On the Complexity of Temporal Arborescence Reconfiguration

Riccardo Dondi

Università degli Studi di Bergamo, Italy

Manuel Lafond

Université de Sherbrooke, Canada

Abstract

We analyze the complexity of Arborescence Reconfiguration on temporal digraphs (Temporal Arborescence Reconfiguration). The problem, given two temporal arborescences in a temporal digraph, asks for the minimum number of arc flips, i.e. arc exchanges, that result in a sequence of temporal arborescences that transforms one into the other. We analyze the complexity of the problem, taking into account also its approximation and parameterized complexity, even in restricted cases. First, we solve an open problem showing that Temporal Arborescence Reconfiguration is NP-hard for two timestamps. Then we show that even if the two temporal arborescences differ only by two arcs, then the problem is not approximable within factor $b \ln |V(D)|$, for any constant $0 < b < 1$, where $V(D)$ is the set of vertices of the temporal arborescences. Finally, we prove that Temporal Arborescence Reconfiguration is W[1]-hard when parameterized by the number of arc flips needed to transform one temporal arborescence into the other.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Graph algorithms analysis; Mathematics of computing → Graph theory

Keywords and phrases Arborescence, Temporal Graphs, Graph Algorithms, Parameterized Complexity, Approximation Complexity

Digital Object Identifier 10.4230/LIPIcs.SAND.2024.10

Acknowledgements The authors thank the anonymous reviewers for their comments on the paper. The authors also thank Shun-ichi Maezawa for introducing the problem to the authors.

1 Introduction

Arborescences, also called branchings, have been deeply studied in theoretical computer science. Given a digraph (a directed graph) and a special vertex, called the root, an arborescence is a directed rooted tree in the digraph that connects the root to every vertex of the digraph. The computation of arborescences of a given digraph finds several applications, for example in communications networks, where the goal is to compute a shortest way to reach some devices [18], to analyze information flow in social networks [3], or in computational biology to analyze mass spectrometry data [7] and reconstruct tumor evolutionary trees [8].

Arborescences have been recently considered also in the temporal graph setting [15, 11, 4, 13], where they can model urban mobility or information dissemination in social networks. Temporal graphs have been studied to model the dynamic evolution of network relations (edges or arcs), that are observed only at certain time instants [17, 9, 19, 10, 1]. In our model of a temporal digraph $D = (V, A)$, the arcs are triples (u, v, t), where u and v are vertices and t is a positive integer, representing that the arc from u to v is seen at timestamp t. A *temporal arborescence* T in D is a rooted tree, whose arcs are directed away from the root, that contains every vertex of D and such that every path in T is *time-respecting*, that is the timestamps on the arcs of every path are non-decreasing.

In this contribution, we consider temporal arborescences through the lens of *combinatorial reconfiguration* [12, 21]. Given two feasible solutions of a problem (in our case being temporal arborescences of a temporal digraph), combinatorial reconfigurations explore the space of feasible solutions and the distance between the two given solutions. Two feasible solutions are adjacent if they can be transformed one into the other by means of a local operation (such as exchanging two arcs). The goal of combinatorial reconfiguration is to study the reachability of two elements of the space of feasible solutions, that is the possibility of transforming the first solution into the second one by means of sequences of local operations, and possibly obtaining a comparative metric by minimizing the number of such operations.

Given two temporal arborescences T_1 and T_2 in D, a reconfiguration of T_1 into T_2 is a transformation of T_1 into T_2 with a sequence of modifications, one at a time, called *arc flips*, where each modification exchanges two arcs. Note that an arc flip may exchange any two arcs of D, the only constraint is that, by applying arc flips, a reconfiguration may compute intermediate subgraphs, that must all be temporal arborescences in D.

We consider a problem related to the reconfiguration of temporal arborescences, called Temporal Arborescence Reconfiguration, introduced in [13]. Given a digraph D, and two arborescences T_1 and T_2 in D, Temporal Arborescence Reconfiguration asks to compute a reconfiguration of T_1 into T_2 consisting of the minimum number of operations. The problem is known to be NP-hard when the temporal graph is defined over 3 timestamps or more [13], and polynomial-time solvable when the number number of timestamps is 1, since in this case the digraph is static and for this case Temporal Arborescence Reconfiguration can be solved in polynomial time [14]. The case of 2 timestamps remained open [13].

An interesting property shown in [13], is that the complexity of Temporal Arborescence Reconfiguration depends on whether the two input temporal arborescences have the same root or not. In the former case, the problem is solvable in polynomial-time, while in the latter the problem is NP-hard, as discussed before.

A decision problem related to Temporal Arborescence Reconfiguration studied in the literature is the reachability of two feasible solutions, that is whether, given two temporal arborescences, one can be transformed into the other (without the requirement of minimizing the number of arc flips). This decision problem is solvable in polynomial time [13] and always admits a positive answer in static directed graphs [14] and when the two arborescences have the same root [13].

Our Results. In this paper we further analyze the complexity of Temporal Arborescence Reconfiguration, considering additional restrictions in the approximation and parameterized complexity frameworks. Note that we consider the temporal graph model of [13], which is a restricted model where each timestamp of an arc specifies its activation time and the arc is present for all times after the activation time. The hardness results we present hold also in this restricted model.

First, we solve the open problem in [13] for the case of two timestamps, and we show in Section 3 that this restriction of Temporal Arborescence Reconfiguration is NP-hard.

Then we consider the case when the two input temporal arborescences are very similar, that is they differ only for a limited number of arcs. We show in Section 4 that if the two temporal arborescences differ by two arc pairs, then the problem is not only NP-hard, but also inapproximable within factor $b \ln |V(D)|$, for any constant $0 < b < 1$, where $V(D)$ is the set of vertices of the arborescences. We also observe that if the two temporal arborescences differ for one pair of arcs, then the problem is easily solvable in polynomial time. Note that the result can be easily extended to the case where two temporal arborescences differ by more than two arc pairs. For example, we can replicate the construction of Fig. 2 and Fig. 3 by adding many copies of the subtree rooted at y and of the subtrees rooted at $v_{i,j}$. Each copy has to be reconfigured independently, thus the the inapproximation ratio is the same as in our result.

Finally, we consider the parameterized complexity of the problem, where the parameter is the number of arc flips required by a reconfiguration. We prove in Section 5 that the problem is W[1]-hard for this parameter (it is, in fact, W[1]-hard in the parameter "number of arc flips plus maximum timestamp"), indicating that a fixed-parameter algorithm is unlikely. We conclude the paper with Section 6 with some open problems. Note that some of the proofs are not included due to page limit.

2 Preliminaries

A *temporal digraph* $D = (V, A)$ is a pair where V is the set of vertices and $A \subseteq V \times V \times \mathbb{N}$ is a set of (temporal) arcs. Note that an arc in a temporal graph is denoted by a triple (u, v, t), where $u \in V$ is the tail of the arc, $v \in V$ is the head of the arc, and $t \in \mathbb{N}$ is called a timestamp. In our version of a temporal graph, an arc (u, v, t) remains *active* from this timestamp t, that is, once it is activated it exists in the temporal digraph from time t and onwards. We may write $V(D)$ and $A(D)$ for the vertex and arc set of D, respectively. Note that we allow multiple arcs between two vertices u and v, but they must be at different timestamps.

For a triple $e = (u, v, t)$, $D - e$ (resp. $D + e$) is the temporal digraph obtained by removing the arc e, if present (resp. adding the arc e, if absent).

An *arborescence* is a digraph in which there is a vertex u, called the *root*, such that there is a unique directed path from u to any vertex. In other words, an arborescence is a tree in which arcs are oriented away from the root. Let $D = (V, A)$ be a digraph. A subgraph T of D is a *spanning arboresence* of D if $V(T) = V(D)$ and T is an arborescence. Unless stated otherwise, all arborescences are spanning, and we may simply call T an arborescence of D.

Given a temporal graph D, a *temporal arborescence* T of D is an arborescence of D, such that T is *time-respecting*, that is for any pair of arcs $(u, v, t), (v, w, t') \in A(T)$ that are consecutive on some path of T, we have $t \leq t'$.

An *arc flip* on a temporal arborescence T of D is an operation that removes an arc $(u, v, t) \in A(T)$ and inserts an arc $(x, y, t') \in A(D) \setminus A(T)$, such that $T - (u, v, t) + (x, y, t)$ is a temporal arborescence of D (hence spanning and time-respecting).

A *reconfiguration* of a temporal arborescence T_1 of D is a sequence of arc flips, each one producing a temporal arborescence. A *reconfiguration* from T_1 to T_2 is a reconfiguration that transforms T_1 into T_2. A *reconfiguration sequence* $\mathcal{R} = (R_1, R_2, \ldots, R_l)$ from T_1 to T_2 is a sequence of temporal arborescences, where $R_1 = T_1$ and $R_l = T_2$ such that each R_i, with $i \in [l]$, is a temporal arborescence of D and each R_j, $j \in \{2, ..., l\}$, can be obtained from R_{j-1} with an arc flip.

Now, we are ready to define the problem we are interested into.

▶ **Problem 1.** (Temporal Arborescence Reconfiguration)
Input: *a temporal digraph D, two temporal arborescences T_1, T_2 of D, and an integer $p \geq 1$.*
Question: *Does there exist a reconfiguration from T_1 to T_2 of at most p arc flips?*

In the optimization version of Temporal Arborescence Reconfiguration, we aim to minimize the number of arc flips.

3 NP-Hardness for Two Timestamps

We show that the Temporal Arborescence Reconfiguration problem is NP-hard even on two timestamps (i.e. each arc has timestamp in $\{1,2\}$) via a reduction from the Set Cover problem. Let (S,U,k) be an instance of Set Cover, where U is the universe, S is a collection of subsets of U, and k is an integer. The question is whether there exists a subcollection $S^* \subseteq S$ of at most k sets of S such that for each $u \in U$ there exists at least one set of S^* that contains u.

We denote $U = \{u_1, \dots, u_n\}$ and $S = \{S_1, \dots, S_m\}$, and we define a corresponding instance $(D = (V,A), T_1, T_2, p)$ of Temporal Arborescence Reconfiguration. First let $S' = \{S'_i : S_i \in S\}$ be a copy of S and let $U' = \{u'_i : u_i \in U\}$ be a copy of U. We let

$$V = \{r_1, r_2, r_3\} \cup S \cup S' \cup U \cup U'.$$

We then add to A the following sets of arcs (we strongly recommend referring to Figure 1):

- $A_r = \{(r_1, r_2, 1), (r_2, r_1, 2), (r_1, r_3, 2), (r_3, r_2, 1)\}$;
- $A_{r_1,U} = \{(r_1, u_i, 1) : u_i \in U\}$;
- $A_{U,U'} = \{(u_i, u'_i, 1) : u_i \in U\}$;
- $A_{r_2,S} = \{(r_2, S_i, 2) : S_i \in S\}$;
- $A_{r_2,S'} = \{(r_2, S'_i, 1) : S_i \in S\}$;
- $A_{r_2,U} = \{(r_2, u_i, 2) : u_i \in S\}$;
- $A_{S,S'} = \{(S_i, S'_i, 2) : S_i \in S\}$;
- $A_{S',U'} = \{(S'_i, u'_j, 1) : S_i \in S \wedge u_j \in S_i\}$;
- $A_{r_3,S'} = \{(r_3, S'_i, 1) : S_i \in S\}$;
- $A_{r_3,U'} = \{(r_3, u'_i, 1) : u_i \in U\}$.

Note that $A_{S',U'}$ is the main set of arcs used to model the set cover instance into D. Finally, we define the input temporal arborescences T_1 (rooted at r_1) and T_2 (rooted at r_3) by specifying their arcs (illustrated in Figure 1, top-right and bottom-right, respectively):

$$A(T_1) = \{(r_1, r_2, 1), (r_1, r_3, 2)\} \cup A_{r_1,U} \cup A_{U,U'} \cup A_{r_2,S} \cup A_{S,S'}$$
$$A(T_2) = \{(r_3, r_2, 1), (r_2, r_1, 2)\} \cup A_{r_2,U} \cup A_{r_2,S} \cup A_{r_3,S'} \cup A_{r_3,U'}.$$

One can verify that T_1 and T_2 are temporal arborescences using Figure 1.

▶ **Theorem 1.** *The Temporal Arborescence problem is NP-hard even when the maximum timestamp of an arc is 2.*

Proof. Using the construction described above, we show that there exists $S^* \subseteq S$ of size at most k that covers U if and only if T_1 can be transformed into T_2 using at most $3n+m+2+k$ arc flips.

Suppose that there exists $S^* \subseteq S$ of size at most k that covers U. We reconfigure T_1 into T_2 as follows (we say that an arc flip is correct if, after applying it, the resulting subgraph is a temporal arborescence, hence time-respecting).

1. For each $S_i \in S^*$ in an arbitrary order, remove $(S_i, S'_i, 2)$ and add $(r_2, S'_i, 1)$.
   Each such arc flip is correct, since r_1 can reach S'_i through $(r_1, r_2, 1), (r_2, S'_i, 1)$.
2. For each $u'_j \in U'$ in an arbitrary order, let S_i be a set of S^* that contains u_j. Remove $(u_j, u'_j, 1)$ and add $(S'_i, u'_j, 1)$, which exists by construction.
   Each arc flip is correct since r_1 can reach u'_j through the path $r_1 \to r_2 \to S'_i \to u'_j$ using arcs of timestamp 1 only. Note that at this stage, r_2 reaches the vertices in S, S', and U' without going through r_1.

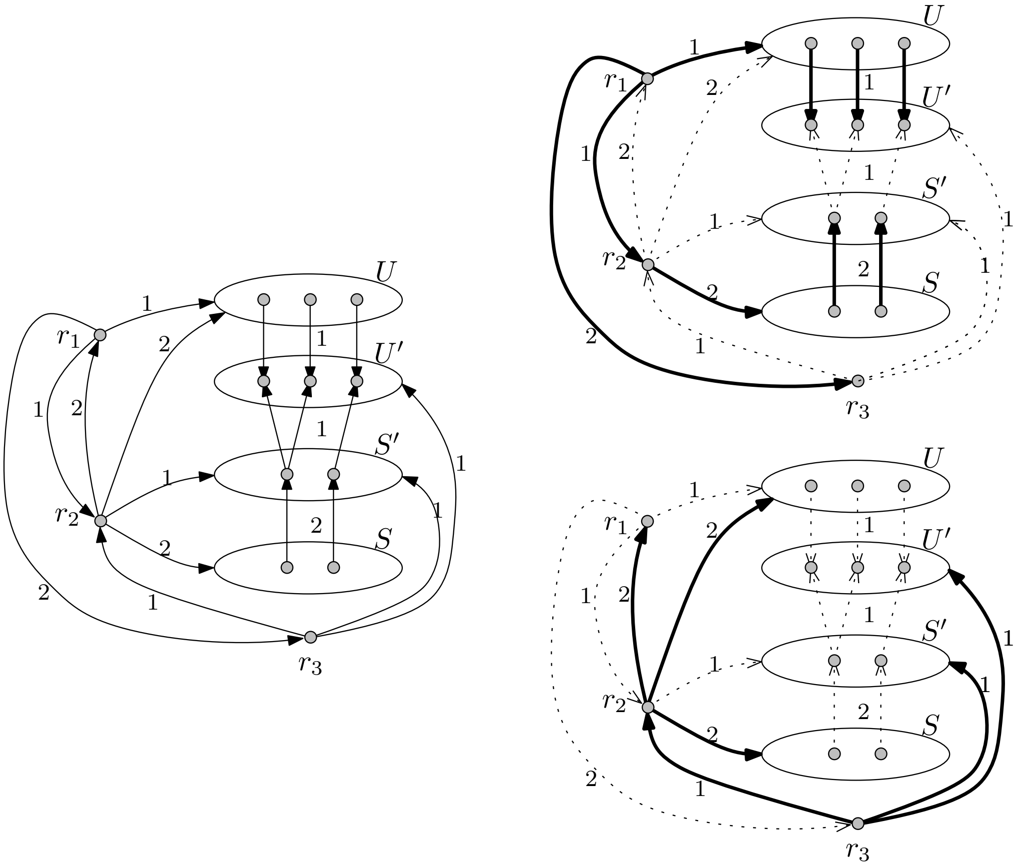

Figure 1 Left: the temporal digraph D obtained from a set cover instance, with $U = \{u_1, u_2, u_3\}$ and $S = \{S_1, S_2\}$, $S_1 = \{u_1, u_2\}$ and $S_2 = \{u_3\}$. Arcs pointing on ellipses indicate that all possible arcs are present (e.g. r_1 has every element of U in its out-neighborhood). The arborescences T_1 and T_2 are shown in thick arcs, top-right and bottom-right, respectively.

3. For each $u_j \in U$ in an arbitrary order, remove $(r_1, u_j, 1)$ and add $(r_2, u_j, 2)$. Each arc flip is correct since r_1 can reach u_j using the time-respecting path $r_1 \to r_2 \to u_j$. At this stage, r_2 also reaches the vertices of U without going through r_1.
4. Reroot to r_2 by removing $(r_1, r_2, 1)$ and adding $(r_2, r_1, 2)$. This arc flip is correct since before the arc flip, r_2 was already able to reach each element of U, U', S', S without r_1, and can now reach r_1 and r_3 through the time-respecting path $r_2 \to r_1 \to r_3$.
5. Reroot to r_3 by removing $(r_1, r_3, 2)$ and adding $(r_3, r_2, 1)$. This arc flip is correct since r_3 reaches r_2 at time 1, and thus r_3 can reach r_1, U, U', S', S through r_2 with a time-respecting path.
6. For $u'_j \in U'$ in an arbitrary order, remove the incoming arc incident to u'_j and add $(r_3, u'_j, 1)$. This is easily seen to be correct since U' vertices are leaves before (and after) the arc flips.
7. For $S'_i \in S'$ in an arbitrary order, remove the incoming arc incident to S'_i and add $(r_3, S'_i, 1)$. This is easily seen to be correct since, because of the previous step, the S' vertices are leaves before (and after) the arc flips.

One can check that this sequence of flips yields T_2. As for the number of arc flips, by summing the number of arc flips required for each of the above steps, we see that we require at most $|S^*| + |U'| + |U| + 1 + 1 + |U'| + |S'| \le k + 3n + m + 2$, as desired.

In the converse direction, suppose that there exists a reconfiguration sequence $\mathcal{R} = (R_1, R_2, \dots, R_l)$ from T_1 to T_2 with $l - 1 \leq 3n + m + 2 + k$, where $T_1 = R_1$ and $T_2 = R_l$, and each R_i can be obtained from R_{i-1} with an arc flip, for $i \in \{2, \dots, l\}$. We gather a set of facts to prove that U can be covered by at most k sets of S.

▶ **Fact 1.** *For each $i \in [l]$, the root of R_i is one of r_1, r_2, or r_3.*

Fact 1 holds because only r_1, r_2, and r_3 can reach r_1 in D.

▶ **Fact 2.** *If r_1 is the root of R_i for some $i \in [l]$, then $(r_1, r_3, 2) \in A(R_i)$ and $(r_1, r_2, 1) \in A(R_i)$.*

Fact 2 is true because $(r_1, r_3, 2)$ is the only incoming arc of r_3 and must thus be in R_i. This prevents using $(r_3, r_2, 1)$ because of the time-respecting condition. The only other incoming arc of r_2 is $(r_1, r_2, 1)$ and it must thus be in R_i as well.

▶ **Fact 3.** *If r_1 is the root of R_i for some $i \in [l-1]$, then r_3 is not the root of R_{i+1}.*

To see that Fact 3 holds, we know by Fact 2 that $(r_1, r_3, 2), (r_1, r_2, 1) \in A(R_i)$. To make r_3 the root in R_{i+1} we have to remove $(r_1, r_3, 2)$, and add some outcoming arc of r_3. But adding $(r_3, r_2, 1)$ makes r_2 of in-degree 2, and adding an arc from r_3 to some element of $S' \cup U'$ makes it impossible to reach r_1 from r_3. Therefore, the root of R_{i+1} is either r_1 or r_2.

We now proceed with the construction of a set cover. Let $a \in [l]$ be the minimum index such that r_2 is the root of R_a (note that there must exist such a R_a since the root of T_2 is r_3 and by Fact 3 the re-rooting from r_1 to r_3 cannot be done with an arc flip). By Fact 1 and Fact 3, we know that r_1 is the root of R_{a-1}, so that R_a is the first time the root is switched. By Fact 2, $(r_1, r_2, 1) \in A(R_{a-1})$ and, because $(r_2, r_1, 2)$ is the only incoming arc of r_1, the only way to switch the root from r_1 to r_2 is by removing $(r_1, r_2, 1)$ and adding $(r_2, r_1, 2)$. This means that in R_{a-1}, there cannot be an arc from r_1 to U, as otherwise $(r_2, r_1, 2)$ followed by such an arc would not be time-respecting. This implies that in R_{a-1}, all arcs from r_2 to U are present, since these are the only other incoming arcs of the U vertices. This in turn implies that in R_{a-1}, there cannot be an arc from U to U' because of the time-respecting condition. Also, by Fact 2, $(r_1, r_3, 2) \in A(R_{a-1})$ and the arcs from r_3 to U' cannot be used because of the time-respecting condition. Therefore, all in-neighbors of U' vertices are in S'. In fact by construction, for each $u'_j \in U'$, the in-neighbor of u'_j in R_{a-1} is some $S'_i \in S'$ such that $u_j \in S_i$. Since every $e \in A_{S',U'}$ is active at timestamp 1, every path from r_1 to a U' vertex in R_{a-1} only uses arcs of timestamps 1. Such a path cannot use an arc in which r_3 is the tail, again because of the $(r_1, r_3, 2)$ arc. Thus such a path must use an arc of $A_{r_2,S'}$. Let

$$S^* = \{S_i : (r_2, S'_i, 1) \in A(R_{a-1})\}.$$

Note that because each $u'_j \in U'$ has an S' in-neighbor such that the corresponding S set contains u_j, S^* is a set cover. It remains to argue that $|S^*| \leq k$.

Observe that $A(R_{a-1}) \setminus A(T_1)$ contains at least $|U| + |U'| + |S^*| = 2n + |S^*|$ arcs, since it has all arcs of $A_{r_2,U}$, the arcs from S' to U', and the arcs from r_2 to $\{S'_i : S_i \in S^*\}$. Thus at least $2n + |S^*| + 1$ arc flips are needed to get to R_a. Then, $A(T_2) \setminus A(R_a)$ contains at least $1 + |S'| + |U| = 1 + n + m$ arcs, namely $(r_3, r_2, 1)$ and the arcs from $A_{r_3,S'}$ and $A_{r_3,U'}$ (which are not in R_{a-1}, and thus not in R_a, because $(r_1, r_3, 2) \in A(R_{a-1})$ by Fact 2). Therefore, the number of arc flips required from T_1 to T_2 is at least $3n + m + 2 + |S^*|$, from which it follows that $|S^*| \leq k$.

Since Set Cover is known to be NP-hard [16], the reduction we have described implies that also Temporal Arborescence Reconfiguration for two timestamps is NP-hard. ◀

4 Inapproximability for Distance Two

In this section we show that, unless $P = NP$, Temporal Arborescence Reconfiguration is not approximable within factor $b \ln |V(D)|$, for any constant $0 < b < 1$, even if the two input temporal arborescences have distance two, that is that is the number of arcs in $A(T_1) \setminus A(T_2)$ and the number of arcs in in $A(T_2) \setminus A(T_1)$ is equal to two. We prove the result via an approximation preserving reduction from the Set Cover problem. Let (S, U) be an instance of Set Cover[1], where $U = \{u_1, \dots, u_n\}$ and $S = \{S_1, \dots, S_m\}$. Construct $(D = (V, A), T_1, T_2)$, an instance of Temporal Arborescence Reconfiguration associated with (U, S), as follows (refer to Fig. 2 for the structure of D).

$$V = \{r_1, r_2, y\} \cup \{v_{i,z} : S_i \in S, i \in [m], z \in [n^2]\} \cup \{w_i : i \in [n], u_i \in U\}.$$

A is defined as

$$A = A_1 \cup A_2 \cup A_3$$

where:

$$A_1 = \{(r_1, r_2, 2)\} \cup \{(r_1, y, 1)\} \cup \{(r_1, v_{i,1}, 4) : i \in [m]\} \cup \{(v_{i,j}, v_{i,j+1}, 4) : i \in [m], j \in [n^2 - 1]\} \cup \{(y, w_i, 2) : i \in [n]\}$$

$$A_2 = \{(r_2, r_1, 2)\} \cup \{(r_2, y, 1)\} \cup \{(r_1, v_{i,1}, 4) : i \in [m]\} \cup \{(v_{i,j}, v_{i,j+1}, 4) : i \in [m], j \in [n^2 - 1]\} \cup \{(y, w_i, 2) : i \in [n]\}$$

$$A_3 = \{(r_1, y, 3)\} \cup \{(r_1, v_{i,1}, 3) : i \in [m]\} \cup \{(v_{i,j}, v_{i,j+1}, 3) : i \in [m], j \in [n^2 - 1]\} \cup \{(v_{i,n^2}, w_j, 3) : u_j \in S_i, i \in [m], j \in [n]\}$$

Now, T_1 is the temporal arborescence induced by A_1, that is $T_1 = (V, A_1)$, and T_2 is the temporal arborescence induced by A_2, that is $T_2 = (V, A_2)$ (see Fig. 3). Note that $|A_1 \setminus A_2| = |A_2 \setminus A_1| = 2$, since $A_1 \setminus A_2 = \{(r_1, r_2, 2), (r_1, y, 1)\}$, while $A_2 \setminus A_1 = \{(r_2, r_1, 2), (r_2, y, 1)\}$.

We define a reconfiguration from T_1 to T_2 as *canonical* if it has the following properties. First, in some order, each w_i, $i \in [n]$, is disconnected from y as follows (we call this the *disconnection step* of the reconfiguration):

1. For some $j \in [m]$, each arc on the path from r_1 to v_{j,n^2}, associated with timestamp 4, is flipped with the arc having the same endpoints and timestamp 3 (starting from $(r_1, v_{j,1}, 4)$ and ending with $(v_{j,n^2-1}, v_{j,n^2}, 4)$).
2. Each arc $(y, w_i, 2)$, $i \in [n]$, is flipped with an arc $(v_{j,n^2}, w_i, 3)$, $j \in [m]$, so that there is a path from r_1 to v_{j,n^2} with all the arcs having timestamps 3.

Once the disconnection step is applied and each w_i, $i \in [n]$, is disconnected from y, a canonical reconfiguration flips arc $(r_1, y, 1)$ and $(r_1, y, 3)$. Then the root of the temporal arborescence is changed by flipping arcs $(r_1, r_2, 2)$ and $(r_2, r_1, 2)$. After these arc flips, $(r_1, y, 3)$ is flipped with arc $(r_2, y, 1)$. In order to compute T_2, each arc $(v_{j,n^2}, w_i, 3)$, $j \in [m]$ and $i \in [n]$, flipped in the disconnection step, is flipped with $(y, w_i, 2)$. Finally, for each path from r_1 to v_{j,n^2}, $j \in [m]$, having arcs with timestamp 3, each arc on the path is flipped with the arc having the same endpoints and timestamp 4 (starting from $(v_{j,n^2-1}, v_{j,n^2}, 3)$ and ending with $(r_1, v_{j,1}, 3)$).

We start by proving that a canonical reconfiguration is correct, that is it computes only temporal arborescences.

[1] Since in this section we consider optimization versions of problems, we do not include in the problem instances the value of a solution

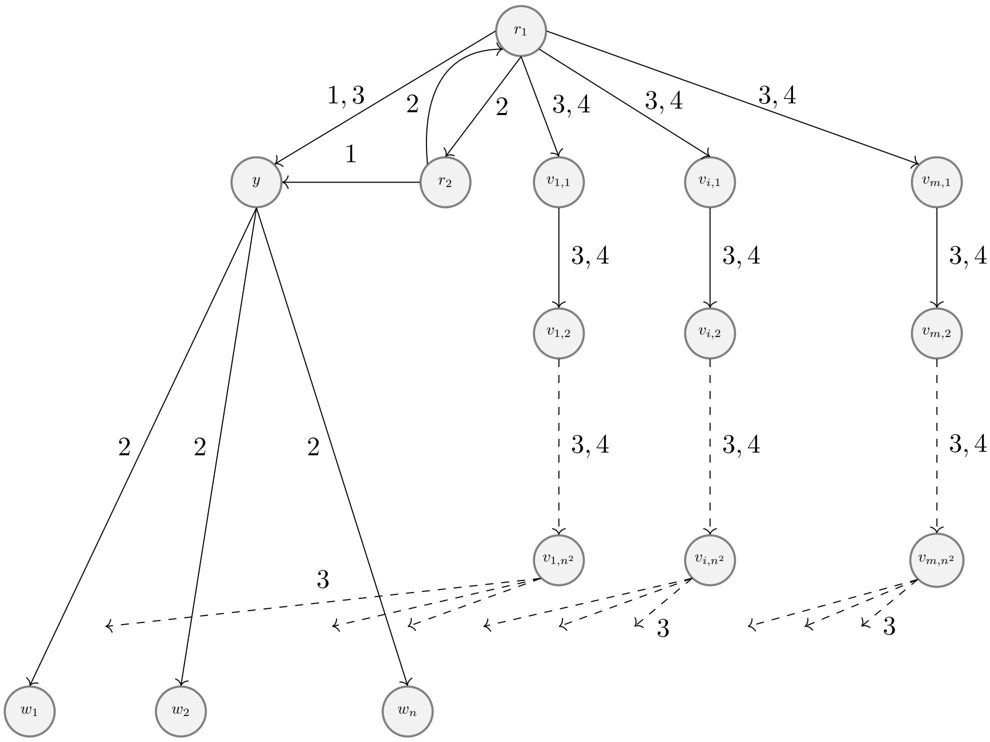

Figure 2 The input digraph D associated with an instance of SET COVER. Each dashed arrow outgoing from $v_{i,2}$, $i \in [m]$, represent a path containing vertices $v_{i,j}$, $j \in \{3, ..., n^2 - 1\}$, and not shown in the figure. The dashed arrows outgoing from v_{1,n^2}, v_{i,n^2}, v_{m,n^2} represent arcs connecting these vertices with some vertices w_z, $z \in [n]$ (the precise arcs depends on the instance of SET COVER).

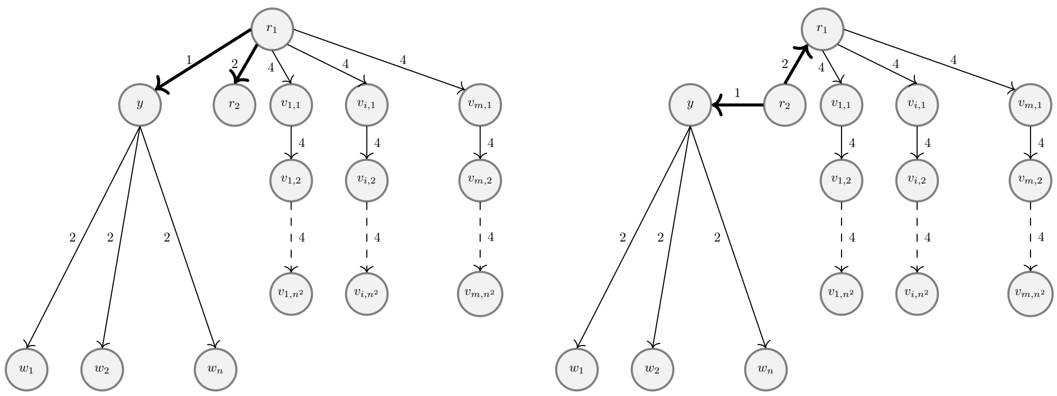

Figure 3 Arborescence T_1 (left) and T_2 (right). The four arcs in bold belong to exactly one the two temporal arborescence, the other arcs belong to both T_1 and T_2.

▶ **Lemma 2.** *Each arborescence computed by a canonical reconfiguration from T_1 to T_2 is a temporal arborescence of D.*

We prove now the first direction of the reduction.

▶ **Lemma 3.** *Let (S, U) be an instance of* Set Cover *and let (D, T_1, T_2) be the corresponding instance of* Temporal Arborescence Reconfiguration*. Given a a set cover of size k we can compute in polynomial time a reconfiguration from T_1 to T_2 consisting of $2kn^2 + 2n + 3$ flips.*

Now, we consider the second part of the reduction, where we prove that a reconfiguration from T_1 to T_2 must apply the disconnection step of a canonical reconfiguration.

▶ **Lemma 4.** *Let (S, U) be an instance of* Set Cover *and let (D, T_1, T_2) be the corresponding instance of* Temporal Arborescence Reconfiguration*. Given a reconfiguration from T_1 to T_2 consisting of $2kn^2 + 2n + 3$ arc flips we can compute in polynomial time a solution of* Set Cover *on instance (S, U) of size k.*

Proof. We start by proving that a reconfiguration from T_1 to T_2 must apply the disconnection step of a canonical reconfiguration.

First, consider arc $(r_1, r_2, 2)$ of T_1 and arc $(r_2, r_1, 2)$ of T_2. Note that $(r_1, r_2, 2)$ ($(r_2, r_1, 2)$, respectively) is the only arc of D incoming into r_2 (into r_1, respectively). Hence whenever $(r_1, r_2, 2)$ is flipped, and hence removed, by a reconfiguration, it must be flipped with $(r_2, r_1, 2)$, and r_2 must become the root of the computed temporal arborescence, otherwise either both r_1 and r_2 have not incoming arcs or r_2 is not connected with other vertices of the temporal arborescence. Note that this arc flip defines r_2 as the root of the computed arborescence and creates a temporal path $(r_2, r_1, 2), (r_1, y, 1)$, if this latter arc (of T_1) belongs to the arborescence, which is not time-respecting. It follows that before $(r_1, r_2, 2)$ and $(r_2, r_1, 2)$ are flipped, $(r_1, y, 1)$ must be flipped with another arc that must be incoming to y (since r_1 remains the root of the arborescence), that is with $(r_2, y, 1)$ or $(r_1, y, 3)$.

Consider $(r_2, y, 1)$ and notice that arcs $(r_1, y, 1)$ and $(r_2, y, 1)$ cannot be flipped, since this flip creates a temporal path $(r_1, r_2, 2), (r_2, y, 1)$, which is not time-respecting, and we have observed that $(r_1, r_2, 2)$ is not flipped before $(r_1, y, 1)$. Arcs $(r_1, y, 1)$ and $(r_1, y, 3)$ cannot be flipped unless y is a leaf, that is all the arcs $(y, w_i, 2)$, with $i \in [n]$, have been flipped. Indeed, if an arc $(y, w_i, 2)$, $i \in [n]$, belongs to a temporal arborescence, then by flipping $(r_1, y, 1)$ and $(r_1, y, 3)$ we have a temporal path $(r_1, y, 3), (y, w_i, 2)$, which is not time-respecting. It follows that, before $(r_1, y, 1)$ is flipped each vertex w_i, $i \in [n]$, must first be disconnected from y. By construction the only incoming arcs to a vertex w_i, $i \in [n]$, other than $(y, w_i, 2)$, are $(v_{j,n^2}, w_i, 3)$, for some $j \in [m]$, hence each vertex w_i must first be disconnected from y by flipping an arc $(y, w_i, 2)$ with an arc $(v_{j,n^2}, w_i, 3)$, for some $j \in [m]$. This implies that the disconnection step of the canonical reconfiguration is applied. This requires that each arc on the path from r_1 to v_{j,n^2}, which have timestamp 4 in T_1, is flipped with the arc having the same endpoints and timestamp 3.

Consider the temporal arborescence T' constructed by the disconnection step. For each w_i, $i \in [n]$, the disconnection step flips all the arcs of one path from r_1 to some v_{j,n_2}, $j \in [m]$, such that $u_i \in S_j$; then we can define a set cover as follows:

$$S^* = \{S_j : \text{the path from } r_1 \text{ to } w_{j,n^2} \text{ is modified in the disconnection step}\}.$$

We claim that S^* contains at most k sets. Note that a reconfiguration from T' to T_2 requires, as in a canonical reconfiguration, to delete arcs in $A(T') \setminus (A(T_2) \cap A(T_1))$ and insert arcs in $(A(T_2) \cap A(T_1)) \setminus A(T')$.

Recall that the reconfiguration of T_1 in T_2 consists of $2kn^2 + 2n + 3$ flips. If S^* consists of at least $k + 1$ sets, then by the definition of S^* the disconnection step includes at least $k + 1$ paths, thus requiring at least $2(k + 1)n^2$ arc flips for these paths, plus $2n$ arc flips for the arcs incident in w_i, $i \in [n]$. We have that $2(k + 1)n^2 + 2n > 2kn^2 + 2n + 3$, since $n \geq 2$. Hence S^* contains at most k sets, thus completing the proof. ◀

Based on Lemma 3 on Lemma 4, on the fact that the digraph D contains $O(n^2 m)$ vertices and on the hardness of approximation of Set Cover [2, 5, 20], we can prove the following result.

▶ **Theorem 5.** *Temporal Arborescence Reconfiguration is not approximable within factor $b \ln |V(D)|$, for any constant $0 < b < 1$, unless $P = NP$, even when the two input temporal arborescences differ for two pairs of arcs.*

Distance One

We have shown that Temporal Arborescence Reconfiguration is hard (also to approximate) when $T_1 = (V, A(T_1))$ and $T_2 = (V, A(T_2))$ have distance two. On the other hand when T_1 and T_2 have distance one, thus $A(T_1) \setminus A(T_2)$ contains a single arc a_1 and $A(T_2) \setminus A(T_1)$ contains a single arc a_2, the problem is easy to solve in polynomial time. Indeed, since by flipping a_1 with a_2 in T_1, hence by removing a_1 and inserting a_2, we obtain T_2, it follows that the arc flip produces a spanning time-respecting arborescence and thus can always be applied.

5 W[1]-Hardness

In all the above reductions (Section 3 and Section 4) and also the reduction in [13], the number of required arc flips is always a function of n. Therefore, an algorithm with complexity of the form $f(p)n^c$, with constant c and f only depending on p (number of arc flips of a reconfiguration from T_1 to T_2), is not excluded. We show that this is unlikely by proving that the Temporal Arborescence Reconfiguration problem is W[1]-hard under this parameter p, and that in fact it is W[1]-hard in parameter $p + \max_{(u,v,t) \in A(D)} t$.

We reduce Multicolored Clique to Temporal Arborescence Reconfiguration. Multicolored Clique, given an undirected graph $G = (V, E)$, whose vertices are colored with k colors, asks whether there exists a clique, called multicolored clique, containing one vertex from each color. The problem is W[1]-hard when the parameter is the number of colors [6].

Let $G = (V, E)$ be an instance a Multicolored Clique, with vertices partitioned into color classes $V_1, \ldots, V_k$. For $i, j \in [k]$, we will denote $E_{i,j} = \{uv \in E : u \in V_i, v \in V_j\}$. Construct an instance (D, T_1, T_2, p) of Temporal Arborescence Reconfiguration as follows.

Let us first construct D, which is shown in Figure 4 (we provide the main intuitions after the description of the construction). We define the vertex set of D as $V(D) = R \cup C \cup U$, where

$$R = \{r_0, r_1, \ldots, r_k\},$$
$$C = \{c_1, c_2, \ldots, c_k\},$$
$$U = \{u' : u \in V(G)\}.$$

For $i \in [k]$, we will denote $U_i = \{u' : u \in V_i\}$.

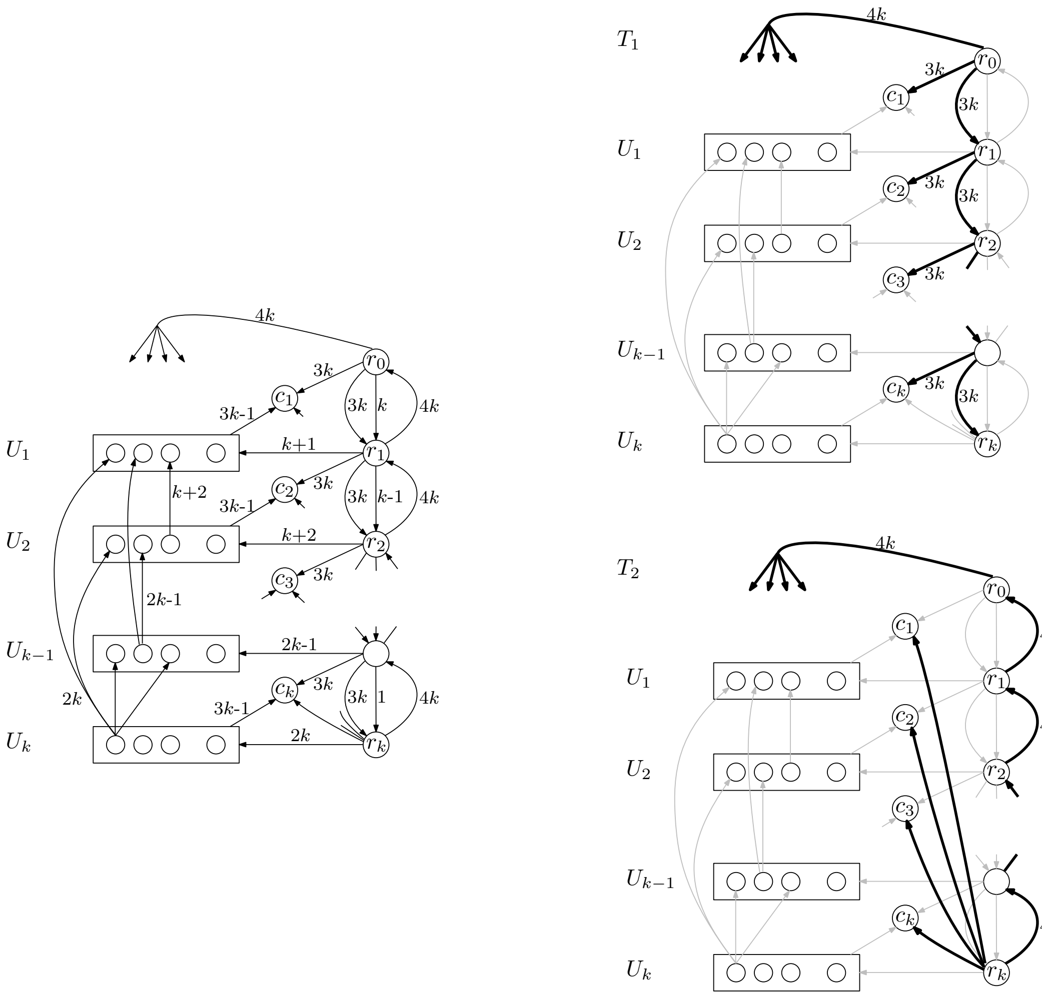

Figure 4 Main construction for the W[1]-hardness proof. Left: the temporal graph D. Top-right: the initial temporal arborescence T_1. Bottom-right: the target temporal arborescence T_2. Note that the timestamps 0 of arcs (r_k, c_i) are not shown.

As for the arc set $A(D)$, add the following arc sets:

- *R-R arcs:* for each $i \in \{0, 1, \ldots, k-1\}$, add the arc $e_i = (r_i, r_{i+1}, 3k)$; the arc $e'_i = (r_i, r_{i+1}, k-i)$; and the arc $f_i = (r_{i+1}, r_i, 4k)$.
- *r_0-U arcs:* for each $u \in V(G)$, add the arc $(r_0, u', 4k)$.
- *r_i-U_i arcs:* for each color class $i \in [k]$ and each $u \in V_i$, add the arc $(r_i, u', k+i)$. Note that $i > 0$, hence r_0 is not concerned here.
- *U_i-U_j arcs:* for each $i, j \in [k]$ with $j < i$ and each $uv \in E_{i,j}$ with $u \in V_i$ and $v \in V_j$, add an arc $(u', v', k+i)$. That is, each vertex u' has an outgoing arc to v' whenever v is a neighbor of u in a "lower" color class. In terms of Figure 4, this means that all arcs between the U_i sets go upwards. The tail of the arc determines its timestamp.
- *R-C arcs:* for each color class $i \in [k]$, add the arcs $(r_{i-1}, c_i, 3k)$ and $(r_k, c_i, 0)$.
- *U_i-c_i arcs:* for each color class $i \in [k]$, and each $u \in V_i$, add the arc $(u', c_i, 3k-1)$.

The arcs of the initial temporal arborescence T_1 consist of: the *R-R* arcs e_i for $i \in \{0, 1, \ldots, k-1\}$, so that there is a path of arcs at time $3k$ from r_0 to r_k; the r_0-U arcs $(r_0, u', 4k)$ for $u \in V$; the *R-C* arcs $(r_{i-1}, c_i, 3k)$ for $i \in [k]$.

The arcs of the target temporal arborescence T_2 consist of the same arc set as T_1, except that: no e_i arc is in T_2, and instead each R-R arc f_i is in T_2; no $(r_{i-1}, c_i, 3k)$ arc is present, and instead each R-C arc $(r_k, c_i, 0)$ is in T_2. It is not difficult to verify that T_1 and T_2 are temporal arborescences (hence time-respecting).

The intuition behind this construction is as follows. To transform T_1 into T_2, one must first re-root from r_0 to r_1, then to r_2, and so on until r_k is the root. If we re-root from r_0 to r_1, we need to insert the arc $(r_1, r_0, 4k)$. This cannot be done in the very first arc flip though, because the arc $(r_0, c_1, 3k)$ in the R-C group would violate temporality. So any solution must first create an alternate path from r_0 to c_1 before the first re-rooting. One can show that the only way to achieve this is to choose some $u'_1 \in U_1$ and create the path $r_0 \to r_1 \to u'_1 \to c_1$, using arcs at times $k, k+1, 3k-1$. Once this is done, we can safely re-root to r_1.

Next, we must re-root to r_2. As before, we cannot insert $(r_2, r_1, 4k)$ because of $(r_1, c_2, 3k)$. So we must create an alternate path $r_1 \to r_2 \to u'_2 \to c_2$ for some $u'_2 \in U_2$. However this time, the arc $(r_1, u'_1, k+1)$ from the previous step is also an issue and we must also have an alternate path from r_1 to u'_1. The key idea is that the most efficient way to do this is, after choosing u'_2, to apply a flip that removes $(r_1, u'_1, k+1)$ and inserts $(u'_2, u'_1, k+2)$. This arc exists only if $u_2u_1 \in E(G)$, forcing us to choose u'_2, u'_1 that form a clique of size 2.

The same idea applies for every $i \in [k]$. Before re-rooting from r_{i-1} to r_i, we must find an alternate path $r_{i-1} \to r_i \to u'_i \to c_i$ by choosing some $u'_i \in U_i$. At this point, there are $u'_1, \ldots, u'_{i-1}$ that are used as in-neighbors of $c_1, \ldots, c_{i-1}$. The most efficient setup is to choose u'_i that allows inserting the $(u'_i, u'_j, k+i)$ arcs for all those $j < i$, requiring all corresponding u_j's to be neighbors of u_i in G. In other words, there are k phases to apply, one for each re-rooting to each r_i, and at each phase i we must choose a u_i (and corresponding u'_i) that is a neighbor of all the previously chosen u_j's, thereby forming a clique. The specific arc timestamps in the construction are chosen to enforce this behavior.

We will show that G contains a multicolored clique if and only if T_1 can be transformed into T_2 using at most $p = 2k + \sum_{i=1}^{k}(i+3)$ arc flips. In essence, each term in the summation represents the arc flips needed to re-root from r_{i-1} to r_i, and the $2k$ term is there for a cleanup phase after having re-rooted to r_k. Note that since p is a function of k only, this shows $W[1]$-hardness in parameter p being the number of required arc flips. Also note that in fact, all timestamps assigned to arcs are a function of k, so the problem is W[1]-hard in parameter $p + t$, where $t = \max_{(u,v,t') \in A(D)} t'$.

▶ **Theorem 6.** *The* Temporal Arborescence *problem is W[1]-hard when parameterized by the number of arc flips plus the maximum timestamp.*

Proof. First note that the construction of D from G can be carried out in polynomial time. As mentioned above, we show that G contains a multicolored clique if and only if T_1 can be transformed into T_2 using at most $p = 2k + \sum_{i=1}^{k}(i+3)$ arc flips.
($\Rightarrow$) Suppose that G has a multicolored clique $K = \{u_1, \ldots, u_k\}$, where for each $i \in [k]$ the vertex u_i belongs to color class V_i. As shown in Figure 5, starting from T_1, one can re-root from r_0 to r_1 (each step can easily be checked to maintain a temporal arborescence, hence time-respecting):

- Remove $e_0 = (r_0, r_1, 3k)$ and insert $e'_0 = (r_0, r_1, k)$, so that r_0 reaches r_1 with the arc at time k instead of the arc at time $3k$;
- Remove $(r_0, u'_1, 4k)$ and insert $(r_1, u'_1, k+1)$, which is now possible. Then remove $(r_0, c_1, 3k)$ and insert $(u'_1, c_1, 3k-1)$;
- Remove e'_0 and insert $(r_1, r_0, 4k)$, thereby re-rooting to r_1.

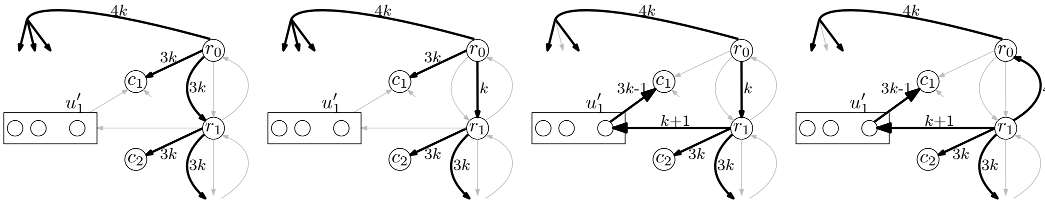

Figure 5 A sequence of arc flips to re-root from r_0 to r_1.

Note that this requires $4 = 1 + 3$ flips. Now let $i \geq 2$ and let us see how to re-root from r_{i-1} to r_i (illustrated in Figure 6). Assume that we have reached a temporal arborescence such that: r_{i-1} is the root; $(r_{i-1}, u'_{i-1}, k + i - 1)$ is active; $(u'_{i-1}, u'_j, k + i - 1)$ is active for each $j < i - 1$; $(u'_j, c_j, 3k - 1)$ is active for each $j \leq i - 1$. Also assume that r_{i-1} reaches r_0 using the f_j upwards arcs at time $4k$, and that r_0 uses $4k$ arcs to reach all the v'_j other than $u'_1, \ldots, u'_{i-1}$. Note that all these conditions hold for $i = 2$ after applying the re-rooting from r_0 to r_1. We show how to re-root from r_{i-1} to r_i, such that the same properties hold but with r_i as the root. To achieve this:

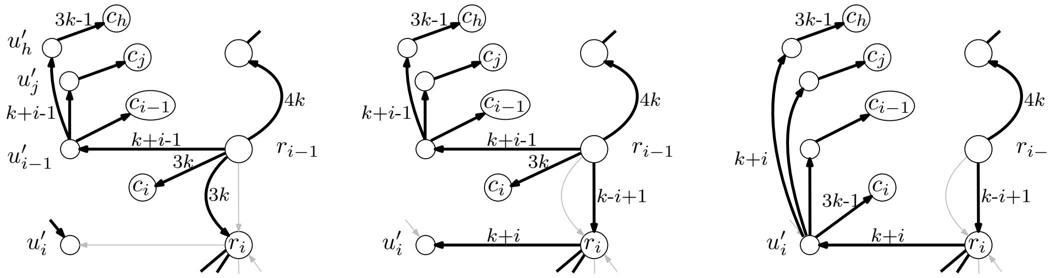

Figure 6 A sequence of arc flips to re-root from r_{i-1} to r_i. Here, we assume $h < j < i - 1$. The middle state is obtained by two arc flips that insert e'_{i-1} and (r_i, u'_i). The rightmost state is obtained by making u'_i the in-neighbor of every u'_j, $j < i$. The last step is not shown and consists in flipping e'_{i-1} to f_{i-1} to re-root to r_i.

- Remove $e_{i-1} = (r_{i-1}, r_i, 3k)$ and add $e'_{i-1} = (r_{i-1}, r_i, k - (i - 1))$, so that r_{i-1} now reaches r_i with an arc at timestamp $k - i + 1$. This preserves temporality since this is akin to lowering the timestamp for the arc from r_{i-1} to r_i, which is an outcoming arc from the root r_{i-1}.
- Remove $(r_0, u'_i, 4k)$ and add $(r_i, u'_i, k + i)$. This preserves temporality since the new path from r_{i-1} to u'_i uses arcs at respective times $k - i + 1$ and $k + i$.
- Remove $(r_{i-1}, c_i, 3k)$ and add $(u'_i, c_i, 3k - 1)$, which is correct since the latter has time $3k - 1 > k + i$.
- For each $j < i-1$, remove the incoming arc $(u'_{i-1}, u'_j, k + i - 1)$ of u'_j and add $(u'_i, u'_j, k + i)$ (which exists because $u_i u_j \in E(G)$).
  This is temporarily correct since u'_i is currently reachable with arcs of timestamp at most $k + i$, each arc from u'_i to u'_j has timestamp $k + i$, and each arc from u'_j to c_j has timestamp $3k - 1 > k + i$.
- Remove $(r_{i-1}, u'_{i-1}, k + i - 1)$ and add $(u'_i, u'_{i-1}, k + i)$, which preserves temporality as in the previous step.
- Finally, re-root ro r_i by removing e'_{i-1} and adding $f_{i-1} = (r_i, r_{i-1}, 4k)$. This preserves temporality because, at this point we have the situation from Figure 6 on the right. The only vertices that r_{i-1} was reaching without going through r_i were $r_{i-2}, \ldots, r_0$ and u' vertices using $(r_0, u', 4k)$ arcs, and all the underlying paths consisted of arcs at time $4k$.

Observe that all the assumptions made before handling step i are true for the next step. Also note that to re-root from r_{i-1} to r_i, the above requires $3 + (i - 2) + 2 = i + 3$ flips.

Once we reach a point where r_k is the root, we can: replace every $(u'_j, c_j, 3k - 1)$ with $(r_k, c_j, 0)$ for $j \in [k]$ (k arc flips); remove all the $(u'_k, u'_j, 2k)$ arcs and insert $(r_0, u'_j, 4k)$ for $j < k$ ($k - 1$ arc flips); replace $(r_k, u'_k, 2k)$ with $(r_0, u'_k, 4k)$ (1 arc flip). This last step adds $2k$ arc flips.

Overall, we have reached T_2 using $\sum_{i=1}^{k}(i + 3) + 2k = p$ arc flips.

($\Leftarrow$) Suppose that T_1 can be transformed into T_2 using at most p flips. It can be shown that this implies that $u_1, \ldots, u_k$ form a multicolored clique of G. The proof is omitted for space reasons and can be found in the full version – the main idea is that the steps described in the forward direction are essentially forced to achieve p flips. Since Multicolored Clique is W[1]-hard (for parameter k), the parameterized reduction we have described implies that Temporal Arborescence Reconfiguration is W[1]-hard for parameters number of arc flips plus maxmum timestamp. ◀

6 Conclusion

We have analyzed the complexity Temporal Arborescence Reconfiguration, proving that it is NP-hard for two timestamps, it is inapproximable within factor $b \ln |V(D)|$, for any $0 < b < 1$, if the two temporal arborescences differ only for two arc pairs, and it is W[1]-hard when parameterized by the number of arc flips needed to transform one arborescence into the other plus maximum timestamp.

A natural future direction is to further study the approximation complexity of the problem, in particular if it is possible to achieve a $c \ln |V(D)|$ approximation factor, for some constant $c \geq 1$. A second future direction is to further investigate the problem when the input temporal digraph has specific properties (for example bounded treewidth or bounded degree), both in the approximation and parameterized complexity framework.

References

1. Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Christoforos L. Raptopoulos. The temporal explorer who returns to the base. *J. Comput. Syst. Sci.*, 120:179–193, 2021. doi:10.1016/j.jcss.2021.04.001.
2. Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177, 2006. doi:10.1145/1150334.1150336.
3. Marco Amoruso, Daniele Anello, Vincenzo Auletta, Raffaele Cerulli, Diodato Ferraioli, and Andrea Raiconi. Contrasting the spread of misinformation in online social networks. *J. Artif. Intell. Res.*, 69:847–879, 2020. doi:10.1613/JAIR.1.11509.
4. Daniela Bubboloni, Costanza Catalano, Andrea Marino, and Ana Silva. On computing optimal temporal branchings. In Henning Fernau and Klaus Jansen, editors, *Fundamentals of Computation Theory - 24th International Symposium, FCT 2023, Trier, Germany, September 18-21, 2023, Proceedings*, volume 14292 of *Lecture Notes in Computer Science*, pages 103–117. Springer, 2023. doi:10.1007/978-3-031-43587-4_8.
5. Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014. doi:10.1145/2591796.2591884.
6. Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009. doi:10.1016/J.TCS.2008.09.065.

7 Guillaume Fertin, Julien Fradin, and Géraldine Jean. The maximum colorful arborescence problem: How (computationally) hard can it be? *Theoretical Computer Science*, 852:104–120, 2021.

8 Ziyun Guang, Matthew Smith-Erb, and Layla Oesper. A weighted distance-based approach for deriving consensus tumor evolutionary trees. *Journal Title Here*, pages 1–9, 2023.

9 Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):234, 2015.

10 Petter Holme and Jari Saramäki. A map of approaches to temporal networks. In *Temporal Network Theory*, pages 1–24. Springer, 2019.

11 Silu Huang, Ada Wai-Chee Fu, and Ruifeng Liu. Minimum spanning trees in temporal graphs. In Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives, editors, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 419–430. ACM, 2015. `doi:10.1145/2723372.2723717`.

12 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011. `doi:10.1016/J.TCS.2010.12.005`.

13 Takehiro Ito, Yuni Iwamasa, Naoyuki Kamiyama, Yasuaki Kobayashi, Yusuke Kobayashi, Shun-ichi Maezawa, and Akira Suzuki. Reconfiguration of time-respecting arborescences. In Pat Morin and Subhash Suri, editors, *Algorithms and Data Structures - 18th International Symposium, WADS 2023, Montreal, QC, Canada, July 31 - August 2, 2023, Proceedings*, volume 14079 of *Lecture Notes in Computer Science*, pages 521–532. Springer, 2023. `doi:10.1007/978-3-031-38906-1_34`.

14 Takehiro Ito, Yuni Iwamasa, Yasuaki Kobayashi, Yu Nakahata, Yota Otachi, and Kunihiro Wasa. Reconfiguring (non-spanning) arborescences. *Theor. Comput. Sci.*, 943:131–141, 2023. `doi:10.1016/J.TCS.2022.12.007`.

15 Naoyuki Kamiyama and Yasushi Kawase. On packing arborescences in temporal networks. *Inf. Process. Lett.*, 115(2):321–325, 2015. `doi:10.1016/J.IPL.2014.10.005`.

16 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. `doi:10.1007/978-1-4684-2001-2_9`.

17 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002. `doi:10.1006/jcss.2002.1829`.

18 Ning Li and Jennifer C. Hou. Topology control in heterogeneous wireless networks: Problems and solutions. In *Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004*. IEEE, 2004. `doi:10.1109/INFCOM.2004.1354497`.

19 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Math.*, 12(4):239–280, 2016. `doi:10.1080/15427951.2016.1177801`.

20 Dana Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. *Theory Comput.*, 11:221–235, 2015. `doi:10.4086/TOC.2015.V011A007`.

21 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. `doi:10.3390/A11040052`.