Number (resp. Minimum-Temporal Hybridization) is fixed-parameter tractable. Their result bounds the number of leaves in a pair of reduced trees (that result from repeated applications of the aforementioned subtree and chain reduction to two trees $\mathcal{T}$ and $\mathcal{T}'$ on X) by a function that is linear in $h(\mathcal{T}, \mathcal{T}')$. However, using the same strategy for $\mathcal{P}$ results in a function that is linear in $h(\mathcal{P})$ and m. Since this function depends on m and, therefore, the size of the input, this approach is unlikely to be a key ingredient in establishing a fixed-parameter tractability result for Hybridization Number (resp. Minimum-Temporal Hybridization) unless m is bounded.

5 Conclusion

In this paper, we have presented the first characterization to compute the minimum-temporal-hybridization number for an arbitrarily large set $\mathcal{P}$ of trees. This characterization is based on the novel concept of cherry-picking sequences that are sequences of the leaves of the elements in $\mathcal{P}$. Furthermore, we have shown that cherry-picking sequences can be used to establish fixed-parameter tractability of 2-Minimum-Temporal Hybridization. A natural follow-up question is to investigate whether or not cherry-picking sequences can also be used to establish a fixed-parameter algorithm for Minimum-Temporal Hybridization for when m is unbounded. Moreover, it is worth noting that, if a temporal network exists for two trees $\mathcal{T}$ and $\mathcal{T}'$ on X, then $h_t(\mathcal{T},\mathcal{T}')$ and $h(\mathcal{T},\mathcal{T}')$ are not necessarily equal. For example, for the two trees $\mathcal{T}$ and $\mathcal{T}'$ shown in Figure 1, we have $h_t(\mathcal{T},\mathcal{T}') = 3$ and $h(\mathcal{T},\mathcal{T}') = 2$. Furthermore, while two trees always admit a hybridization network, they do not necessarily also admit a temporal network. It would therefore be interesting to investigate whether or not it is computationally hard to decide if two trees have a cherry-picking sequence and, consequently, admit a temporal network.

Acknowledgements We thank the referees for their helpful comments.

References

1. B. Albrecht, S. Linz, C. and Scornavacca, (2012), A first step toward computing all hybridization networks for two rooted binary phylogenetic trees, J. Comput. Biol. 19, 1227–1242.
2. M. Baroni, S. Grünewald, V. Moulton, and C. Semple, (2005), Bounding the number of hybridization events for a consistent evolutionary history, J. Math. Biol. 51, 171–182.
3. M. Baroni, C. Semple, and M. Steel, (2006), Hybrids in real time, Sys. Biol. 44, 46–56.
4. M. Bordewich, S. Linz, K. St. John, and C. Semple, (2007), A reduction algorithm for computing the hybridization number of two trees, Evol. Bioinform. 3, 86–98.
5. M. Bordewich and C. Semple, (2007), Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable, IEEE Trans. Comput. Biol. Bioinf. 4, 458–466.
6. G. Cardona, F. Rossello, and G. Valiente, (2009), Comparison of tree-child phylogenetic networks, IEEE Trans. Comput. Biol. Bioinf. 6, 552–569.
7. Z.Z. Chen and L. Wang, (2012), Algorithms for reticulate networks of multiple phylogenetic trees, IEEE Trans. Comput. Biol. Bioinf. 9, 372–384.
8. Z.Z. Chen and L. Wang (2013). An ultrafast tool for minimum reticulate networks. J. Comput. Biol. 20, 38–41.

9. J. Collins, S. Linz, and C. Semple, (2011), Quantifying hybridization in realistic time, J. of Comp. Biol. 18, 1305–1318.

10. J. Gramm and R. Niedermeier, (2003), A fixed-parameter algorithm for minimum quartet inconsistency, J. Comput. Syst. Sci. 67, 723–741.

11. J. Gramm, A. Nickelsen, and T. Tantau, (2008), Fixed-parameter algorithms in phylogenetics, Comput. J. 51, 79–101.

12. P.J. Humphries, S. Linz, and C. Semple: On the complexity of computing the temporal hybridization number for two phylogenies, accepted for publication in Discrete Appl. Math.

13. D.H. Huson and C. Scornavacca, (2011), A survey of combinatorial methods for phylogenetic networks, Genome Biol. Evol. 3, 23–35.

14. L. van Iersel and S. Kelk, (2011), When two trees go to war, J. Theo. Biol. 269, 245–255.

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

16. S. Linz, C. Semple, and T. Stadler, (2010), Analyzing and reconstructing reticulation networks under timing constraints, J. Math. Biol. 61, 715–735.

17. T. Piovesan and S. Kelk: A simple fixed parameter tractable algorithm for computing the hybridization number of two (not necessarily binary) trees, IEEE Trans. Comput. Biol. Bioinf. 10, 18–25.

18. C. Whidden, R. G. Beiko, and N. Zeh: Fixed-parameter and approximation algorithms for maximum agreement forests, submitted.

19. Y. Wu, (2010), Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees, Bioinformatics 26, i140–i148.

20. Y. Wu and J. Wang, (2010), Fast computation of the exact hybridization number of two phylogenetic trees, Proceedings of the International Symposium on Bioinformatics Research and Applications, LNCS 6053, pp. 203–214.

Appendix A.1: Fixed-parameter tractability of 2-Minimum-Temporal Hybridization

In this appendix, we provide full details of how to establish the fixed-parameter tractability of 2-Minimum-Temporal Hybridization. In fact, we prove the following theorem.

Theorem 3 *The decision problem* 2-Minimum-Temporal Hybridization*, parameterized by* $h_t(\mathcal{T}, \mathcal{T}')$*, is fixed-parameter tractable.*

The appendix is organized as follows. We start by providing some notation and terminology that is needed to establish Theorem 3. Subsequently, we describe two reductions that are used to reduce the size of an instance of 2-Minimum-Temporal Hybridization and establish three results that play an important role in showing that the algorithm 2-MinTempHybrid, which is presented thereafter, is correct. We then investigate the running time of 2-MinTempHybrid from which the fixed-parameter tractability of 2-Minimum-Temporal Hybridization immediately follows. Lastly, we describe an improved version of the algorithm 2-MinTempHybrid that yields a better theoretical running time and finish off with an additional reduction that is likely to have a further positive impact on the practical running time.

Let $\mathcal{T}$ be a tree on X. Let a be an element in X. The unique vertex u of $\mathcal{T}$ such that $\{u, a\}$ is an edge is called the *parent* of a and is denoted by $p_\mathcal{T}(a)$. Moreover, a tuple $(a_1, a_2, \ldots, a_t)$ of leaves of $\mathcal{T}$ with $t \geq 2$ is called a *chain* of $\mathcal{T}$ if the following two conditions are satisfied:

(1) either $p_\mathcal{T}(a_1) = p_\mathcal{T}(a_2)$ or $p_\mathcal{T}(a_1)$ is a child of $p_\mathcal{T}(a_2)$;
(2) for each $i \in \{2, 3, \ldots, t-1\}$, $p_\mathcal{T}(a_i)$ is a child of $p_\mathcal{T}(a_{i+1})$.

In particular, we call $(a_1, a_2, \ldots, a_t)$ a *t-chain* if $t \geq 3$ and a *2-chain* if $t = 2$. As we will soon see, a reduction of each t-chain that is common to a pair of trees on

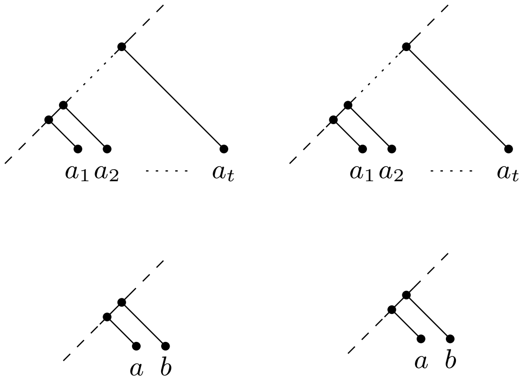

Fig. 5 Top: Two trees $\mathcal{T}$ and $\mathcal{T}'$ on X with a common t-chain $(a_1, a_2, \ldots, a_t)$. Bottom: Two trees $\mathcal{S}$ and $\mathcal{S}'$ that have been obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by applying a chain reduction.

X to a 2-chain will play an important role in establishing Theorem 3. To make the reduction work, it is necessary to keep track of the length of each original t-chain. We describe how this will be achieved next.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X. Furthermore, let P be a disjoint collection of 2-element subsets of X such that each pair $\{a, b\}$ is a 2-chain that is common to $\mathcal{T}$ and $\mathcal{T}'$. Let $w : P \rightarrow \mathbb{Z}^+$ be a weight function on the elements of P. We refer to $\mathcal{T}$ and $\mathcal{T}'$ with associated set P and weight function w as a pair of *weighted trees on* X.

We are now in a position to state two reductions that are crucial in establishing the correctness of Theorem 3. We note that these reductions have been used previously to establish fixed-parameter tractability of a related problem [5]. Let $\mathcal{T}$ and $\mathcal{T}'$ be two weighted trees on X.

(i) Subtree Reduction. Replace any maximal pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$ with at least two leaves by a single leaf with a new label.

(ii) Chain Reduction. Replace any maximal t-chain of $\mathcal{T}$ and $\mathcal{T}'$ with a two chain (a, b) and add the 2-element set $\{a, b\}$ to P with weight $w(\{a, b\}) = t - 2$.

Figure 5 illustrates an example of the chain reduction.

Let $\sigma = (x_1, x_2, \ldots, x_n)$ be a cherry-picking sequence for a pair of weighted trees $\mathcal{T}$ and $\mathcal{T}'$ on X. Since $c_i(\sigma) \leq 1$ for any $i \in [n-1]$, we say that x_i is *special* precisely if $c_i(\sigma) = 1$ and sometimes refer to special and not special as the two possible *statuses* of x_i. Furthermore, we use $s(\sigma)$ to denote the number of specials in σ. Furthermore, we define the *weight of* σ, denoted by $s_w(\sigma)$, to be

$$s_w(\sigma) = s(\sigma) + \sum_{\{a,b\} \in P; a \text{ is special in } \sigma} w(\{a, b\}),$$

and set $s_w(\mathcal{T}, \mathcal{T}')$ to be the minimum weight of a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. Note that we always have $s_w(\mathcal{T}, \mathcal{T}') \geq s(\mathcal{T}, \mathcal{T}')$ since the weight function is non-negative and $s_w(\mathcal{T}, \mathcal{T}') = s(\mathcal{T}, \mathcal{T}')$ whenever P is empty. Lastly, if $s_w(\sigma) = s_w(\mathcal{T}, \mathcal{T}')$, we say that σ is a *minimum-weight cherry-picking sequence.*

Let $\mathcal{T}$ and $\mathcal{T}'$ be two weighted trees on X, and let $A = \{a_1, a_2, \ldots, a_m\}$ be a subset of X. Furthermore, let σ be a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. We say that $O = (a'_1, a'_2, \ldots, a'_m)$ is the *ordering of* A *induced by* σ precisely if $(a'_1, a'_2, \ldots, a'_m)$ is a subsequence of σ. Typically, $a_1, a_2, \ldots, a_m$ are the elements of a chain that is common to $\mathcal{T}$ and $\mathcal{T}'$, in which case it is important to note that a'_i is not necessarily equal to a_i.

Lastly, let $\sigma = (x_1, x_2, \ldots, x_m)$ be a sequence, and let x_i be an element of σ with $i \in [m]$. We denote the substring $(x_1, x_2, \ldots, x_{i-1})$ by X_i^- and the substring $(x_{i+1}, x_{i+2}, \ldots, x_m)$ by X_i^+. For convenience, we sometimes write (X_i^-, x_i, X_i^+) when referring to σ. Note that, if $i = 1$, then X_i^- is empty while, if $i = m$, then X_i^+ is empty.

Lemma 2 *Let σ be cherry-picking sequence for two weighted trees $\mathcal{T}$ and $\mathcal{T}'$ on X. Let A be the leaf set of a pendant subtree that is common to $\mathcal{T}$ and $\mathcal{T}'$ with $|A| = m$, and let $(a_1, a_2, \ldots, a_{m-1})$ be the ordering of the first $m-1$ elements of the subtree induced by σ. Then each element in $A - \{a_m\}$ is not special in σ.*

Proof Since A is the leaf set of a pendant subtree of both $\mathcal{T}$ and $\mathcal{T}'$, it follows that, for each $i \in \{1, 2, \ldots, m-1\}$, a_i is in a common cherry of $\mathcal{T}[-A_i^-]$ and $\mathcal{T}'[-A_i^-]$ and, therefore, not special in σ. □

To prove a similar result (Lemma 4) for a common t-chain of two weighted trees, requires a lemma and a corollary.

Lemma 3 *Let σ be a minimum-weight-cherry-picking sequence for two trees $\mathcal{T}$ and $\mathcal{T}'$ on X. Furthermore, let $(a'_1, a'_2, \ldots, a'_t)$ be the ordering of the elements of a common t-chain A of $\mathcal{T}$ and $\mathcal{T}'$ induced by σ. If $\sigma = (A_i'^-, a'_i, b, B^+)$ with $a'_i \in \{a'_1, a'_2, \ldots, a'_{t-2}\}$ and $b \notin \{a'_1, a'_2, \ldots, a'_t\}$, then $\sigma' = (A_i'^-, b, a'_i, B^+)$ is a minimum-weight-cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ such that each element in σ is special if and only if it is special in σ'.*

Proof Let $A = (a_1, a_2, \ldots, a_t)$. We start with two observations that will implicitly be used throughout the proof. First, since $a'_i \notin \{a'_{t-1}, a'_t\}$ and $a_t \in \{a'_{t-1}, a'_t\}$, it follows that $a'_i \neq a_t$. Second, each element in $X - \{a'_i, b\}$ has the same status in σ and σ'. Now, let $\{a'_i, y\}$ be a cherry of $\mathcal{T}[-A_i'^-]$, and let $\{a'_i, z\}$ be a cherry of $\mathcal{T}'[-A_i'^-]$. Note that y and z may or may not be elements of A. The proof partitions into two cases.

Case 1. If $b \notin \{y, z\}$, then $\{a'_i, y\}$ is a cherry of $\mathcal{T}[-(A_i'^- \cup \{b\})]$ and $\{a'_i, z\}$ is a cherry of $\mathcal{T}'[-(A_i'^- \cup \{b\})]$. Hence, as σ is a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$, σ' is also such a sequence. Furthermore, as each of a'_i and b has the same status in σ' as in σ, it follows that σ' is of minimum weight.

Case 2. Suppose that $b \in \{y, z\}$. Without loss of generality we may assume that $b = y$. We next consider two subcases. First, assume that $y = z$. As $a'_i \in \{a'_1, a'_2, \ldots, a'_{t-2}\}$, this implies that both a'_i and b are not special in σ and σ'. Thus, as σ' is a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ and each of a'_i and b has the same status in σ' as in σ, it follows that σ' is of minimum weight. Second, assume that $z \neq y$. Then $p_{\mathcal{T}'}(b)$ is not a descendant of $p_{\mathcal{T}'}(a'_t)$ and so each of b and a'_i is special in σ and σ'. Thus σ' is also of minimum weight.

Combining both cases establishes the lemma. □

Repeated applications of Lemma 3 result in the following corollary. In particular, let $(a'_1, a'_2, \ldots, a'_t)$ be an ordering of the elements of a common t-chain of two trees $\mathcal{T}$ and $\mathcal{T}'$. Then, for each a'_i with $i \in \{1, 2, \ldots, t-2\}$, Lemma 3 is applied once for each element $b \notin \{a'_1, a'_2, \ldots, a'_t\}$ that succeeds a'_i and precedes a'_{t-1} in a minimum-weight-cherry picking sequence for $\mathcal{T}$ and $\mathcal{T}'$.

Corollary 1 *Let σ be a minimum-weight-cherry-picking sequence for two trees $\mathcal{T}$ and $\mathcal{T}'$ on X. Furthermore, let $(a'_1, a'_2, \ldots, a'_t)$ be the ordering of the elements of a common t-chain of $\mathcal{T}$ and $\mathcal{T}'$ induced by σ. Then there exists a minimum-weight-cherry-picking sequence σ' for $\mathcal{T}$ and $\mathcal{T}'$ such that $(a'_1, a'_2, \ldots, a'_{t-1})$ is a substring of σ' and each element in σ is special if and only if it is special in σ'.*

Lemma 4 *Let σ be a minimum-weight-cherry-picking sequence for two weighted trees $\mathcal{T}$ and $\mathcal{T}'$ on X. Let $(a'_1, a'_2, \ldots, a'_t)$ be the ordering of a common t-chain of $\mathcal{T}$ and $\mathcal{T}'$ induced by σ. Then all elements in $A = \{a'_1, a'_2, \ldots, a'_{t-1}\}$ have the same status in σ.*

Proof Assume that the lemma does not hold. Thus, there exists an element a'_i in A that is special in σ, and there exists an element a'_j in A that is not special in σ. Without loss of generality, choose the indices i and j to be as small as possible.

By Corollary 1, there exists a cherry-picking sequence σ' for $\mathcal{T}$ and $\mathcal{T}'$ of minimum weight such that the status for each of a'_i and a'_j in σ' is the same as its status in σ and $(a'_1, a'_2, \ldots, a'_{t-1})$ is a substring of σ'. We next establish the lemma for σ', thereby establishing the lemma for σ. We consider two cases:

Case 1. If $1 = j < i$, then σ' is of the form

$$(A'^{-}_j, a'_j, a'_{j+1}, \ldots, a'_{i-1}, a'_i, a'_{i+1}, \ldots, a'_{t-1}, A'^{+}_{t-1}).$$

Let $\{a'_j, b\}$ be a common cherry of $\mathcal{T}[-A'^{-}_j]$ and $\mathcal{T}'[-A'^{-}_j]$. If $b \notin A$, then, as $(a'_1, a'_2, \ldots, a'_{t-1})$ is a substring of σ', the cherry $\{a'_i, b\}$ is common to $\mathcal{T}[-A'^{-}_i]$ and $\mathcal{T}'[-A'^{-}_i]$, contradicting a'_i being special in σ'. On the other hand, if $b \in A$, then, as $i + 1 \leq t$, the cherry $\{a'_i, a'_{i+1}\}$ is common to $\mathcal{T}[-A'^{-}_i]$ and $\mathcal{T}'[-A'^{-}_i]$; again a contradiction.

Case 2. If $1 = i < j$, then σ' is of the form

$$(A'^{-}_i, a'_i, a'_{i+1}, \ldots, a'_{j-1}, a'_j, a'_{j+1}, \ldots, a'_{t-1}, A'^{+}_{t-1}).$$

Let $\{a'_i, b\}$ be a cherry of $\mathcal{T}[-A'^{-}_i]$, and let $\{a'_i, b'\}$ be a cherry of $\mathcal{T}'[-A'^{-}_i]$. Note that $b \neq b'$ and that at most one of b and b' is an element of $A \cup \{a'_t\}$. Without loss of generality, we may therefore assume by the choice of σ' that $b \notin A'^{-}_j$. Now, since $\{a'_i, b\}$ is a cherry of $\mathcal{T}[-A'^{-}_i]$, it follows that $\{a'_j, b\}$ is a cherry of $\mathcal{T}[-A'^{-}_j]$. Moreover, regardless of whether or not $b' \in A$, $\{a'_j, b\}$ is not a cherry of $\mathcal{T}'[-A'^{-}_j]$, contradicting a'_j being not special in σ'.

Combining both cases establishes the lemma. □

Now, let $\mathcal{T}$ and $\mathcal{T}'$ be a pair of trees on X for which a temporal network exists that displays them. The next proposition shows that the minimum weight of a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ is equal to the minimum weight of a cherry-picking sequence for the two trees obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by applying a subtree or chain reduction.

Proposition 1 *Let $\mathcal{T}$ and $\mathcal{T}'$ be a pair of weighted trees on X for which a temporal network exists that displays them. Furthermore, let $\mathcal{S}$ and $\mathcal{S}'$ be a pair of weighted trees obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by either a single subtree or chain reduction. Then $s_w(\mathcal{T}, \mathcal{T}') = s_w(\mathcal{S}, \mathcal{S}')$.*

Proof We first show that the lemma holds when $\mathcal{S}$ and $\mathcal{S}'$ are obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by a applying a subtree reduction; i.e. replacing a maximal pendant subtree with leaf set A that is common to $\mathcal{T}$ and $\mathcal{T}'$ with a single leaf labeled a. Let $|A| = m$. First, let σ be a minimum-weight-cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$, and let j be the maximum index of an element in σ such that $x_j \in A$. Obtain a sequence σ' from σ by removing each element x_i with $x_i \in A$ and $i \neq j$, and replacing x_j with a. It is easily checked that σ' is a cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$. Furthermore, x_j is special in σ if and only if a is special in σ'. By Lemma 2, each element in $A - \{x_j\}$ is not special in σ and, thus,

$$s_w(\mathcal{T}, \mathcal{T}') = s_w(\sigma) = s_w(\sigma') \geq s_w(\mathcal{S}, \mathcal{S}'). \tag{1}$$

Second, let σ' be a minimum-weight-cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$. Obtain a sequence σ from σ' by replacing a with the string $(x_1, x_2, \ldots, x_m)$ that contains precisely all the elements in A such that, for each $i \in \{1, 2, \ldots, m-1\}$, x_i labels a leaf of a common cherry in $\mathcal{T}|(A - \{x_1, x_2, \ldots, x_{i-1}\})$ and $\mathcal{T}'|(A - \{x_1, x_2, \ldots, x_{i-1}\})$. Since A is the leaf set of a pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$, this is always possible. Now, as σ' is a cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$, we have that σ is such a sequence for $\mathcal{T}$ and $\mathcal{T}'$. Moreover, since a is special in σ' if and only if x_m is special in σ, we have

$$s_w(\mathcal{S}, \mathcal{S}') = s_w(\sigma') = s_w(\sigma) \geq s_w(\mathcal{T}, \mathcal{T}'). \tag{2}$$

Combining (1) and (2) establishes the proposition for a single application of the subtree reduction.

We complete the proof of this lemma by showing that the result holds for when $\mathcal{S}$ and $\mathcal{S}'$ are obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by applying a chain reduction; i.e. replacing a maximal common t-chain $(a_1, a_2, \ldots, a_t)$ of $\mathcal{T}$ and $\mathcal{T}'$ with the 2-chain (a, b) with weight $w(\{a, b\}) = t - 2$. Let $A = \{a_1, a_2, \ldots, a_t\}$. First, let σ be a minimum-weight-cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$, and let j be the maximum index of an element of A in σ. By Lemma 4, one of the following holds:

(i) each element of $A - \{a_j\}$ is not special in σ or

(ii) each element of $A - \{a_j\}$ is special in σ.

Let i be the minimum index of an element of A in σ. Now, obtain a sequence σ' from σ by replacing a_i with a, replacing a_j with b, and removing all elements in $A - \{a_i, a_j\}$. Since σ is a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$, it follows that σ' is such a sequence for $\mathcal{S}$ and $\mathcal{S}'$. Moreover, if σ satisfies (1), then a is not special in σ', and if σ satisfies (2), then a is special in σ'. Furthermore, recall that $w(\{a, b\})$ contributes to $s_w(\sigma')$ if and only if a is special in σ'. It now follows from the definition of the weight of σ' that, in both cases, the contribution of specials in A to $s_w(\sigma)$ is precisely the same as the contribution of specials in $\{a, b\}$ to $s_w(\sigma')$. Hence,

$$s_w(\mathcal{T}, \mathcal{T}') = s_w(\sigma) = s_w(\sigma') \geq s_w(\mathcal{S}, \mathcal{S}'). \tag{3}$$

Second, let σ' be a minimum-weight-cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$. Obtain a sequence σ from σ' by replacing a with the string $a_1, a_2, \ldots, a_{t-1}$ and b with a_t. It is easily checked that, as σ' is a cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$, we have that σ is such a sequence for $\mathcal{T}$ and $\mathcal{T}'$. Depending upon the status of a and b, there are four cases to consider. First, if neither a nor b is special in σ', then no element in A is special in σ. Second, if a and b are both special in σ', then each element in A is special in σ. Third, if a is special in σ' and b is not special in σ', then each element in $A - \{a_t\}$ is special in σ while a_t is not special in σ. Fourth, if a is not special in σ and b is special in σ', then each element in $A - \{a_t\}$ is not special in σ while a_t is special in σ. It now follows from the definition of the weight of a cherry-picking sequence that, in all four cases, the contribution of specials in $\{a, b\}$ to $s_w(\sigma')$ is precisely the same as the contribution of specials in A to $s_w(\sigma)$. Thus, we have

$$s_w(\mathcal{S}, \mathcal{S}') = s_w(\sigma') = s_w(\sigma) \geq s_w(\mathcal{T}, \mathcal{T}'). \tag{4}$$

Combining (3) and (4) establishes the result for a single application of the chain reduction, and thereby establishes the proposition. □

We now give the pseudocode for the algorithm 2-MinTempHybrid that exactly solves instances of 2-Minimum-Temporal Hybridization.

We start with some new definitions. Let $\sigma = (x_1, x_2, \ldots, x_p)$ be a sequence, and let S be a set with $|S| = p'$. In the following, we write $\sigma || S$ to denote the sequence $(x_1, x_2, \ldots, x_p, y_1, y_2, \ldots, y_{p'})$, where the substring $(y_1, y_2, \ldots, y_{p'})$ is any ordering of the elements in S. Furthermore, we denote the tree formed by replacing a pendant subtree with leaf set A with a new leaf labeled a by $\mathcal{T}[A \rightarrow a]$. Similarly, we write $\mathcal{T}[(a_1, a_2, \ldots, a_t) \rightarrow (a, b)]$ for the tree obtained from $\mathcal{T}$ by replacing a t-chain $(a_1, a_2, \ldots, a_t)$ with the 2-chain (a, b).

The algorithm 2-MinTempHybrid takes as input two trees $\mathcal{T}$ and $\mathcal{T}'$ on X and a positive integer k, and outputs $h_t(\mathcal{T}, \mathcal{T}')$ precisely if $h_t(\mathcal{T}, \mathcal{T}') < k$; otherwise it outputs k. In particular, if no temporal network exists that displays $\mathcal{T}$ and $\mathcal{T}'$, then the algorithm always returns k. To this end, the algorithm exhaustively calculates all cherry-picking sequences for $\mathcal{T}$ and $\mathcal{T}'$ and, at each step, resets k to be the current best minimum weight over all cherry-picking sequences for $\mathcal{T}$ and $\mathcal{T}'$ that have previously been considered. For reasons of completeness, we include the pseudocode for the subroutines SubtreeReduction and ChainReduction that were first published in [4]. Furthermore, because of the chain reduction and the necessity of a weight function w, we initialize this function to be zero for all pairs of elements in X, which is indicated by $w \leftarrow ()$ in the pseudocode.

Algorithm 5.1: 2-MinTempHybrid($\mathcal{T}, \mathcal{T}', k$)

```
procedure SubtreeReduction($\mathcal{T}, \mathcal{T}', w$)
  $A \leftarrow$ maximal common pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$
  if $|A| > 1$
    do
      $\mathcal{T} \leftarrow \mathcal{T}[A \rightarrow a]$
      $\mathcal{T}' \leftarrow \mathcal{T}'[A \rightarrow a]$
      $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ SubtreeReduction($\mathcal{T}, \mathcal{T}', w$)
  return $(\mathcal{T}, \mathcal{T}', w)$

procedure ChainReduction($\mathcal{T}, \mathcal{T}', w$)
  $(a_1, \ldots, a_t) \leftarrow$ maximal common $t$-chain of $\mathcal{T}$ and $\mathcal{T}'$
  if $t \geq 3$
    do
      $w(a, b) \leftarrow t - 2$
      $\mathcal{T} \leftarrow (\mathcal{T}[(a_1, a_2, \ldots, a_t) \rightarrow (a, b)]$
      $\mathcal{T}' \leftarrow (\mathcal{T}'[(a_1, a_2, \ldots, a_t) \rightarrow (a, b)]$
      $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ ChainReduction($\mathcal{T}, \mathcal{T}', w$)
  return $(\mathcal{T}, \mathcal{T}', w)$

procedure CherryPicking($\mathcal{T}, \mathcal{T}', k, \sigma, w$)
  if $|\mathcal{L}(\mathcal{T})| = 2$
    then
      $\sigma \leftarrow \sigma || \mathcal{L}(\mathcal{T})$
      return $(s_w(\sigma))$
    else
      for each $x \in \mathcal{L}(\mathcal{T})$ that labels a leaf of a cherry in $\mathcal{T}$ and $\mathcal{T}'$
        do $k \leftarrow \min(k,$ CherryPicking($\mathcal{T}[-x], \mathcal{T}'[-x], k, \sigma || \{x\}, w$))
      return $(k)$

main
  $w \leftarrow ()$
  $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ SubtreeReduction($\mathcal{T}, \mathcal{T}', w$)
  $(\mathcal{T}, \mathcal{T}', w) \leftarrow$ ChainReduction($\mathcal{T}, \mathcal{T}', w$)
  if $\mathcal{L}(\mathcal{T}) \geq 9k$
    then return $(k)$
    else
      $\sigma \leftarrow ()$
      $k \leftarrow$ CherryPicking($\mathcal{T}, \mathcal{T}', k, \sigma, w$)
      return $(k)$
```

Remarks.

(1) Suppose that two trees $\mathcal{T}$ and $\mathcal{T}'$ on X can be displayed by a temporal network. Then $h_t(\mathcal{T},\mathcal{T}')$ is bounded from above by $n-2$. To see that this holds, let $\sigma = (x_1, x_2, ..., x_n)$ be a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. Then neither x_{n-1} nor x_n is special in σ because $\mathcal{T}|(x_{n-1}, x_n)$ and $\mathcal{T}'|(x_{n-1}, x_n)$ are isomorphic and, thus, $h_t(\mathcal{T},\mathcal{T}') = s(\mathcal{T},\mathcal{T}') \leq n-2$, where the equality follows from Theorem 2. Hence, calling 2-MinTempHybrid$(\mathcal{T},\mathcal{T}', n-1)$ either outputs $h_t(\mathcal{T},\mathcal{T}')$ or $n-1$ with the latter implying that no temporal network exists that displays $\mathcal{T}$ and $\mathcal{T}'$.

(2) As it is currently written, 2-MinTempHybrid does not construct a minimum-temporal network for $\mathcal{T}$ and $\mathcal{T}'$ if such a network exists. However, by simple bookkeeping, one could construct a minimum-weight-cherry-picking sequence $\sigma = (x_1, x_2, \ldots, x_n)$ for $\mathcal{T}$ and $\mathcal{T}'$ and then use the algorithm TemporalHybrid [16] to construct a minimum-temporal network for $\mathcal{T}$ and $\mathcal{T}'$. Briefly, for each $i \in [n]$, TemporalHybrid constructs a temporal network that displays $\mathcal{T}|\{x_i, x_{i+1}, \ldots, x_n\}$ and $\mathcal{T}'|\{x_i, x_{i+1}, \ldots, x_n\}$. Ultimately, this results in a temporal network for $\mathcal{T}$ and $\mathcal{T}'$. In particular, note that TemporalHybrid attaches each special element of σ via two new arcs to the previously reconstructed network while each element that is not special in σ is attached via a single arc. Hence, by Theorem 2, the reconstructed network is a minimum-temporal network for $\mathcal{T}$ and $\mathcal{T}'$.

(3) 2-MinTempHybrid first applies the subtree reduction until no such reduction is possible, then the chain reduction until no such reduction is possible, and then immediately proceeds with the exhaustive search part of the algorithm. The fact that this is indeed sufficient and that, after applying the chain reduction, no application of the subtree reduction is possible, follows from the next lemma.

Lemma 5 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X that do not have any common pendant subtree with at least two leaves, and let A be a maximal t-chain that is common* *to $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, let $\mathcal{S}$ and $\mathcal{S}'$ be the two trees that are obtained from $\mathcal{T}$ and $\mathcal{T}'$ by reducing A to the 2-chain (a, b). Then, the following hold:*

1. *$\mathcal{S}$ and $\mathcal{S}'$ do not have any common pendant subtree with at least two leaves, and*
2. *$\mathcal{S}$ and $\mathcal{S}'$ do not have a common chain with at least three leaves that contains a or b.*

Proof Let $A = (a_1, a_2, \ldots, a_t)$. The proof is by contradiction. First, assume that $\mathcal{S}$ and $\mathcal{S}'$ have a common subtree with leaf set A' such that $|A'| \geq 2$. Since $\mathcal{T}$ and $\mathcal{T}'$ do not contain any such subtree, a or b is an element of A'. But then $A' \cup \{a_1, a_2, \ldots, a_t\}$ is the leaf set of a common pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$; a contradiction. Now assume that $\mathcal{S}$ and $\mathcal{S}'$ have a common chain with at least three leaves such that one of them is labeled a or b. Then it is easily checked that A is not maximal in $\mathcal{S}$ and $\mathcal{S}'$; again a contradiction. This completes the proof of the lemma. □

The fact that 2-Minimum-Temporal Hybridization is fixed-parameter tractable (Theorem 3) is a direct consequence of the next theorem.

Theorem 4 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X, and let k be an integer. The algorithm* 2-MinTempHybrid *returns $h_t(\mathcal{T}, \mathcal{T}')$ if and only if there exists a temporal network that displays $\mathcal{T}$ and $\mathcal{T}'$ and $h_t(\mathcal{T}, \mathcal{T}') < k$; otherwise, it returns k. Furthermore, the running time of* 2-MinTempHybrid *is $O((9k)^{9k} + n^3)$.*

Proof The correctness of 2-MinTempHybrid immediately follows from Proposition 1, thereby establishing the first part of the theorem. We complete the proof by showing that the running time of 2-MinTempHybrid is $O((9k)^{9k} + n^3)$. Let X' be the leaf set of the two trees $\mathcal{S}$ and $\mathcal{S}'$ that are obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by first applying the subtree reduction as many times as possible and then the chain reduction as many times as possible. It follows from Lemma 5 that no further subtree or chain reduction can be applied to $\mathcal{S}$ and $\mathcal{S}'$. Now, Kelk et al. [15, Theorem 3.2] showed that $|X'| \leq 9h(\mathcal{T}, \mathcal{T}')$, where $h(\mathcal{T}, \mathcal{T}')$ is the minimum number of hybridization vertices of any hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$. Therefore, since each temporal network that displays $\mathcal{T}$ and $\mathcal{T}'$ is a particular type of a hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$, it follows by Theorem 2 that

$$|X'| \leq 9h(\mathcal{T}, \mathcal{T}') \leq 9h_t(\mathcal{T}, \mathcal{T}') = 9s(\mathcal{T}, \mathcal{T}'),$$

Hence, since X' is at most $9h_t(\mathcal{T}, \mathcal{T}')$, we declare $h_t(\mathcal{T}, \mathcal{T}') > k$ if $|X'| > 9k$. Now, suppose that $|X'| \leq 9k$. Since the procedure CherryPicking is recursively called at most $9k$ times to compute a cherry-picking sequence for $\mathcal{S}$ and $\mathcal{S}'$, each such call branches into at most $9k$ new recursive calls, and the subtree and chain reduction can both be computed in $O(n^3)$ [5], the running time of 2-MinTempHybrid is $O((9k)^{9k} + n^3)$. □

We now describe an improved version of 2-MinTempHybrid whose input and output are the same as those for 2-MinTempHybrid but whose theoretical worst-case running time is reduced. This speed-up is inspired by the results that are presented in [9]. Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X for which a cherry-picking sequence exists. We next algorithmically describe a sequence that is central to what follows. Set $i = 1$, $\mathcal{T}_0 = \mathcal{T}$, and $\mathcal{T}_0' = \mathcal{T}'$. Now obtain $\mathcal{T}_i$ and $\mathcal{T}_i'$ from $\mathcal{T}_{i-1}$ and $\mathcal{T}_{i-1}'$, respectively, by applying the subtree reduction until no such further reduction is possible. If $|\mathcal{L}(\mathcal{T}_i)| > 1$, choose x_i to be a leaf label of a cherry in $\mathcal{T}_i$ and $\mathcal{T}_i'$. Reset $\mathcal{T}_i$ to be $\mathcal{T}_i[-x_i]$ and, similarly, reset $\mathcal{T}_i'$ to be $\mathcal{T}_i'[-x_i]$. Increment i by 1, apply the subtree reduction until no such further reduction is possible, and repeat this process. Otherwise, if $|\mathcal{L}(\mathcal{T}_i)| = 1$, stop. Eventually, we obtain a sequence $\sigma = (x_1, x_2, \ldots, x_p)$ of the chosen x_i with $0 \leq p \leq n$ which we refer to as a *subtree-reduced-cherry-picking sequence*. Note that x_i is not necessarily an element of X since the subtree reduction replaces each common subtree by a single leaf with a new label. To illustrate, consider the two trees that are shown in Figure 1. Since $\{d, e\}$ is a cherry that is common to both of them, an application of the subtree reduction replaces this cherry with a single leaf with a new label, say {de}. The resulting two trees do not have any other common subtree. Since a labels a leaf of a cherry in both trees, we pick it to be the first element of a subtree-reduced-cherry-picking sequence σ for the original two trees and delete leaf a in both trees. We now repeat this process. Since we cannot apply the subtree reduction, we choose b to be the second element of σ and, again, delete the corresponding leaf in both trees. Since the resulting two trees have a common subtree whose label set is $\{c, de\}$, we replace this subtree with a new leaf labeled $\{cde\}$. Continuing in this way until the two trees have been reduced to a single vertex, we obtain $\sigma = (a, b, \{cde\}, f, g, h)$. Note that this sequence consists of only six elements.

Now, let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X. It follows from the definition of a minimum-cherry-picking sequence that $s(\mathcal{T}, \mathcal{T}')$ is equal to the length of a shortest subtree-reduced-cherry-picking sequence σ for $\mathcal{T}$ and $\mathcal{T}'$ plus the weight of any 2-chain (a, b) for which a is an element of σ. Calculating the minimum weight of a subtree-reduced-cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ instead of the minimum weight of a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ by repeatedly applying the subtree reduction as described in the last paragraph, and calling the resulting algorithm 2-MinTempHybrid*, we derive the following corollary.

Corollary 2 *The running time of* 2-MinTempHybrid* *is* $O((9k)^k n^3)$.

Proof The result can be established by following the argument in the proof of Theorem 4 and considering that CherryPicking is recursively called at most k times because each such call adds a special element to the subtree-reduced-cherry-picking sequence, thereby increasing $s_w(\sigma)$ by at least 1, where σ is the subtree-reduced-cherry picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ that is considered in some computational path of the algorithm. □

We end the appendix with the description of the so-called *cluster reduction* that splits an instance of 2-Minimum-Temporal Hybridization into a number of smaller and more tractable subproblems which is favorable for any exhaustive search. The cluster reduction has originally been described in [3] to calculate the (ordinary) minimum-hybridization number for two trees. Nevertheless, the next theorem shows that the cluster reduction can also be used in the context of calculating the minimum-temporal-hybridization number and is therefore likely to have a positive impact on practical running times, for example in combination with the algorithm 2-MinTempHybrid.

Before stating the theorem, we need a new definition. Let $\mathcal{T}$ be a tree on X, and let A be a subset of X. We call A a *cluster* of $\mathcal{T}$ if there is a vertex v in $\mathcal{T}$ whose set of descendants in X is precisely A.

Theorem 5 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees on X, and let $A \subset X$ be a cluster that is common to $\mathcal{T}$ and $\mathcal{T}'$. Then*

$$s(\mathcal{T}, \mathcal{T}') = s(\mathcal{T}|A, \mathcal{T}'|A) + s(\mathcal{T}_a, \mathcal{T}'_a),$$

where $\mathcal{T}_a$ and $\mathcal{T}'_a$ are the trees obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by replacing the pendant subtrees $\mathcal{T}|A$ and $\mathcal{T}'|A$ with a new vertex labeled a.

Proof We first show that

$$s(\mathcal{T}, \mathcal{T}') \leq s(\mathcal{T}|A, \mathcal{T}'|A) + s(\mathcal{T}_a, \mathcal{T}'_a). \tag{5}$$

Let $\sigma_A = (x_1, x_2, \ldots, x_m)$ be a cherry-picking sequence for $\mathcal{T}|A$ and $\mathcal{T}'|A$, and let $\sigma_a = (y_1, y_2, \ldots, y_{m'})$ be a cherry-picking sequence for $\mathcal{T}_a$ and $\mathcal{T}'_a$. Furthermore, let y_i with $i \in \{1, 2, \ldots, m'\}$ be the element of σ_a such that $y_i = a$. Then it is easily checked that

$$\sigma = (y_1, y_2, \ldots, y_{i-1}, x_1, x_2, \ldots, x_m, y_{i+1}, \ldots, y_{m'})$$

is a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, since A is the label set of a pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$, each element z in $X - \{x_m\}$ has the same status in σ as it does in σ_A if $z \in A$ and σ_a if $z \in X - A$, while x_m has the same status in σ as a has in σ_a. By recalling that a is not an element of σ, we deduce (5).

We next show that

$$s(\mathcal{T}, \mathcal{T}') \geq s(\mathcal{T}|A, \mathcal{T}'|A) + s(\mathcal{T}_a, \mathcal{T}'_a). \qquad (6)$$

Let σ be a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, let $\sigma_A = (x_1, x_2, \ldots, x_m)$ be the subsequence of σ that contains precisely all elements of A. Let i be the maximum index of an element in σ such that $x_i \in A$. Obtain $\sigma_a = (y_1, y_2, \ldots, y_{m'})$ from σ by replacing x_i with a and removing all elements in $A - \{x_i\}$. Note that x_i is an element of σ_A (in particular, $x_i = x_m$) and σ_a (in particular, $x_i = a$). Now, since σ is a cherry-picking sequence for $\mathcal{T}$ and $\mathcal{T}'$ and A is a cluster common to $\mathcal{T}$ and $\mathcal{T}'$, it follows that σ_A and σ_a are cherry-picking sequences for $\mathcal{T}|A$ and $\mathcal{T}'|A$, and $\mathcal{T}_a$ and $\mathcal{T}'_a$, respectively. Since A is the label set of a pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$, each element in $A - \{x_i\}$ has the same status in σ_A and σ while each element in $X - (A \cup \{x_i\})$ has the same status in σ_a and σ. If x_i is not special in σ, then a is not special in σ_a. Moreover, x_m is not special in σ_A because it is the last element and, therefore, not special by definition. Thus (6) holds.

Combining (5) and (6) establishes the theorem. □

For simplicity, we have presented the result of Theorem 5 for a pair of unweighted trees. Nevertheless, it only requires a minor modification to make it also work for a pair of weighted trees. Instead of considering an arbitrarily large common cluster, one now has to consider a minimal common cluster A, which is favorable in any case as it guarantees that the problem is broken up into as many smaller subproblems as possible. It is then straightforward to check that A either intersects any weighted 2-chain (a, b) that results from applying the chain reduction to a (maximal) t-chain in either both elements a and b or neither.