Theoretical Computer Science 1055 (2025) 115502

Contents lists available at ScienceDirect

Theoretical Computer Science

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

On the complexity of temporal arborescence reconfiguration ☆

Riccardo Dondi [a,*], Manuel Lafond [b]

[a] Università degli Studi di Bergamo, Italy
[b] Université de Sherbrooke, Canada

ARTICLE INFO

Section Editor: Paul G. Spirakis

Handling Editor: Arnaud Casteigts

Keywords:
Arborescence
Temporal graphs
Graph algorithms
Parameterized complexity
Approximation complexity

ABSTRACT

In this contribution we study the 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 transforming 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 pairs of 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.

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 that spans the digraph, that is, it 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 [21], to analyze information flow in social networks [3], or in computational biology to analyze mass spectrometry data [9] and reconstruct tumor evolutionary trees [10].

Arborescences have been recently considered also in the temporal graph setting [18,14,4,16], where they can model urban mobility or information dissemination in social networks. The temporal graph model extends the classical graph model by incorporating temporal information on the relations between elements (i.e., vertices). Temporal graphs have been studied to model the dynamic evolution of network relations (edges or arcs), that are observed only at certain time instants [20,11,22,12,1,13,5]. 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 and such that: (1) T contains every vertex of D and (2) every path in T is *time-respecting*, that is the timestamps on the arcs of every path are non-decreasing.

☆ A preliminary version of this paper appeared in [7].

* Corresponding author.
E-mail addresses: riccardo.dondi@unibg.it (R. Dondi), manuel.lafond@usherbrooke.ca (M. Lafond).

https://doi.org/10.1016/j.tcs.2025.115502
Received 16 October 2024; Received in revised form 19 July 2025; Accepted 30 July 2025

Available online 6 August 2025

In this contribution, we consider temporal arborescences through the lens of *combinatorial reconfiguration* [15,24]. 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*. An arc flip exchanges two arcs, one arc is deleted from the temporal arborescence and one arc (of D) is inserted in the temporal arborescence, with the constraint that it produces another temporal arborescence. Therefore a reconfiguration computes 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 [16]. 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 arc flips. The problem is known to be NP-hard when the temporal graph is defined over 3 timestamps or more [16], and polynomial-time solvable when the 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 [17]. The case of 2 timestamps remained open [16].

An interesting property shown in [16], 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 [16] and always admits a positive answer in static directed graphs [17] and when the two arborescences have the same root [16].

Our Results. In this paper we further analyze the computational, approximation and parameterized complexity of Temporal Arborescence Reconfiguration, considering additional restrictions on the instance. Note that we consider the temporal graph model of [16], 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 [16] 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. This hardness holds even if the maximum timestamp of an arc is 4. Note that the hardness result easily extends to the case where the number of different arcs between the two temporal arborescences is larger than 2. We also observe that if the two temporal arborescences differ for one pair of arcs, then the problem is easily solvable in polynomial time.

Finally, we consider the parameterized complexity of the Temporal Arborescence Reconfiguration 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 combination "number of arc flips plus maximum timestamp"), indicating that a fixed-parameter algorithm is unlikely. Note that in the conference version of this paper [7], our approximation and W[1]-hardness results were proved only on instances in which the digraph could have parallel arcs (that is, $(u, v, t), (u, v, t')$ with $t \neq t'$ could both be present in the digraph). We adapted our previous reductions to remove such repeated arcs, so that the hardness holds even if we require each arc to be associated with one single activation time. We conclude the paper in Section 6 with some open problems.

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. An arc in a temporal graph is thus 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. One can view the time t as the deactivation time of the arc, that is, if the arc needs to be used it needs to occur at or before time t. We may write $V(D)$ and $A(D)$ for the vertex and arc set of D, respectively. Note that even though the notation allows parallel arcs, our harness results hold even if each pair of vertices is connected by at most one arc.

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

An *arborescence* is a digraph in which there is a vertex u of in-degree 0, 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 (static) digraph. A subgraph T of D is a *spanning arborescence* 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 digraph $D = (V, A)$, we define D_U as the underlying static digraph of D, where $D_U = (V, A_U)$, with

$$A_U = \{(u, v) : (u, v, t) \in A, \text{for some } t \in \mathbb{N}\}.$$

Now, we introduce the definition of temporal arborescence of a temporal graph.

Definition 1. Given a temporal digraph D, T is a *temporal arborescence* of D if:

- T_U (the underlying static digraph of T) is an arborescence of D_U (the underlying static digraph of D)
- 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 in.

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

Note that p is encoded in unary, thus the input size is equal to the size of D plus the size of T_1 and T_2 plus p. In this case Temporal Arborescence Reconfiguration is in NP, as, given a sequence of arc flips, we can check that each arc flip produces a temporal arborescence and that the sequence starting from T_1 produces T_2.

In the optimization version of Temporal Arborescence Reconfiguration, we aim to minimize the number of arc flips of a reconfiguration from T_1 to T_2.

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 $U=\{u_1,\ldots,u_n\}$ and $S=\{S_1,\ldots,S_m\}$, an instance of Set Cover 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 Fig. 1):

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

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 Fig. 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 Fig. 1.

To give some intuition, a reconfiguration from T_1 to T_2 needs to re-root the temporal arborescence to r_3. However, it is impossible to re-root from r_1 to r_3 directly, because r_3 has too many outgoing arcs at time 1. The reconfiguration sequence must first re-root the arborescence to r_2 as an intermediate step. In order to make r_2 the root of a temporal arborescence we must first flip arcs $(r_1,u_i,1)$

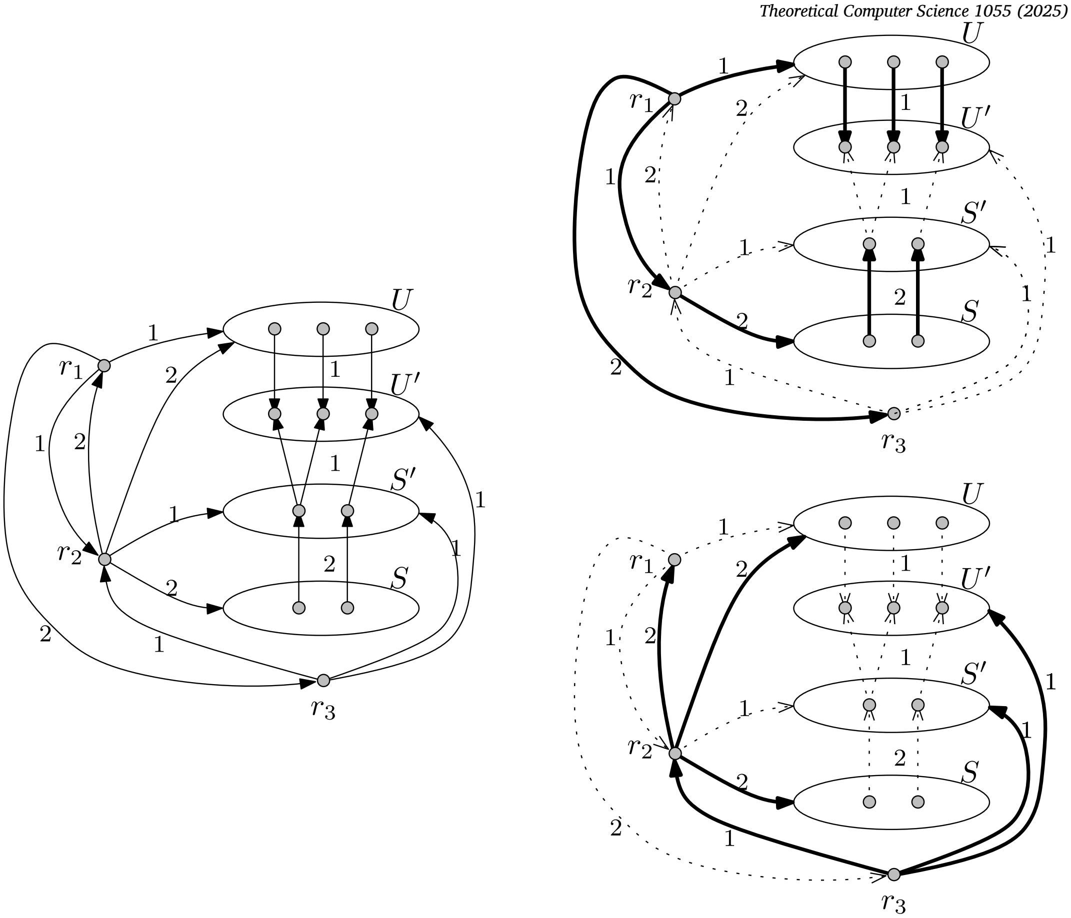

Fig. 1. Left: the temporal digraph D obtained from a set cover instance, with $U = \{u_1, u_2, u_3\}$ and $S = \{S_1, S_2\}$, $S_1 = \{u_1, u_2\}$ and $S_2 = \{u_3\}$. Arcs pointing on ellipses indicate that all possible arcs are present (e.g. r_1 has every element of U in its out-neighborhood). The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

with arcs $(r_2, u_i, 2)$, $u_i \in U$, otherwise there would be a path that is not time-respecting. These arcs flips can be applied only if arcs $(u_i, u_i', 2)$, $u_i' \in U$, are removed, hence flipped with arcs $(S_i', u_i', 2)$, for some $S_i' \in S'$, and this requires to flip arcs $(S_i, S_i', 2)$ with arcs $(r_2, S_i', 1)$, with $S_i \in S$: these latter flips encode a set cover.

Theorem 2. *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 $(r_2, S_i', 1)$ ($|S^*|$ arc flips).
   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 ($|U'|$ arc flips).
   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.
3. For each $u_j \in U$ in an arbitrary order, remove $(r_1, u_j, 1)$ and add $(r_2, u_j, 2)$ ($|U|$ arc flips).
   Each arc flip is correct since before the flips, the u_j vertices were leaves of the arborescence, and after the flip r_1 can reach the u_j vertices 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 $(r_2, r_1, 2)$ (one arc flip).
   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 $(r_3, r_2, 1)$ (one arc flip).

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 $(r_3, u'_j, 1)$ ($|U'|$ arc flips).
   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 $(r_3, S'_i, 1)$ ($|S'|$ arc flips).
   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 $l - 1 \le 3n + m + 2 + k$, 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 (recall these have timestamp 1), 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. These arcs have timestamp 2, which 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 active 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 $|S^*| \le k$.

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 $\{S'_i : S_i \in S^*\}$. Since each such arc must be inserted by a distinct flip, 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 $|S^*| \le k$.

Since Set Cover is known to be NP-hard [19], the reduction we have described implies that also Temporal Arborescence Reconfiguration for two timestamps is NP-hard. □

4. Inapproximability for two c pairs difference

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 differ for two arc pairs, that is, the number of arcs in $A(T_1) \setminus A(T_2)$ and the number of arcs 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 $S = \{S_1, \dots, S_m\}$.

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

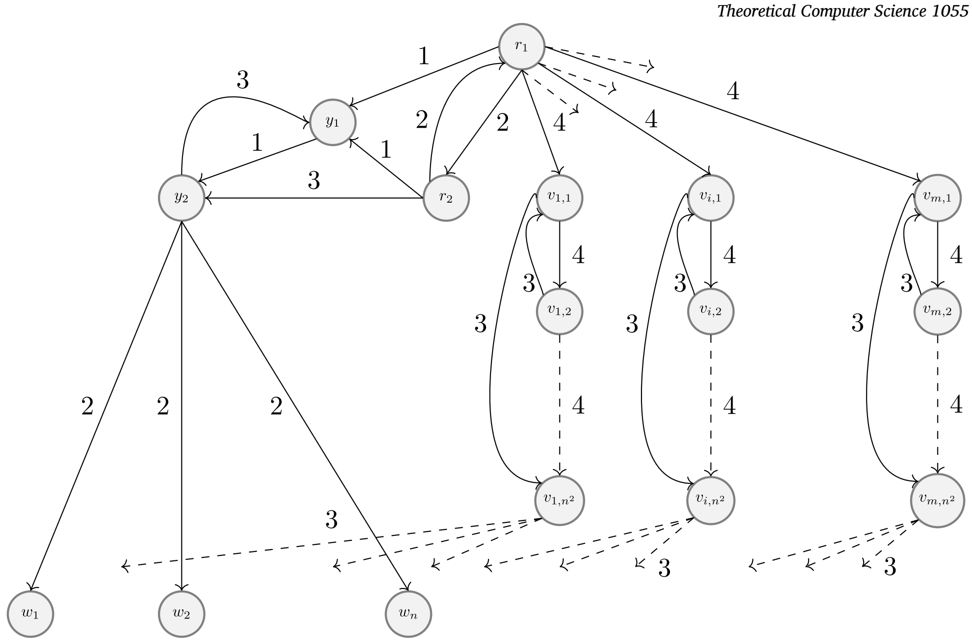

Fig. 2. The input digraph D associated with an instance of Set Cover. Each dashed arrow outgoing from $v_{i,2}$, $i \in [m]$, represent a temporal path containing vertices $v_{i,j}$, $j \in \{3, ..., n^2 - 1\}$, and not shown in the figure. The dashed arrows outgoing from v_{1,n^2}, v_{i,n^2}, v_{m,n^2} represent temporal arcs connecting these vertices with some vertices w_z, $z \in [n]$ (the precise arcs depends on the instance of Set Cover). The dashed arrows outgoing from r_1 represent temporal arcs to v_{1,n^2-1}, v_{i,n^2-1}, v_{m,n^2-1}, with timestamp 3.

Construct $(D = (V, A), T_1, T_2)$, an instance of (the optimization version of) Temporal Arborescence Reconfiguration associated with (U, S), as follows (refer to Fig. 2 for the structure of D).

$$V = \{r_1, r_2, y_1, y_2\} \cup \{v_{i,z} : S_i \in S, i \in [m], z \in [n^2]\} \cup \{w_i : i \in [n], u_i \in U\}.$$

The set of arcs A is defined as

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

where (note that some arcs appear in more than one set on purpose since this makes it easier to define the sets A_1 and A_2 that belong to T_1 and T_2, respectively):

$$\begin{aligned}
A_1 = &\{(r_1, r_2, 2)\} \cup \{(r_1, y_1, 1)\} \cup \{(y_1, y_2, 1)\} \cup \\
&\{(r_1, v_{i,1}, 4) : i \in [m]\} \cup \\
&\{(v_{i,j}, v_{i,j+1}, 4) : i \in [m], j \in [n^2 - 1]\} \cup \\
&\{(y_2, w_i, 2) : i \in [n]\} \\
A_2 = &\{(r_2, r_1, 2)\} \cup \{(r_2, y_1, 1)\} \cup \{(y_1, y_2, 1)\} \cup \\
&\{(r_1, v_{i,1}, 4) : i \in [m]\} \cup \\
&\{(v_{i,j}, v_{i,j+1}, 4) : i \in [m], j \in [n^2 - 1]\} \cup \\
&\{(y_2, w_i, 2) : i \in [n]\} \\
A_3 = &\{(r_2, y_2, 3)\} \cup \{(y_2, y_1, 3)\} \cup \\
&\{(r_1, v_{i,n^2-1}, 3) : i \in [m]\} \cup \\
&\{(v_{i,j}, v_{i,j-1}, 3) : i \in [m], j \in [2, n^2 - 1]\} \cup \{(v_1, v_{i,n^2}, 3) : i \in [m]\} \cup \\
&\{(v_{i,n^2}, w_j, 3) : u_j \in S_i, i \in [m], j \in [n]\}
\end{aligned}$$

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 $A_1 \setminus A_2 = \{(r_1, r_2, 2), (r_1, y_1, 1)\}$, while $A_2 \setminus A_1 = \{(r_2, r_1, 2), (r_2, y_1, 1)\}$.

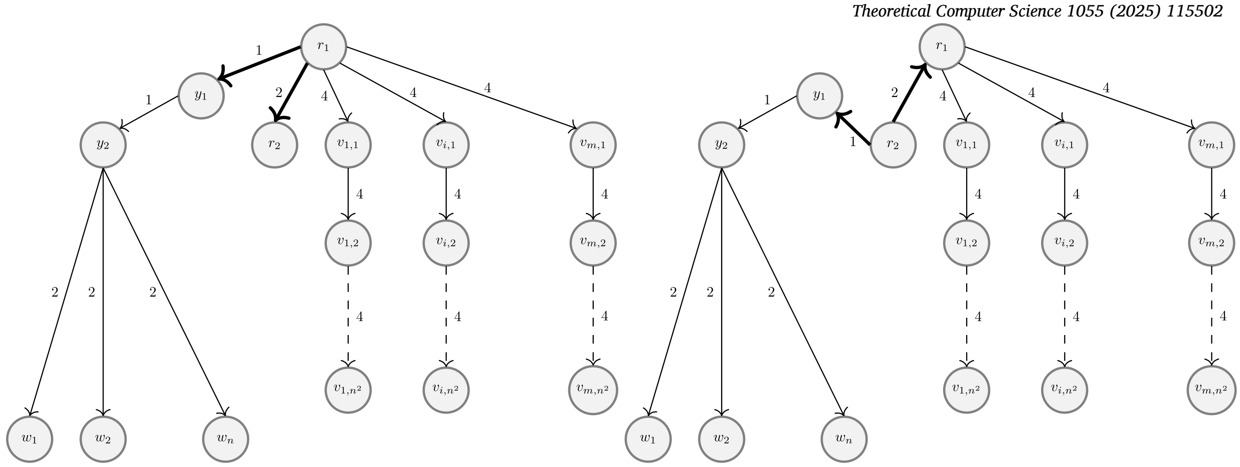

Fig. 3. Temporal arborescence T_1 (left) and T_2 (right). The four arcs in bold belong to exactly one the two temporal arborescences, the other arcs belong to both T_1 and T_2.

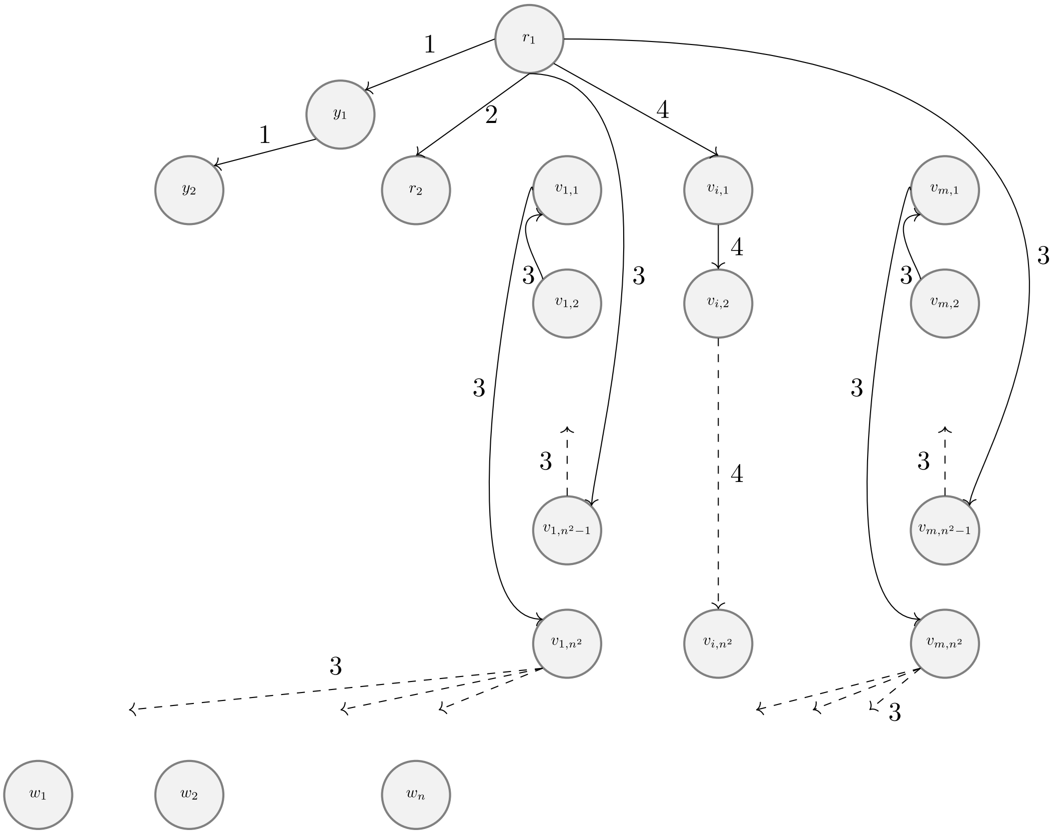

Fig. 4. The temporal arborescence obtained after the disconnection step applied to paths from r_1 to v_{1,n^2} and from r_1 to v_{m,n^2}, while the path from r_1 to v_{i,n^2} is not affected by the disconnection step. The dashed arrows represent arcs from v_{1,n^2}, v_{m,n^2} to w_i, $i \in [n]$. Note that y_2 is a leaf in the temporal arborescence and that the root can be made r_2 by the following arc flips: $(y_1, y_2, 1)$ with $(r_2, y_2, 3)$, $(r_1, y_1, 1)$ with $(y_2, y_1, 3)$ and $(r_1, r_2, 2)$ with $(r_2, r_1, 2)$.

We define a reconfiguration from T_1 to T_2 as *canonical* if it has the following properties (see Fig. 4). 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 so that it becomes a path from r_1 to v_{j,n^2} having timestamp 3. First, $(v_{j,n^2-2}, v_{j,n^2-1}, 4)$ is flipped with $(r_1, v_{j,n^2-1}, 3)$; then, for each h with $h \in [2, n^2 - 3]$,

$(v_{j,n^2-h-1}, v_{j,n^2-h}, 4)$ is flipped with $(v_{j,n^2-h}, v_{j,n^2-h-1}, 3)$. Finally, $(r_1, v_{j,1}, 4)$ is flipped with $(v_{j,2}, v_{j,1}, 3)$ and $(v_{j,n^2-1}, v_{j,n^2}, 4)$ is flipped with $(v_{j,1}, v_{j,n^2}, 3)$.

2. Each arc $(y_2, w_i, 2)$, $i \in [n]$, is flipped with some arc $(v_{j,n^2}, w_i, 3)$, where j is such that a path from r_1 to v_{j,n^2}, with all the arcs having timestamps 3 has been constructed at point 1.

Once the disconnection step is applied and each w_i, $i \in [n]$, is disconnected from y_2, a canonical reconfiguration makes the following flips so that the computed arborescence is rooted at r_2:

1. Arc $(y_1, y_2, 1)$ is flipped with arc $(r_2, y_2, 3)$
2. Arc $(r_1, y_1, 1)$ is flipped with arc $(y_2, y_1, 3)$
3. The root of the temporal arborescence is changed by flipping arcs $(r_1, r_2, 2)$ and $(r_2, r_1, 2)$
4. Arc $(y_2, y_1, 3)$ is flipped with arc $(r_2, y_1, 1)$
5. Arc $(r_2, y_2, 3)$ is flipped with arc $(y_1, y_2, 1)$

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, the arc flips applied in the disconnection step are applied in reverse order (starting from $(v_{j,1}, v_{j,n^2}, 3)$ flipped with $(v_{j,n^2-1}, v_{j,n^2}, 4)$ and ending with $(r_1, v_{n^2-1}, 3)$ flipped with $(v_{j,n^2-2}, v_{j,n^2-1}, 4)$).

Before presenting the detailed proofs, we give an intuition of the reduction. In order to re-root from r_1 to r_2 we must apply a disconnection step to some paths from r_1 to v_{j,n^2}, $j \in [m]$: these paths encode a set cover. By applying the disconnection step, we are able to flip each $(y_2, w_i, 2)$ with $(v_{j,n^2}, w_i, 3)$, $i \in [n]$ and $j \in [m]$, so that y_2 becomes a leaf of the computed temporal arborescence. This allows us, with the arc flips described at points 1–3 above, to re-root the temporal arborescence to r_2.

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

Lemma 3. *Each arborescence computed by a canonical reconfiguration from T_1 to T_2 is a temporal arborescence of D.*

Proof. Consider point 1 of the disconnection step. For some $j \in [m]$, each arc on the path from r_1 to v_{j,n^2}, associated with timestamp 4, is flipped so that it is built a path from r_1 to v_{j,n^2} having timestamp 3. First, the flip of $(v_{j,n^2-2}, v_{j,n^2-1}, 4)$ with $(r_1, v_{j,n^2-1}, 3)$ produces a temporal arborescence, since obviously v_{j,n^2-1} is reachable from r_1 and the temporal path consisting of arcs $(r_1, v_{j,n^2-1}, 3)$ and $(v_{j,n^2-1}, v_{j,n^2}, 4)$ reaches v_{j,n^2} from r_1. For the remaining vertices there is no changing with respect to T_1. Next, for each h with $h \in [2, n^2-2]$, $(v_{j,n^2-h-1}, v_{j,n^2-h}, 4)$ is flipped with $(v_{j,n^2-h}, v_{j,n^2-h-1}, 3)$. This produces a temporal path p_1 from r_1 to v_{j,n^2-h}, whose arcs have all timestamps 3, and a temporal path p_2 from r_1 to v_{j,n^2-h-1}, whose arcs have all timestamps 4. When $(r_1, v_{j,1}, 4)$ is flipped with $(v_{j,2}, v_{j,1}, 3)$, p_1 reaches $v_{j,1}$, while p_2 is removed. The flip of $(v_{j,n^2-1}, v_{j,n^2}, 4)$ with $(v_{j,1}, v_{j,n^2}, 3)$ leads to a temporal path that from r_1 reaches all the vertices $v_{j,h}$, with $h \in [n^2]$.

Let T'_1 be the arborescence produced by the application of the point 1 of the disconnection step. Consider the flip of an arc $(y_2, w_i, 2)$ of T'_1, $i \in [n]$, with an arc $(v_{j,n^2}, w_i, 3)$, $j \in [m]$. Since point 1 of the disconnection step has defined a path from r_1 to v_{j,n^2} with arcs having all timestamp equal to 3, for some $j \in [m]$, the arc flip produces a time-respecting path and the obtained arborescence is spanning since w_i is a leaf in the obtained arborescence.

Assume then that T' is produced by the disconnection step. The flip of arc $(y_1, y_2, 1)$ with arc $(r_2, y_2, 3)$ produces a time-respecting spanning arborescence T'', since y_2 is a leaf in T', and similarly the flip of $(r_1, y_1, 1)$ with $(y_2, y_1, 3)$, since y_1 is now a leaf in the temporal arborescence and the temporal path from r_1 to y_2 is time-respecting. The flip of $(r_1, r_2, 2)$ with $(r_2, r_1, 2)$ changes the root of the temporal arborescence and it is time-respecting, since all the temporal arcs outgoing from r_1 before the flip, except that to r_2, have timestamp 3. The flips of $(y_2, y_1, 3)$ with $(r_2, y_1, 1)$ produces a temporal arborescence, since y_1 is a leaf in the computed arborescence. After the application of this flip y_2 is a leaf in the computed temporal arborescence, hence the flip of $(r_2, y_2, 3)$ with $(y_1, y_2, 1)$ produces a temporal arborescence. Let T^z be the temporal arborescence computed after these flips.

Now, in T^z, each arc $(v_{j,n^2}, w_i, 3)$, for some $j \in [m]$ and $i \in [n]$, is flipped with arc $(y_2, w_j, 2)$. Since y_2 is reached from the new root r_2 with a temporal path whose arcs have all timestamp 1, each of these arc flips produces a (time-respecting) temporal arborescence. Let T^q be the temporal arborescence produced by these flips. Finally, a path from r_1 to v_{j,n^2} $j \in [m]$, with arcs having timestamp 4 is produced by applying in the reverse order the flips of the first part of the disconnection step. Each of these arc flips produces a temporal arborescence computed in the first phase of the disconnection step. After these last arc flips, the temporal arborescence computed by the canonical reconfiguration is exactly T_2, thus concluding the proof. □

We prove now the first direction of the reduction.

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 set cover of size at most k we can compute in polynomial time a reconfiguration from T_1 to T_2 consisting of at most $2kn^2 + 2n + 5$ flips.*

Proof. Consider a set cover S^* consisting of at most k set. Define in polynomial time a canonical reconfiguration from T_1 to T_2, where, in the disconnection step, for each set $S_j \in S^*$, $j \in [m]$, each arc on a temporal path from r_1 to v_{j,n^2} with timestamp 4 is flipped in order to compute a temporal path from r_1 to v_{j,n^2} whose arcs have all timestamp 3. By Lemma 3 the canonical reconfiguration computes a sequence of temporal (time-respecting) arborescences. The canonical reconfiguration requires $2|S^*|n^2 + 2n + 5$ arc flips, hence at most $2kn^2 + 2n + 5$ arc flips. Indeed, the disconnection step requires $|S^*|n^2$ arc flips for defining $|S^*|$ temporal paths with timestamp 3 from r_1 to v_{j,n^2}. Then n arc flips are required for disconnecting each w_i, $i \in [n]$, from y_2 and connecting to some v_{j,n^2}, $j \in [m]$. After the disconnection step, five arc flips are applied: $(y_1, y_2, 1)$ is flipped with $(r_2, y_2, 3)$, $(r_1, y_1, 1)$ is flipped with $(y_2, y_1, 3)$, $(r_1, r_2, 2)$ is flipped with $(r_2, r_1, 3)$, $(y_2, y_1, 3)$ is flipped with $(r_2, y_1, 1)$, and $(r_2, y_2, 3)$ is flipped with $(y_1, y_2, 1)$. Then each arc flipped in the disconnection step is flipped in reverse order, for $|S^*|n^2$ more flips, and each w_i vertex is reconnected with y_2 for n more flips. Thus the canonical reconfiguration requires at most $2kn^2 + 2n + 5$ arc flips, concluding the proof. □

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 5. *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 at most $2kn^2 + 2n + 5$ arc flips we can compute in polynomial time a solution of Set Cover on instance (S, U) of size at most 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 not time-respecting temporal path $(r_2, r_1, 2), (r_1, y_1, 1)$, if this latter arc (of T_1) belongs to the temporal arborescence. It follows that before $(r_1, r_2, 2)$ and $(r_2, r_1, 2)$ are flipped, $(r_1, y_1, 1)$ must be flipped with another arc that must be incoming to y_1 (since r_1 remains the root of the arborescence), that is with $(r_2, y_1, 1)$ or $(y_2, y_1, 3)$.

Consider the first case, arc $(r_2, y_1, 1)$. Notice that arcs $(r_1, y_1, 1)$ and $(r_2, y_1, 1)$ cannot be flipped, since this flip creates a temporal path $(r_1, r_2, 2), (r_2, y_1, 1)$, which is not time-respecting, and we have observed that $(r_1, r_2, 2)$ has not been flipped yet.

It follows that second case must hold, that is $(r_1, y_1, 1)$ must be flipped with $(y_2, y_1, 3)$. Arcs $(r_1, y_1, 1)$ and $(y_2, y_1, 3)$ cannot be flipped unless $(y_1, y_2, 1)$ is removed. This latter arc can be flipped only with $(r_2, y_2, 3)$, since this is the only other arc incoming into y_2. However, this flip can be applied (in particular the insertion of $(r_2, y_2, 3)$) only if y_2 is a leaf of the computed temporal arborescence, that is all the arcs $(y_2, w_i, 2)$, with $i \in [n]$, have been flipped. Indeed, if an arc $(y_2, w_i, 2)$, $i \in [n]$, belongs to a temporal arborescence, then by flipping $(y_1, y_2, 1)$ with $(r_2, y_2, 3)$ we have a temporal path $(r_2, y_2, 3), (y_2, w_i, 2)$, which is not time-respecting. It follows that, before $(y_1, y_2, 1)$ is flipped with $(r_2, y_2, 3)$ each vertex w_i, $i \in [n]$, must first be disconnected from y_2. By construction the only incoming arcs to a vertex w_i, $i \in [n]$, other than $(y_2, w_i, 2)$, are $(v_{j,n^2}, w_i, 3)$, for some $j \in [m]$, hence each vertex w_i must first be disconnected from y_2 by flipping an arc $(y_2, w_i, 2)$ 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 temporal path from r_1 to v_{j,n^2}, which have timestamp 4 in T_1, is flipped so that a temporal from r_1 to v_{j,n^2}, whose arcs have all timestamp 3, is defined.

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 at most $2kn^2 + 2n + 5$ 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 $2(k + 1)n^2 + 2n > 2kn^2 + 2n + 5$, since $n \geq 2$ (otherwise the problem is trivial). Hence S^* contains at most k sets, thus completing the proof. □

Based on Lemma 4, on Lemma 5, on the fact that the digraph D contains $O(n^2 m)$ vertices and on the hardness of approximation of Set Cover [2,6,23], we can prove the following result.

Theorem 6. *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.*

Proof. We prove that we have designed an approximation preserving reduction. Denote the value of an optimal solution of Temporal Arborescence Reconfiguration on instance (D, T_1, T_2) (Set Cover on instance (S, U), respectively) by $OPT(AR)$ ($OPT(SC)$, respectively); denote the value of an approximate solution of Temporal Arborescence Reconfiguration on instance (D, T_1, T_2) (Set Cover on instance (S, U), respectively) by $APX(AR)$ ($APX(SC)$, respectively). Next, consider the approximation factor of Temporal Arborescence Reconfiguration, that is

$$\frac{APX(AR)}{OPT(AR)}.$$

We have that, by Lemma 5, given an approximated solution of Temporal Arborescence Reconfiguration of value $APX(AR)$, we can compute in polynomial time an approximation solution of Set Cover of value $APX(SC)$ such that $APX(AR) \geq 2APX(SC)n^2 + 2n + 5$. Thus

$$\frac{APX(AR)}{OPT(AR)} \geq \frac{2APX(SC)n^2 + 2n + 5}{OPT(AR)}.$$

By Lemma 4, it follows that $OPT(AR) \leq 2OPT(SC)n^2 + 2n + 5$, thus

$$\frac{APX(AR)}{OPT(AR)} \geq \frac{2APX(SC)n^2 + 2n + 5}{OPT(AR)} \geq \frac{2APX(SC)n^2 + 2n + 5}{2OPT(SC)n^2 + 2n + 5} > \frac{2APX(SC)n^2}{2OPT(SC)n^2 + 2n + 5} \geq$$
$$\frac{2APX(SC)n^2}{2OPT(SC)n^2} \cdot \frac{2OPT(SC)n^2}{2OPT(SC)n^2 + 2n + 5} \geq (1 - o(1))\frac{APX(SC)n^2}{OPT(SC)n^2}$$

since, assuming that $OPT(SC)$ is large enough (it does not consist of a constant number of sets, otherwise Set Cover is solvable in polynomial time), it holds that

$$\frac{2OPT(SC)n^2}{2OPT(SC)n^2 + 2n + 5} \geq 1 - o(1).$$

Set Cover is not approximable within factor $c \ln n$, for any constant $0 < c < 1$, unless P = NP [2,6], thus

$$\frac{APX(AR)}{OPT(AR)} \geq (1 - o(1))\frac{APX(SC)n^2}{OPT(SC)n^2} > c' \ln n,$$

for any constant $0 < c' < 1$. Finally, observe that by construction $|V(D)| \leq mn^2 + n + 4$ and that the hardness of Set Cover holds also if $m \leq poly(n)$ [23], thus $|V(D)| \leq poly(n)$ and hence for any constant $0 < b < 1$ it holds that

$$\frac{APX(AR)}{OPT(AR)} > c \ln n \geq b \ln |V(D)|.$$

This concludes the proof. □

Note that the hardness result easily extends to the case where the 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_1 and of the subtrees rooted at $v_{i,j}$. Each copy has to be reconfigured independently, thus the inapproximation ratio is the same as in our result.

One Pair Arcs Difference

We have shown that Temporal Arborescence Reconfiguration is hard (also to approximate) when $T_1 = (V, A(T_1))$ and $T_2 = (V, A(T_2))$ differ for two pairs of arcs. On the other hand when T_1 and T_2 differ for one pair, we have the following result.

Remark 7. Temporal Arborescence Reconfiguration is solvable in polynomial-time when the two input temporal arborescences differ for one pair of arcs.

Since T_1 and T_2 have differ for one pair, $A(T_1) \setminus A(T_2)$ 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, by flipping a_1 with a_2 in T_1, hence by removing a_1 and inserting a_2, we obtain T_2, which is a temporal arborescence. It follows that the arc flip of a_1 with a_2 can always be applied and the solution is optimal, since T_1 is different from T_2 and at least one arc flip is required to compute a reconfiguration from T_1 to T_2.

5. W[1]-hardness for the number of arc flips

In all the above reductions (Section 3 and Section 4) and also the reduction in [16], 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 [8].

Let $G = (V, E)$ be an instance a Multicolored Clique, with vertices partitioned into color classes $V_1, \dots, V_k$. For $i, j \in [k]$, we will denote $E_{i,j} = \{uv \in E : u \in V_i, v \in V_j\}$. Let us construct an instance (D, T_1, T_2, p) of Temporal Arborescence Reconfiguration, where D is a temporal digraph, T_1, T_2 are arborescences, and p is the number of flips.

Let us first construct D, which is shown in Fig. 5 (we provide the main intuitions after the description of the construction). We define the vertex set of D as $V(D) = R \cup \tilde{R} \cup C \cup U$, where

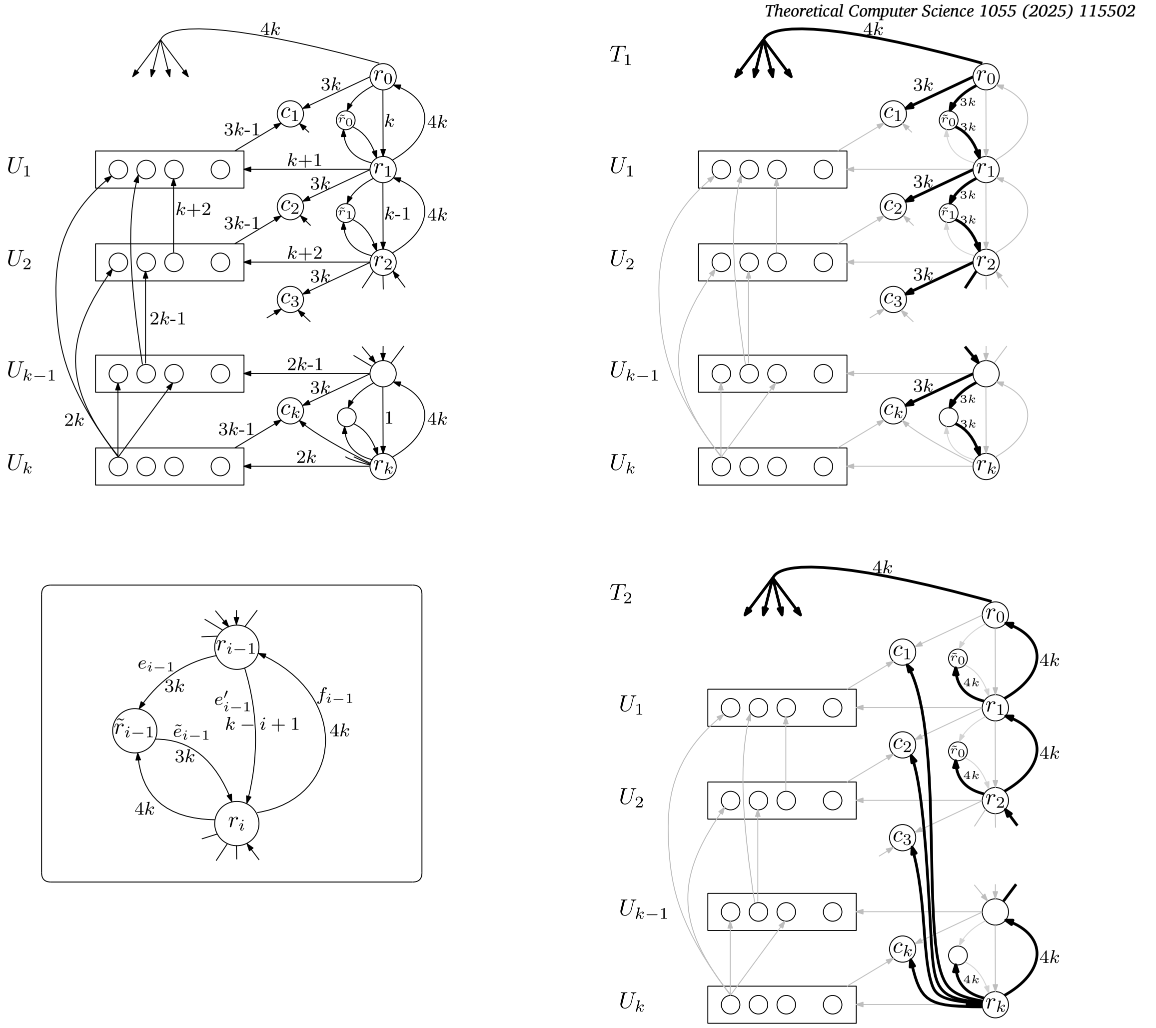

Fig. 5. Main construction for the W[1]-hardness proof. Top-left: the temporal graph D. Bottom-left: a zoom-in on how vertices $r_{i-1}, \tilde{r}_{i-1}, r_i$ are connected with the arcs $e_{i-1}, \tilde{e}_{i-1}, e'_{i-1}, f_{i-1}$. Top-right: the initial temporal arborescence T_1. Bottom-right: the target temporal arborescence T_2. Note that the timestamps 0 of arcs (r_k, c_i) are not shown.

$$R = \{r_0, r_1, \dots, r_k\}$$

$$\tilde{R} = \{\tilde{r}_0, \tilde{r}_1, \dots, \tilde{r}_{k-1}\}$$

$$C = \{c_1, c_2, \dots, c_k\}$$

$$U = \{u' : u \in V(G)\}.$$

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

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

- *R-$\tilde{R}$ arcs:* for each $i \in \{0, 1, \dots, k-1\}$, add the arc $e_i = (r_i, \tilde{r}_i, 3k)$; the arc $\tilde{e}_i = (\tilde{r}_i, r_{i+1}, 3k)$; the arc $e'_i = (r_i, r_{i+1}, k-i)$; and the arc $f_i = (r_{i+1}, r_i, 4k)$. Also add the arc $(r_{i+1}, \tilde{r}_i, 4k)$.
- *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 Fig. 5, 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 $(u', c_i, 3k-1)$.

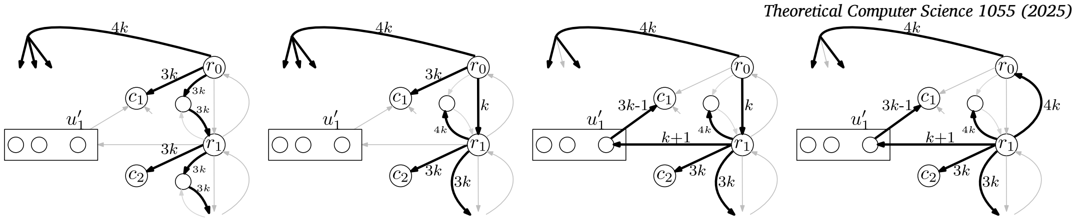

Fig. 6. A sequence of arc flips to re-root from r_0 to r_1.

The arcs of the initial temporal arborescence T_1 consist of: the R-$\tilde{R}$ arcs $e_i, \tilde{e}_i$ 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 $e_i, \tilde{e}_i$ arc is in T_2, and instead each R-$\tilde{R}$ arc f_i is in T_2 for $i \in \{0, 1, \ldots, k-1\}$ and each arc $(r_i, \tilde{r_{i-1}}, 4k)$ is in T_2 for $i \in [k]$; 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 first choose some $u'_1 \in U_1$ and perform some flips to 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 $u_2 u_1 \in E(G)$, 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 $p = 2k + \sum_{i=1}^{k}(i+4)$ 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. As previously claimed, all timestamps assigned to arcs are a function of k, so the problem is in fact W[1]-hard in parameter $p + t$, where $t = \max_{(u,v,t') \in A(D)} t'$.

Theorem 8. *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 $p = 2k + \sum_{i=1}^{k}(i+4)$ 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 (from now on, the u_i and u'_i vertices with subscripts refer to vertices of the clique). As shown in Fig. 6, 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 $\tilde{e}_0 = (\tilde{r}_0, r_1, 3k)$ 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 arcs at time $3k$. Then remove $(r_0, \tilde{r}_0, 3k)$ and insert $(r_1, \tilde{r}_0, 4k)$; This results in the second arborescence shown in Fig. 6.
- 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.

Note that this requires $5 = 1 + 4$ flips. Now let $i \geq 2$ and let us see how to re-root from r_{i-1} to r_i (illustrated in Fig. 7). 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:

- Remove $\tilde{e}_{i-1} = (\tilde{r}_{i-1}, r_i, 3k)$ 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 any path previously going through r_i reached it at time $3k$, and now r_i is reached at a lower time $k-i+1$.

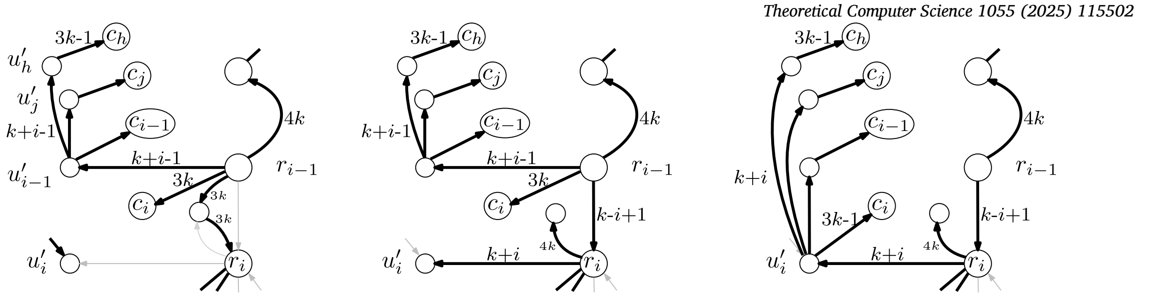

Fig. 7. A sequence of arc flips to re-root from r_{i-1} to r_i. Here, we assume $h < j < i-1$. The middle state is obtained by two arc flips that insert e'_{i-1} and (r_i, u'_i). The rightmost state is obtained by making u'_i the in-neighbor of every u'_j, $j < i$. The last step is not shown and consists in flipping e'_{i-1} to f_{i-1} to re-root to r_i.

Then, remove $e_{i-1} = (r_{i-1}, \tilde{r}_{i-1}, 3k)$ and add $(r_i, \tilde{r}_{i-1}, 4k)$.

- 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 Fig. 7 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 $4 + (i-2) + 2 = i + 4$ 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 $\sum_{i=1}^{k}(i+4) + 2k = p$ arc flips.

($\Leftarrow$) Suppose that T_1 can be transformed into T_2 using at most p flips. Let $\mathcal{R}$ be the sequence of arborescences encountered in this transformation, so that the first element of $\mathcal{R}$ is T_1 and the last is T_2.

We first establish that we must move the roots from $r_0, r_1, \ldots, r_k$, and then gather several conditions that must occur during each re-rooting.

Fact 4. *For any $i \in [k]$, $\mathcal{R}$ eventually reaches a temporal arborescence in which r_{i-1} is the root, such that after the next flip, r_i becomes the root.*

Proof. Assume that some r_{i-1} is the root of a temporal arborescence T at some point in $\mathcal{R}$, and that the next flip yields an arborescence T' whose root is different. We argue that this new root is either r_{i-2} (if it exists) or r_i. Note that since r_{i-1} has an in-neighbor in T' but not in T, the arc inserted by the flip must be an incoming arc of r_{i-1}. There are three such possible arcs:

1. $\tilde{e}_{i-2} = (\tilde{r}_{i-2}, r_{i-1}, 3k)$ is inserted to obtain T'. In that case, consider how r_{i-1} reaches $\tilde{r}_{i-2}$ in T. The path $r_{i-1} - r_{i-2} - \tilde{r}_{i-2}$ is not time-respecting, and the only other option is to use the arc $(r_{i-1}, \tilde{r}_{i-2}, 4k)$. In that case, the flip that yields T' must remove that arc. Since $\tilde{r}_{i-2}$ loses its in-neighbor in T', it must now be the root. However, there is no time-respecting path from $\tilde{r}_{i-2}$ to c_{i-2}. Indeed, if we assume that such a path exists, it must start with the arc $\tilde{e}_{i-2} = (\tilde{r}_{i-2}, r_{i-1}, 3k)$ at time $3k$. Then the successor of r_{i-1} cannot be r_{i-2}, since this requires taking the arc f_{i-2} at time $4k$, whereas all the in-neighbors of c_{i-2} have time $3k$ or less. Since there are no other ways to get to r_{i-2}, the in-neighbor of c_{i-2} cannot be r_{i-2}. It must thus be some $u \in U_{i-2}$ or r_k, which is again impossible since the arcs from U_{i-2} to c_{i-2} have time $3k-1 < 3k$ and the $r_k - c_{i-2}$ arc has time 0. It follows that this case cannot occur.
2. $e'_{i-2} = (r_{i-2}, r_{i-1}, k-i+2)$ is inserted. In this scenario, note that $f_{i-2} = (r_{i-1}, r_{i-2}, 4k)$ must be present in T for r_{i-1} to reach r_{i-2}. The flip that yields T' must remove that arc, in which case r_{i-2} loses its in-neighbor and becomes the new root.
3. $f_{i-1} = (r_i, r_{i-1}, 4k)$ is inserted. In this case, in T the path from r_{i-1} to r_i either uses $e_{i-1}, \tilde{e}_{i-1}$, or it uses e'_{i-1}. In the first case, to obtain T', we cannot remove e_{i-1}, because $\tilde{r}_{i-1}$ would become the root but be unable to reach c_{i-1} (similarly as in case 1 above). So we have to remove $\tilde{e}_{i-1}$, in which case r_i loses its in-neighbor and becomes the root. If the r_{i-1} to r_i path of T uses e'_{i-1} instead, it must be removed and again r_i becomes the root.

We deduce that over the sequence of arc flips, only vertices from R can be roots, and the subscript of the root can only increase of decrease by 1. Since the root of T_1 is r_0 and the root of T_2 is r_k, we must go through each of $r_0, r_1, \dots, r_k$ as the root at least once. Moreover, if T^+ is the first arborescence encountered in $\mathcal{R}$ whose root is r_i, the fact that root subscripts can only increase or decrease by 1 implies that the root prior to T^+ could only have been r_{i-1}, which proves the statement. □

In what follows, when we say that an arc e is *active before and after a re-rooting* from r_{i-1} to r_i, we mean that we have a temporal arborescence T with r_{i-1} as the root, perform an arc flip to obtain T' with a new root r_i, and that e is active in both T and T' (with no guarantees before T nor after T'). Analogously, when we say that a path is active before and after re-rooting, we mean that all the arcs of the path are.

Fact 5. *Let $i \in [k-1]$. When re-rooting from r_{i-1} to r_i, the arcs e_i and $\tilde{e}_i$ are active before and after re-rooting. Moreover, e'_i is not active before re-rooting, nor after.*

Proof. When the root is r_{i-1}, the path from r_{i-1} to r_i must either use arcs $e_{i-1}, \tilde{e}_{i-1}$ at times $3k$, or e'_{i-1} at time $k-(i-1) = k-i+1$. Either way, since e'_i has time $k-i < k-i+1$, it cannot be active at this moment. This also means that r_i reaches r_{i+1} through active arcs $e_i, \tilde{e}_i$, the only other possibility. To re-root to r_i, the arc flip must insert f_{i-1} for r_i to reach r_{i-1}, which means that e'_i is not inserted and still not active. This also means that $e_i, \tilde{e}_i$ remain active for r_i to reach r_{i+1}. □

Fact 6. *Let $i \in [k]$. When re-rooting from r_{i-1} to r_i, there is $u' \in U_i$ such that the path $r_i \to u' \to c_i$ is active before and after re-rooting.*

Proof. Let T be a temporal arborescence in which r_{i-1} is the root, such that a flip yields T' in which r_i becomes the root. Recall that the in-neighbors of c_i are r_{i-1}, r_k, and U_i. Any path from r_{i-1} to r_k uses an arc at timestamp greater than 0, and so the arc $(r_k, c_i, 0)$ cannot be active in T when r_{i-1} is the root. Also, since re-rooting to r_i inserts $f_{i-1} = (r_i, r_{i-1}, 4k)$, that arc cannot be inserted and is thus not in T', and moreover the arc $(r_{i-1}, c_i, 3k)$ cannot be active in T' because of time-consistency. The only option is that some arc $(u', c_i, 3k-1)$ must be active T'. Because that arc was not inserted, it was also active in T.

Consider the first arc on the path from r_i to that u' in T', where $u' \in U_i$. By Fact 5, e_i is active in T', but it cannot be the first arc of the r_i to u' path: e_i has timestamp $3k$ and prevents using $(u', c_i, 3k-1)$. The first arc of the path also cannot be f_{i-1} (at time $4k$) for the same reason. Also by Fact 5, e'_i is not active in T' and cannot be used on the path. By inspecting the set of out-neighbors of r_i, the only remaining option is some $(r_i, v', k+i)$ with $v' \in U_i$. Observe that if $v' \neq u'$, then v' cannot reach u', so $v' = u'$. Therefore, the path $r_i \to u' \to c_i$ is active after re-rooting, and since f_{i-1} is inserted to obtain T', no arc of this path is inserted by the flip, and it was thus also active before the re-rooting. □

Fact 7. *Let $i \in [k]$. When re-rooting from r_{i-1} to r_i, for each $j < i$ there are $u' \in U_i$ and $v' \in U_j$ such that the path $r_i \to u' \to v' \to c_j$ is active before and after re-rooting.*

Proof. Let $j < i$ and consider the path from r_i to c_j when r_i just became the root after a flip. This path cannot go through r_{i-1} because of the arc at time $4k$ from r_i to r_{i-1}, and because all incoming arcs of c_j have time $3k$ or less. As in the previous fact, the arc $(r_k, c_j, 0)$ also cannot be used (even when $i = k$, because that arc could not have been active prior to re-rooting, and was not inserted by the flip). Thus the in-neighbor of c_j on that path must be some $v' \in U_j$, using the arc $(v', c_j, 3k-1)$. If $i < k$, by Fact 5, e_i is active and, since it has time $3k > 3k-1$, it cannot be used on the path. Also e'_i is not active. Likewise, f_{i-1} cannot be on that path (also true when $i = k$). Thus, the path starts with some arc $(r_i, u', k+i)$, $u' \in U_i$. If v' is the out-neighbor of u' on the path, we are done, so assume otherwise. This out-neighbor cannot be c_i as it does not reach v', and it must be some $w' \in U_l$, where $j < l < i$. Notice that all the outgoing arcs of w' have timestamp $k+l < k+i$ and cannot be used because of $(r_i, u', k+i)$, except for the usable arc $(w', c_l, 3k-1)$. However, the latter leads to the sink c_l and cannot be used to reach c_j, contradicting that w' is on the path. We deduce that the arc $(u', v', k+i)$ is active. Therefore, the path $r_i \to u' \to v' \to c_j$ is active after the re-rooting occurred, and since the re-rooting was obtained by inserting f_{i-1}, no arc of this path was inserted by the arc flip that caused the re-rooting, so they were also active before. □

We can now establish that the in-neighbors of the c_j vertices correspond, as we iterate the roots from $r_0, r_1, \dots, r_k$, to a "growing" multicolored clique. To this end, for an arborescence T of D, define

$$C(T) = \{u' : (u', c_j, 3k-1) \text{ is active in } T, j \in [k]\}.$$

By Fact 6 and Fact 7, when T is the result of re-rooting to some r_i, $C(T)$ has one element $u' \in U_j$ for each $j \le i$ (and exactly one, since c_j cannot have two in-neighbors).

Fact 8. *Let $i \in [k]$. Let T be the first temporal arborescence of the sequence $\mathcal{R}$ in which r_{i-1} is the root, and let T' be the first arborescence in which r_i is the root. Then at least $i+4$ arc flips are required to transform T into T'.*

Moreover, exactly $i+4$ arc flips can be achieved only if $C(T') = C(T) \cup \{u'\}$ for some $u' \in U_i$, and u' is the in-neighbor of every element of $C(T)$ in T'.

Proof. We first argue that the statement is true for $i = 1$. Note that in this case, $r_{i-1} = r_0$ and thus $T = T_1$. By Fact 6, before the arc flip that yields T', we must have some arcs $(r_1, u', k+1)$ and $(u', c_1, 3k-1)$ that are active, which are absent from T_1. Moreover, before $(r_1, u', k+1)$ can become active, e'_0 must be inserted because e_0 has time $3k > k+1$. In addition, before r_1 can become the root, $(r_1, \tilde{r}_0, 4k)$ must have been inserted, since this is the only way for r_1 to reach $\tilde{r}_0$ in a time-respecting manner (using f_0 will not work). Finally, $(r_1, r_0, 4k)$ must be inserted for r_1 to reach r_0. Thus at least $5 = i+4$ new arcs, and thus flips, are required. Clearly $C(T') = \{u'\}$ and our statement holds since $C(T_0) = \emptyset$.

Now consider $i > 1$, still considering T with root r_{i-1} and T' with root r_i. We count the number of vertices that must change their in-neighbor from T to T', noting that an arc flip changes the in-neighbor of at most one vertex. Let $v' \in C(T)$, where $v' \in U_j$ is the in-neighbor of c_j for some $j \le i-1$. We argue that the in-neighbor of v' has changed from T to T'. Suppose that $j < i-1$. By Fact 7, since T is the first temporal arborescence with root r_{i-1}, in T the path $r_{i-1} \to u' \to v' \to c_j$ is active, where $u' \in U_{i-1}$. In particular, the arc $(u', v', k+i-1)$ is active in T. In T', by Fact 5, e_i is active and e'_i is not, meaning that all usable arcs that go out of r_i have time $k+i$ or more. This means that the $(u', v', k+i-1)$ arc cannot be active in T', and thus v' has changed its in-neighbor in T'. This proves the case $j < i$, and now suppose that $v' \in U_{i-1}$. Then by Fact 6 the arc $(r_{i-1}, v', k+i-1)$ is active in T. That arc cannot remain active in T', since f_{i-1} at time $4k$ must be used from r_i to r_{i-1}. Thus, v' also has a different in-neighbor in T'. Since this holds for every j up to $i-1$, we need $i-1$ distinct arc flips to change all the in-neighbors of $C(T)$.

Next, by Fact 6, in T' there is a path $r_i \to u' \to c_i$ with $u' \in U_i$. We argue that no arc of this path was in T. In T, e_{i-1} is active again by Fact 5 (at time $3k$). One can see that in T, any path from r_{i-1} to u' either uses that active e_{i-1}, or goes to r_0 and uses $(r_0, u', 4k)$. Aside from the latter, all the incoming arcs of u' have timestamp at most $2k$, preventing the usage of e_{i-1} and leaving the use of $(r_0, u', 4k)$ as the only possibility. This in turn implies that u' is not the in-neighbor of c_i in T. We thus require 2 more arc flips to change the in-neighbors of u' and c_i.

In T, by Fact 5, r_{i-1} reaches r_i through e_{i-1} and $\tilde{e}_{i-1}$, meaning that $\tilde{r}_{i-1}$ is not the in-neighbor of r_i in T. The in-neighbor of r_i must become r_{i-1} instead of $\tilde{r}_{i-1}$ at some point (by inserting e'_i) to allow the usage of arc $(r_i, u', k+i)$, which as we know must be activated before the re-rooting to r_i. Also, the in-neighbor of $\tilde{r}_{i-1}$ must eventually become r_i by inserting $(r_i, \tilde{r}_{i-1}, 4k)$ just as in the $i = 1$ case, and one more arc flip to insert f_{i-1} and make r_i the root. This adds 3 more arc flips, for a total of at least $i+4$, as claimed.

For the other part of our claim, to achieve $i+4$ arc flips we note that no vertex aside from those mentioned could have changed in-neighbor from T to T'. In particular, c_j keep the same in-neighbor for each $j \le i-1$. It follows that $C(T') = C(T) \cup \{u'\}$ for some u'. Moreover, only one $u' \in U_i$ may change its in-neighbor to r_i, and thus in T', only one u' can be used as new in-neighbor of every element of $C(T)$. □

We can finally conclude with the cleanup phase required by S.

Fact 9. *Let T be the first temporal arborescence in $\mathcal{R}$ in which r_k is the root. Then at least $2k$ arc flips are required to transform T into T_2.*

Proof. By Fact 6 and Fact 7, for every $i \in [k]$, there is some $u' \in U_i$ whose in-neighbor is not r_0, and the in-neighbor of c_i is not r_k. This accounts for at least $2k$ vertices whose in-neighbor must be changed through a (distinct) arc flip to attain T_2. □

To conclude the proof, since $p = 2k + \sum_{i=1}^{k}(i+4)$, Fact 8 and Fact 9 combined imply that for each $i \in [k]$, we must perform exactly $i+4$ arc flips to re-root from r_{i-1} to r_i. Again by Fact 8, each re-rooting to r_i adds exactly one vertex $u'_i \in U_i$ to the $C(T)$ set. Let $u'_1, \dots, u'_k$ be the vertices added during these re-rootings. We have that u'_2 has u'_1 as an out-neighbor, u'_3 has u'_1, u'_2 as out-neighbors, and more generally each u'_i has $u'_1, \dots, u'_{i-1}$ as out-neighbors. By construction, this implies that $u_1, \dots, u_k$ form a multicolored clique of G. 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 maximum 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. Note that for the parameter number of arc flips, the problem is in class XP. Indeed, let p be the number of arc flips needed to transform one arborescence into the other. We can enumerate the $O(|A|^{2p})$ sets of at most p arcs involved in some flip. For each set, we can enumerate the (at most) $O(p!)$ orders of the arc flips and check if they are reconfiguration sequences or not.

We present now some future directions.

- 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 $c \ge 1$.
- We also do not know how well the problem can be approximated when only two or three timestamps are present, and whether the problem is fixed-parameter tractable when using $O(1)$ timestamps.
- Further investigate the problem when the underlying digraph has specific structural properties (for example bounded treewidth or bounded degree), both in the approximation and parameterized complexity framework.
- Investigate whether the reconfiguration sequences, for yes-instances, have always polynomial length.

- In our definition of Temporal Arborescence Reconfiguration, p is encoded in unary, and in this case the problem is in NP; is the problem in NP if p is encoded in binary?

CRediT authorship contribution statement

Riccardo Dondi: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Manuel Lafond:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank the reviewers for their useful comments that helped us improve the presentation of the paper.

References

[1] Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, Christoforos L. Raptopoulos, The temporal explorer who returns to the base, J. Comput. Syst. Sci. 120 (2021) 179–193, https://doi.org/10.1016/j.jcss.2021.04.001.
[2] Noga Alon, Dana Moshkovitz, Shmuel Safra, Algorithmic construction of sets for k-restrictions, ACM Trans. Algorithms 2 (2) (2006) 153–177, https://doi.org/10.1145/1150334.1150336.
[3] Marco Amoruso, Daniele Anello, Vincenzo Auletta, Raffaele Cerulli, Diodato Ferraioli, Andrea Raiconi, Contrasting the spread of misinformation in online social networks, J. Artif. Intell. Res. 69 (2020) 847–879, https://doi.org/10.1613/jair.1.11509.
[4] Daniela Bubboloni, Costanza Catalano, Andrea Marino, Ana Silva, On computing optimal temporal branchings and spanning subgraphs, J. Comput. Syst. Sci. 148 (2025) 103596, https://doi.org/10.1016/J.JCSS.2024.103596.
[5] Isnard Lopes Costa, Raul Lopes, Andrea Marino, Ana Silva, On computing large temporal (unilateral) connected components, J. Comput. Syst. Sci. 144 (2024) 103548, https://doi.org/10.1016/J.JCSS.2024.103548.
[6] Irit Dinur, David Steurer, Analytical approach to parallel repetition, in: David B. Shmoys (Ed.), Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014, ACM, 2014, pp. 624–633.
[7] Riccardo Dondi, Manuel Lafond, On the complexity of temporal arborescence reconfiguration, in: 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2024), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.
[8] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, Stéphane Vialette, On the parameterized complexity of multiple-interval graph problems, Theor. Comput. Sci. 410 (1) (2009) 53–61, https://doi.org/10.1016/j.tcs.2008.09.065.
[9] Guillaume Fertin, Julien Fradin, Géraldine Jean, The maximum colorful arborescence problem: how (computationally) hard can it be?, Theor. Comput. Sci. 852 (2021) 104–120.
[10] Ziyun Guang, Matthew Smith-Erb, Layla Oesper, A weighted distance-based approach for deriving consensus tumor evolutionary trees, Bioinformatics 39 (2023) i204–i212.
[11] Petter Holme, Modern temporal network theory: a colloquium, Eur. Phys. J. B 88 (9) (2015) 234.
[12] Petter Holme, Jari Saramäki, A map of approaches to temporal networks, in: Temporal Network Theory, Springer, 2019, pp. 1–24.
[13] Mohammad Mehdi Hosseinzadeh, Mario Cannataro, Pietro Hiram Guzzi, Riccardo Dondi, Temporal networks in biology and medicine: a survey on models, algorithms, and tools, Netw. Model. Anal. Health Inform. Bioinform. 12 (1) (2023) 10, https://doi.org/10.1007/S13721-022-00406-X.
[14] Silu Huang, Ada Wai-Chee Fu, Ruifeng Liu, Minimum spanning trees in temporal graphs, in: Timos K. Sellis, Susan B. Davidson, Zachary G. Ives (Eds.), Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015, ACM, 2015, pp. 419–430.
[15] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, Yushi Uno, On the complexity of reconfiguration problems, Theor. Comput. Sci. 412 (12–14) (2011) 1054–1065, https://doi.org/10.1016/j.tcs.2010.12.005.
[16] Takehiro Ito, Yuni Iwamasa, Naoyuki Kamiyama, Yasuaki Kobayashi, Yusuke Kobayashi, Shun-ichi Maezawa, Akira Suzuki, Reconfiguration of time-respecting arborescences, in: Pat Morin, Subhash Suri (Eds.), Algorithms and Data Structures - 18th International Symposium, WADS 2023, Proceedings, Montreal, QC, Canada, July 31 - August 2, 2023, in: Lecture Notes in Computer Science, vol. 14079, Springer, 2023, pp. 521–532.
[17] Takehiro Ito, Yuni Iwamasa, Yasuaki Kobayashi, Yu Nakahata, Yota Otachi, Kunihiro Wasa, Reconfiguring (non-spanning) arborescences, Theor. Comput. Sci. 943 (2023) 131–141, https://doi.org/10.1016/j.tcs.2022.12.007.
[18] Naoyuki Kamiyama, Yasushi Kawase, On packing arborescences in temporal networks, Inf. Process. Lett. 115 (2) (2015) 321–325, https://doi.org/10.1016/j.ipl.2014.10.005.
[19] Richard M. Karp, Reducibility among combinatorial problems, in: Raymond E. Miller, James W. Thatcher (Eds.), 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, in: The IBM Research Symposia Series, Plenum Press, New York, 1972, pp. 85–103.
[20] David Kempe, Jon M. Kleinberg, Amit Kumar, Connectivity and inference problems for temporal networks, J. Comput. Syst. Sci. 64 (4) (2002) 820–842, https://doi.org/10.1006/jcss.2002.1829.
[21] Ning Li, 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.
[22] Othon Michail, An introduction to temporal graphs: an algorithmic perspective, Internet Math. 12 (4) (2016) 239–280, https://doi.org/10.1080/15427951.2016.1177801.
[23] Dana Moshkovitz, The projection games conjecture and the np-hardness of ln n-approximating set-cover, Theory Comput. 11 (2015) 221–235, https://doi.org/10.4086/toc.2015.v011a007.
[24] Naomi Nishimura, Introduction to reconfiguration, Algorithms 11 (4) (2018) 52, https://doi.org/10.3390/a11040052.