# Algorithms and Complexity for Path Covers of Temporal DAGs

Dibyayan Chakraborty, Antoine Dailly, Florent Foucaud, Ralf Klasing

# Algorithms and Complexity for Path Covers of Temporal DAGs

**Dibyayan Chakraborty**
School of Computing, University of Leeds, UK

**Antoine Dailly**
Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

**Florent Foucaud**
Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France

**Ralf Klasing**
Université de Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR 5800, Talence, France

## Abstract

A *path cover* of a digraph is a collection of paths collectively containing its vertex set. A path cover with minimum cardinality for a *directed acyclic graph* can be found in polynomial time [Fulkerson, AMS'56; Cáceres et al., SODA'22]. Moreover, Dilworth's celebrated theorem on chain coverings of partially ordered sets equivalently states that the minimum size of a path cover of a DAG is equal to the maximum size of a set of mutually unreachable vertices. In this paper, we examine how far these classic results can be extended to a dynamic setting.

A temporal digraph has an arc set that changes over discrete time-steps; if the underlying digraph is acyclic, then it is a *temporal DAG*. A temporal path is a directed path in the underlying digraph, such that the time-steps of arcs are strictly increasing along the path. Two temporal paths are temporally disjoint if they do not occupy any vertex at the same time. A *temporal path cover* is a collection $\mathcal{C}$ of temporal paths that covers all vertices, and $\mathcal{C}$ is *temporally disjoint* if all its temporal paths are pairwise temporally disjoint. We study the computational complexities of the problems of finding a minimum-size temporal (disjoint) path cover (denoted as Temporal Path Cover and Temporally Disjoint Path Cover).

On the negative side, we show that both Temporal Path Cover and Temporally Disjoint Path Cover are NP-hard even when the underlying DAG is planar, bipartite, subcubic, and there are only two arc-disjoint time-steps. Moreover, Temporally Disjoint Path Cover remains NP-hard even on temporal oriented trees. We also observe that natural temporal analogues of Dilworth's theorem on these classes of temporal DAGs do not hold.

In contrast, we show that Temporal Path Cover is polynomial-time solvable on temporal oriented trees by a reduction to Clique Cover for (static undirected) *weakly chordal* graphs (a subclass of perfect graphs for which Clique Cover admits an efficient algorithm). This highlights an interesting algorithmic difference between the two problems. Although it is NP-hard on temporal oriented trees, Temporally Disjoint Path Cover becomes polynomial-time solvable on temporal oriented *lines* and temporal *rooted directed* trees.

Motivated by the hardness result on trees, we show that, in contrast, Temporal Path Cover admits an XP time algorithm with respect to parameter $t_{\max} + tw$, where $t_{\max}$ is the maximum time-step and $tw$ is the treewidth of the underlying static undirected graph; moreover, Temporally Disjoint Path Cover admits an FPT algorithm with respect to the same parameterization.

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

**Keywords and phrases** Temporal Graphs, Dilworth's Theorem, DAGs, Path Cover, Temporally Disjoint Paths, Algorithms, Oriented Trees, Treewidth

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2024.38

**Related Version** *Full Version*: https://arxiv.org/abs/2403.04589 [9]





**Funding** This research was partially supported by the ANR projects GRALMECO (ANR-21-CE48-0004) and TEMPOGRAL (ANR-22-CE48-0001) and by the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25.

## 1 Introduction

A classic theorem of Dilworth from 1950 [14] states that in any partially ordered set (poset), the minimum number of chains required to cover all the elements is equal to the maximum size of an antichain. Dilworth's theorem is fundamental from the mathematical point of view; furthermore, an algorithmic proof (that enables to construct a chain cover and an antichain in polynomial time) was published by Fulkerson in 1956 [17]. This theorem and its algorithmic form have many applications, not only in combinatorics, but also in various fields such as bioinformatics [6], scheduling [32], databases [22], program testing [36], etc.

A collection $\mathcal{P}$ of (resp. pairwise vertex-disjoint) directed paths of a digraph $D$ is a *path cover* (resp. *path partition*) of $D$ if all vertices of $D$ are contained in some path of $\mathcal{P}$. Dilworth's theorem can be restated in an equivalent form, equating the minimum cardinality of path covers on directed acyclic graphs (DAGs) and the maximum size of a set of pairwise "unreachable" vertices, or *antichain* vertices [4, 5, 16].

Fulkerson [17] showed that finding a minimum-size path cover of a DAG can be done in polynomial time. Moreover, by using similar methods, one can also find a minimum-size path *partition* in polynomial time for arbitrary DAGs (see [11, Probl. 26-2] or [15, Chapter 11.5]). Improving the best known algorithms for path cover and partitions of DAGs is still an active field of research, see for example [4, 5, 10, 28, 31] for some recent results.

The notions of directed paths and path covers naturally extends to *temporal (di)graphs*. Informally, the arc set of a temporal digraph changes over discrete time-steps and *labels* of an arc are the time-steps where the arc appears. Temporal (di)graphs have been extensively studied in the two last decades, with contributions from and applications to various fields, see [7, 21, 23, 34, 35, 37]. A *temporal path* of a digraph is a path that traverses edges appearing at strictly increasing time-steps. The asymmetric nature of temporal paths has motivated many recent algorithmic works on related reachability or path problems on temporal graphs, such as [1, 2, 3, 8, 24, 33].

Two temporal paths are *temporally disjoint* if they do not occupy a vertex at the same time-step. Motivated by applications in artificial intelligence, this definition was introduced by Klobas et al. [25] and has since then garnered some attention from the graph algorithmic community [29]. Even though the above notion was introduced in the context of temporal undirected graphs, it naturally extends to temporal digraphs and motivates the corresponding covering problems. The objective of Temporal Path Cover (resp. Temporally Disjoint Path Cover) is to cover an input temporal digraph by a minimum number of temporal paths (resp. temporally disjoint paths).

**Main objectives.** In this paper, we initiate the algorithmic study of Temporal Path Cover and Temporally Disjoint Path Cover and focus on *temporal directed acyclic graphs* (or simply, temporal DAGs). A temporal digraph is a *temporal DAG* if the union of all arcs across all time-steps induces a (static) DAG. We say that a temporal digraph satisfies the *Dilworth property* (resp. *temporally disjoint Dilworth property*, or *TD-Dilworth property* for short) if the largest size of a *temporal antichain* (understood as a set of pairwise temporally unreachable vertices) is equal to the smallest size of a temporal path cover (resp. temporally disjoint path cover). The main goals of this paper are the following:

**(a)** Determine classes of temporal DAGs satisfying the (TD-)Dilworth property.
**(b)** Study the computational complexities of Temporal Path Cover and Temporally Disjoint Path Cover on temporal digraphs.

**Practical motivation.** Similar to the problems studied by Klobas et al. [25], one natural application is *Multi-Agent Path Finding* (MAPF) [13, 40], which can be applied for example to crime prevention in transportation networks [44]. In this setting, $k$ agents are assigned the task of surveying a location: collecting data, moving objects around, looking out for hazards, etc. When the changes over time are predictable (train network, irrigation periods in a field, departure and arrival of vehicles in a logistics area, blockades at given times in a post-disaster area, naturally occurring blockades such as tides, ...), the location is modeled as a temporal digraph. If the location digraph does not contain directed cycles, it is modeled by a temporal DAG (for example, if it is inherently directed from a start area towards a target area). The exploration path of an agent can be modeled by a temporal path. Now, Temporal Path Cover corresponds to the situation where the agents need to explore the whole location, while for Temporally Disjoint Path Cover, the agents also cannot be simultaneously at the same place, a scenario described as *vertex-conflicts* in the literature [39]. In both cases, we want to minimize the number $k$ of agents.

*Spatio-temporal security games* [42, 43] are a natural example of applicability of MAPF. In these games, defenders want to protect spatially-located resources from attackers by covering them, and have to take time into account whenever they are moving (to be sure that the edges are available when they need to use them). In these games, the temporal dimension is generally represented by adding one layer of space per time-step. This representation induces a DAG, with as many vertices as the size of the initial graph multiplied by the number of time-steps. It seems natural to encode time through the more compact model of temporal graphs, which allows these games to be modeled by Temporal Path Cover and Temporally Disjoint Path Cover.

**Our results.** We begin by formally defining the problems studied in this paper.

Temporal Path Cover (TPC)

Input: A temporal digraph $D$, an integer $k$.

Problem: Does there exist a set $\mathcal{C}$ of $k$ temporal paths in $D$ such that every vertex of $D$ is covered by some path of $\mathcal{C}$?

Temporally Disjoint Path Cover (TD-PC)

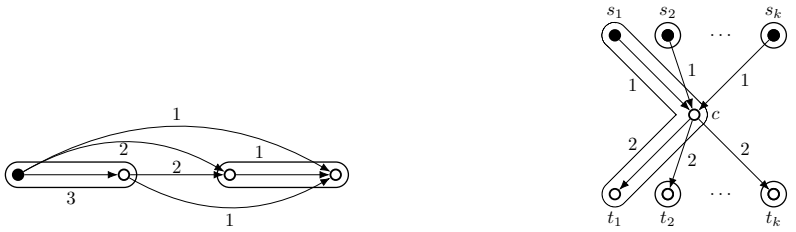
Input: A temporal digraph $D$, an integer $k$.

Problem: Does there exist a set $\mathcal{C}$ of $k$ temporally disjoint temporal paths in $D$ such that every vertex of $D$ is covered by some path of $\mathcal{C}$?

We observe that in general, temporal DAGs do not have the Dilworth property (see Figure 1a). Then, we prove the following negative result.

▶ **Theorem 1.** *Temporal Path Cover and Temporally Disjoint Path Cover are NP-hard on temporal DAGs, even if the input is planar, bipartite, subcubic, of girth 10, has only one time label per arc, and every label is either* 1 *or* 2*.*

A temporal DAG $D$ is a *temporal oriented tree* if the underlying directed graph of $D$ is a tree. On the positive side, we prove the following.




**(a)** A temporal DAG not having the Dilworth property.

**(b)** A temporal oriented tree not having the TD-Dilworth property.

**Figure 1** Each encircled area is a path of a minimum-size (temporally disjoint) temporal path cover, vertices in a maximum-size temporal antichain are in black.

▶ **Theorem 2.** *There is an $\mathcal{O}(\ell n^2 + n^3)$-time algorithm for* Temporal Path Cover *on temporal oriented trees with $n$ vertices and at most $\ell$ many labels per arc. Furthermore, temporal oriented trees satisfy the Dilworth property.*

We briefly describe the technique we use for proving Theorem 2. Two vertices of a temporal digraph are *temporally connected* if there exists a temporal path from one to the other. The *connectivity graph* of a temporal digraph $D$ is an undirected (static) graph whose vertex set is the same as that of $D$, and whose edge set consists of all pairs of temporally connected vertices. To prove the above theorem, we show that the connectivity graph of a temporal oriented tree is a *weakly chordal graph* [19] (a subclass of *perfect graphs*). We show Temporal Path Cover can be reduced to Clique Cover on weakly chordal graphs. The above observation, combined with the Weak Perfect Graph Theorem (proved by Lovász [30]), proves that temporal oriented trees satisfy the Dilworth property. Moreover, the existing $\mathcal{O}(nm)$-time algorithm [20] to compute a minimum clique cover of a weakly chordal graph (having $n$ vertices and $m$ edges) completes the proof of Theorem 2. Our proof gives interesting structural information on the interaction between temporal paths in temporal oriented trees. Interestingly, another important class of perfect graphs plays an important role in connection with Dilworth's theorem and its translation to the setting of static DAGs: the class of comparability graphs, see [18, Chapter 5.7]. In our case, there does not appear to be any connection to comparability graphs.

On the other hand, temporal oriented trees do not satisfy the TD-Dilworth property (see Figure 1b for an example). Then, we prove the following negative result.

▶ **Theorem 3.** *Temporally Disjoint Path Cover is NP-hard on temporal oriented trees.*

To find classes that satisfy the TD-Dilworth property, we study *temporal oriented lines* (that is, where the underlying digraph is an oriented path) and *temporal rooted directed trees*. A tree is a *rooted directed tree* if it is an oriented tree with a single source vertex called the *root*. We prove the following result.

▶ **Theorem 4.** *Temporal Path Cover and Temporally Disjoint Path Cover can be solved in time:*
**(a)** *$\mathcal{O}(\ell n)$ on temporal oriented lines;*
**(b)** *$\mathcal{O}(\ell n^2)$ on temporal rooted directed trees;*
*where $\ell$ is the maximum number of labels per arc and $n$ is the number of vertices. Furthermore, both classes satisfy the TD-Dilworth property.*

Note that some related problems remain NP-hard for temporal lines, such as Temporally Disjoint Walks [26]. Theorem 4(a) shows that this is not the case here. To prove

**Table 1** Summary of our algorithmic results. For all polynomial-time solvable classes of temporal DAGs, we also show that the Dilworth property (or TD-Dilworth property for TD-PC) holds.

| temporal graph class | TPC | TD-PC |
|---|---|---|
| temporal DAGs (planar bipartite subcubic, girth 10, two arc-disjoint time-steps) | NP-c. | NP-c. |
| temporal oriented trees | poly $\mathcal{O}(\ell n^2 + n^3)$ | NP-c. |
| temporal rooted directed trees | poly $\mathcal{O}(\ell n^2)$ | poly $\mathcal{O}(\ell n^2)$ |
| temporal oriented lines | poly $\mathcal{O}(\ell n)$ | poly $\mathcal{O}(\ell n)$ |
| general temporal digraphs with bounded treewidth tw and number of time-steps $t_{\max}$ | poly (XP w.r.t. tw and $t_{\max}$) | poly (FPT w.r.t. tw and $t_{\max}$) |

Theorem 4(b), we begin by constructing a temporal path cover before transforming it into a temporally disjoint one of the same size. This is in contrast with general temporal oriented trees, for which, by Theorem 3, such an approach is not possible.

As Temporally Disjoint Path Cover is NP-hard even on temporal oriented trees and on temporal DAGs with two time-steps, a natural question is what happens when the number of time-steps is small *and* the underlying digraph is a tree. Motivated by this question, we study the case where both the number of time-steps and the treewidth of the underlying digraph are bounded (where we define the *treewidth* of a temporal digraph as the treewidth of the underlying static undirected graph). We show that both problems become tractable in this setting. More precisely, we give a fixed-parameter tractable (FPT) algorithm for Temporally Disjoint Path Cover with treewidth and number of time-steps as parameters. The same technique gives an XP algorithm for Temporal Path Cover.

▶ **Theorem 5.** *There is an algorithm for Temporally Disjoint Path Cover on general temporal digraphs that is FPT with respect to the treewidth of the underlying undirected graph and the maximum number of labels per arc. For Temporal Path Cover on general temporal digraphs, there is an XP algorithm for the same parameter.*

See Table 1 for a summary of our algorithmic results.

**Further related work.** Algorithms for solving several types of path and distance problems in temporal graphs have been developed, see for example [3, 24, 41]. Recently, the problem Temporally Disjoint Paths was introduced in [25], as a generalization of the notorious Disjoint Paths problem (also known as Linkage). In Temporally Disjoint Paths, one is given a temporal graph with $k$ pairs of vertices called *terminals*, and the goal is to find a set of $k$ pairwise temporally disjoint paths, each of them connecting one pair of terminals. Temporally Disjoint Paths is NP-hard, even for temporal lines and two paths [25] or temporal stars [29], but becomes FPT for trees when parameterized by the number of paths [25]. Algorithms that are FPT for certain structural parameters are given in [29].

**Structure of the paper.** We start with the hardness result for temporal DAGs (Theorem 1) in Section 3. We then prove our results for temporal oriented trees (Theorem 2 and Theorem 3) in Sections 4 and 5. We prove Theorem 4, the polynomial-time algorithms for special temporal oriented trees (temporal rooted directed trees and temporal oriented lines), in Section 6. We then prove our results for temporal digraphs of bounded treewidth and number of time-steps

(Theorem 5) in Section 7. We conclude in Section 8. Due to space constraints, proofs of propositions and lemmas marked with **(*)** are omitted here and can be found in the full version of the paper [9].

## 2 Preliminaries

A *temporal digraph* $\mathcal{D} = (V, A_1, \ldots, A_{t_{\max}})$ is given by a sequence of arc-sets representing $t_{\max}$ discrete *time-steps* $\{1, \ldots, t_{\max}\}$, where an arc in $A_i$ is *active* at time-step $i$ [25]. We denote by $D = (V, A)$, where $A = \cup_{i=1}^{t_{\max}} A_i$, the *underlying digraph* of $\mathcal{D}$ (sometimes called the *footprint (di)graph* [7]). Equivalently, one can view the time-steps as an arc-labelling function $\lambda : A(D) \to 2^{[t_{\max}]}$, where $\lambda(\overrightarrow{xy}) \subseteq [1, t_{\max}]$ is the set of time-steps where $\overrightarrow{xy}$ is active [24]. In that case, we denote the temporal digraph as $\mathcal{D} = (D, \lambda)$. We say that a temporal digraph has a given property $\mathcal{P}$ (planarity, given girth, ...) if the underlying digraph obtained by forgetting the orientation of the arcs of its underlying digraph has property $\mathcal{P}$. Similarly, we call a temporal digraph a temporal DAG (resp. temporal oriented tree, temporal line, ...) if its underlying digraph is a DAG (resp. oriented tree, path, ...). For a given temporal digraph, we denote by $\ell$ the maximum number of labels per arc and by $n$ the number of vertices in the underlying digraph.

For a (temporal) (di)graph $\mathcal{D}$ and subset $S$ of its vertices (resp. edges), $\mathcal{D} \setminus S$ denotes the (temporal) (di)graph obtained by removing the vertices (resp. edges) in $S$ from $\mathcal{D}$.

In a temporal digraph, a *temporal (directed) path* is a sequence $(v_1, v_2, t_1), (v_2, v_3, t_2), \ldots, (v_{k-1}, v_k, t_{k-1})$ such that for any $i, j$ with $1 \leq i < j \leq k$, $v_i \neq v_j$ and for any $i$ with $1 \leq i \leq k-1$, $t_i < t_{i+1}$ and there is an arc $\overrightarrow{v_i v_{i+1}}$ at time-step $t_i$. These paths are sometimes called *strict* in the literature.[1] For a temporal path $P = (v_1, v_2, t_1), \ldots, (v_{k-1}, v_k, t_{k-1})$, we denote by $V(P)$ the set $\cup_{i=1}^{k} \{v_i\}$ and by $A(P)$ the set $\cup_{i=1}^{k-1} \{\overrightarrow{v_i v_{i+1}}\}$. Note that we allow a temporal path to contain exactly one vertex and no arc.

The *length* of a temporal path is the number of arcs it uses. We say that a temporal path $P = (v_1, v_2, t_1), \ldots, (v_{k-1}, v_k, t_{k-1})$ *occupies* vertex $v_i$ during the time interval $[t_{i-1}, t_i]$. Two temporal paths $P_1, P_2$ *temporally intersect* if there is a vertex $v \in V(P_1) \cap V(P_2)$ and two time intervals $[a_1, b_1], [a_2, b_2]$ where $[a_1, b_1] \cap [a_2, b_2] \neq \emptyset$ such that $P_1$ (resp. $P_2$) occupies $v$ during $[a_1, b_1]$ (resp. $[a_2, b_2]$). Two temporal paths are *temporally disjoint* if they do not temporally intersect. In other words, they do not occupy the same vertex at the same time. A *temporal path cover* (resp. *temporally disjoint path cover*) of a temporal digraph $D$ is a collection of temporal paths (resp. temporally disjoint paths) that cover all vertices of $D$. Two vertices are *temporally connected* in $D$ if there exists a temporal path between them. A *temporal antichain* is a set of vertices that are pairwise not temporally connected.

▶ **Definition 6.** *A class $\mathcal{C}$ has the* Dilworth property *(resp.* TD-Dilworth property*) if, for every digraph $D \in \mathcal{C}$, the cardinality of a minimum temporal path cover (resp. temporally disjoint path cover) of $D$ is equal to the maximum cardinality of a temporal anti-chain in $D$.*

A *hole* of a static undirected graph is an induced cycle of length at least 5, and an *anti-hole* is the complement of a hole. A hole or anti-hole is *even* (resp. *odd*) if it has an even (resp. odd) number of vertices. A graph $G$ is *weakly chordal* if it has neither a hole nor an anti-hole. A *(minimum) clique cover* of a graph $G$ is a (minimum cardinality) set of complete subgraphs of $G$ that covers all vertices. A *(maximum) independent set* of a graph $G$ is a (maximum cardinality) set of pairwise non-adjacent vertices. We shall use the following results for weakly chordal graphs.

[1] For non-strict paths, the condition $t_i < t_{i+1}$ is replaced with $t_i \leq t_{i+1}$; argued in [29], the strict definition is more natural for applications where an agent cannot traverse any number of arcs at once.

▶ **Theorem 7** ([20, 30, 38]). *Let $H$ be a weakly chordal graph with $n$ vertices and $m$ edges. Then, a minimum clique cover of $H$ can be found in $\mathcal{O}(nm)$-time. Furthermore, the maximum size of an independent set of $H$ equals the minimum size of a clique cover of $H$.*

## 3 Temporal DAGs

We provide a reduction from a restricted variant of 3-Dimensional Matching to prove the following (proof deferred to the full version [9] due to space constraints).

▶ **Theorem 1.** *Temporal Path Cover and Temporally Disjoint Path Cover are NP-hard on temporal DAGs, even if the input is planar, bipartite, subcubic, of girth 10, has only one time label per arc, and every label is either 1 or 2.*

We also show the following.

▶ **Proposition 8 (*).** *There are temporal DAGs (whose underlying digraph is a transitive tournament) that satisfy neither the Dilworth nor the TD-Dilworth property. Moreover, the ratio between the minimum-size temporal path cover and the maximum-size temporal antichain can be arbitrarily large.*

## 4 Temporal Path Cover on temporal oriented trees

In this section we prove the following theorem.

▶ **Theorem 2.** *There is an $\mathcal{O}(\ell n^2 + n^3)$-time algorithm for Temporal Path Cover on temporal oriented trees with $n$ vertices and at most $\ell$ many labels per arc. Furthermore, temporal oriented trees satisfy the Dilworth property.*

For the rest of this section, $\mathcal{T} = (T, \lambda)$ shall denote a temporal oriented tree with $n$ vertices and at most $\ell$-many labels per edge. We construct the *connectivity graph of* $\mathcal{T}$, denoted by $G$, as follows: $V(G) = V(T)$ and $E(G) = \{uv \mid u \neq v$ and $u$ and $v$ are temporally connected$\}$. In other words, the connectivity graph of a temporal oriented tree connects vertices that are temporally connected. Observe that $G$ can be constructed in $\mathcal{O}(\ell n^2)$-time. The next observation follows immediately from the definition.

▶ **Observation 9.** *A set $S$ of vertices of $\mathcal{T}$ is a temporal antichain if and only if $S$ induces an independent set in $G$.*

We have the following relationship between temporal paths in $\mathcal{T}$ and cliques in $G$.

▶ **Lemma 10.** *Let $S$ be a set of vertices of $\mathcal{T}$. Then $S$ is contained in a temporal path in $\mathcal{T}$ if and only if $S$ is contained in a clique of $G$.*

**Proof.** Let $S$ be contained in temporal path $P$ in $\mathcal{T}$. Let $u_1, u_2, \ldots, u_k$ where $k = |S|$, be the ordering of the vertices in $S$ as they are encountered while traversing $P$ from the source to the sink. Notice that, for each $1 \leq i < j \leq k$, there is a temporal path from $u_i$ to $u_j$. Therefore, $u_i$ is adjacent to $u_j$ in $G$. Hence, $S$ is contained in a clique of $G$.

Let $S$ be contained in a clique of $G$ and $S'$ be a maximal complete subgraph of $G$ such that $S \subseteq V(S')$. Now, we orient the edges of $S'$ to create a digraph $\overrightarrow{S'}$ as follows. For an edge $uv \in E(S')$, we introduce an arc $\overrightarrow{uv} \in A(\overrightarrow{S'})$ if there is a temporal path from $u$ to $v$ in $\mathcal{T}$. Since $\mathcal{T}$ is acyclic, $\overrightarrow{S'}$ is a transitive tournament. Hence, there is an ordering $u_1, u_2, \ldots, u_k$ of the vertices of $S'$ where $k = |V(S')|$ such that for $1 \leq i < j \leq k$, there is a temporal

path from $u_i$ to $u_j$ in $\mathcal{T}$. Now, consider any temporal path $P$ from $u_1$ to $u_k$ in $\mathcal{T}$. ($P$ exists as $\overrightarrow{u_1u_k} \in A(\overrightarrow{S'})$. Since $\mathcal{T}$ is a temporal oriented tree, $P$ will contain all vertices of $S'$ and therefore of $S$. ◀

Following is an immediate corollary of the above.

▶ **Corollary 11.** *The minimum cardinality of a temporal path cover of $\mathcal{T}$ is equal to the minimum cardinality of a clique cover of $G$.*

We will often use the following lemma.

▶ **Lemma 12.** *Let $\{u, v, w, x\} \subseteq V(T)$ be four vertices such that there is a temporal path $P_1$ from $u$ to $v$ and a temporal path $P_2$ from $w$ to $x$. If $P_1$ and $P_2$ have a vertex in common, then, there is a temporal path from $u$ to $x$, or a temporal path from $w$ to $v$, or both.*

**Proof.** Assume that there is no temporal path from $u$ to $x$. Let $y$ be the vertex of a temporal path from $w$ to $x$ that is closest to $u$ in $T$. Let $t$ be the smallest integer such that there is a temporal path from $u$ to $v$ that reaches $y$ at time-step $t$. Observe that no temporal path from $y$ to $x$ can start at time-step $t' > t$ since, otherwise, there would be a temporal path from $u$ to $x$. This implies that all temporal paths between $w$ and $x$ reach $y$ at time-step $t'' \leq t$. Let $P_1$ be a temporal path from $w$ to $y$ which is also a subpath of a temporal path from $w$ to $x$. Let $P_2$ be a temporal path from $y$ to $v$ which is also a subpath of a temporal path from $u$ to $v$. The above arguments imply that the arc incident with $y$ in $P_1$ has time-step at most $t$. Similarly, the arc incident with $y$ in $P_2$ has time-step strictly greater than $t$. Hence, the concatenation of $P_1$ and $P_2$ is a temporal path in $\mathcal{T}$ from $w$ to $v$. ◀

## 4.1 The case of holes

In this subsection, we will show that the connectivity graph $G$ does not contain any holes. We use the following lemma.

▶ **Lemma 13.** *Let $H$ be an induced cycle of length at least 4 in $G$. Then, for every vertex $v \in V(H)$ and every arc $\overrightarrow{a}$ of $T$ incident with $v$, the vertices of $H \setminus \{v\}$ lie in the same weakly connected component of $T \setminus \{\overrightarrow{a}\}$.*

**Proof.** For the sake of contradiction, let there exist vertices $\{u, v, w\} \subseteq V(H)$ and an arc $\overrightarrow{a}$ of $T$ incident with $v$ such that $u$ and $w$ lie in two different connected components of $T' = T \setminus \{\overrightarrow{a}\}$. Let $C_u$ and $C_w$ be the sets of vertices of $H \setminus \{v\}$ contained in the same connected component as $u$ and $w$, respectively. Since $H \setminus \{v\}$ is connected, there exist $u' \in C_u$ and $w' \in C_w$ such that $u'w' \in E(H)$ *i.e.* $u'w' \in E(G)$. Hence, there is a temporal path $P$ from $u'$ to $w'$ or $w'$ to $u'$ in $\mathcal{T}$. Since $T$ is a tree, $P$ must contain $v$. Lemma 10 implies that $\{u', v, w'\}$ forms a subset of a clique in $G$, and therefore $\{u', v, w'\}$ forms a triangle. But this contradicts that $H$ is a hole. ◀

Going forward, we need the following notations. For an edge $e = uv \in E(G)$, let $Q_e$ denote a temporal path from $u$ to $v$ or $v$ to $u$ in $\mathcal{T}$. For an induced cycle $H$ of length at least 4 in $G$, let $T_H$ denote the smallest connected subtree of $T$ containing all vertices of $H$. Lemma 13 implies that every vertex of $H$ must be a leaf in $T_H$ (that is, a vertex with degree 1 in the underlying undirected tree). For a vertex $v \in V(H)$, let $\overrightarrow{a}(v)$ be the arc incident with $v$ in $T_H$. Let $H$ be an induced cycle of length at least 4 in $G$. We can partition the vertex set of $H$ into two sets $IN(H)$ and $OUT(H)$ as follows: a vertex $v \in V(H)$ is in


**Figure 2** On the left, a hole $H$ in the connectivity graph. On the right, its corresponding vertices in the oriented subtree $T_H$, with $T'_H = T_H \setminus V(H)$. Vertices in $IN(H)$ are in black.

$IN(H)$ if $\overrightarrow{a}(v)$ is directed towards $v$, and otherwise $v$ is in $OUT(H)$ (see Figure 2 for an illustration of these definitions).

For a vertex $v \in IN(H)$, notice that both neighbors of $v$ in $H$ must lie in $OUT(H)$, and vice versa, since they must be connected by a directed path in $T$. Hence, $H$ is bipartite, and therefore $G$ does not contain any odd hole:

▶ **Lemma 14.** *The connectivity graph $G$ does not contain any odd hole.*

Without loss of generality, we assume in the following that $OUT(H)$ (resp. $IN(H)$) contains every odd-indexed (resp. even-indexed) vertex of $H$. For an even hole $H$ whose vertices are cyclically ordered as $u_1, u_2, \ldots, u_k$, we use a cyclic definition of addition, so $u_{k+1} = u_1$. We first prove the following lemmas.

▶ **Lemma 15.** *Let $H$ be an even hole in the connectivity graph $G$. Then, for every $i$, $Q_{u_iu_{i+1}}$ and $Q_{u_{i+2}u_{i+3}}$ share a common vertex in $T$.*

**Proof.** Assume by contradiction that $Q_{u_iu_{i+1}}$ and $Q_{u_{i+2}u_{i+3}}$ are vertex-disjoint. Assume without loss of generality that $Q_{u_iu_{i+1}}$ goes from $u_i$ to $u_{i+1}$. Note that, since each vertex of the hole is a leaf of $T_H$ as a consequence of Lemma 13, the two paths $Q_{u_iu_{i+1}}$ and $Q_{u_{i+1}u_{i+2}}$ have to share a common vertex other than $u_{i+1}$ (its neighbour in $T_H$). By the same reasoning, $Q_{u_{i+1}u_{i+2}}$ and $Q_{u_{i+2}u_{i+3}}$ share a common vertex other than $u_{i+2}$. Hence, since the three paths $Q_{u_iu_{i+1}}$, $Q_{u_{i+1}u_{i+2}}$ and $Q_{u_{i+2}u_{i+3}}$ are in $T_H$, and $Q_{u_iu_{i+1}}$ and $Q_{u_{i+2}u_{i+3}}$ are vertex-disjoint, there is an arc $\overrightarrow{a}$ contained in $Q_{u_{i+1}u_{i+2}}$ that separates $Q_{u_iu_{i+1}}$ and $Q_{u_{i+2}u_{i+3}}$.

Removing $\overrightarrow{a}$ from $T$ partitions the vertices of $H$ into two sets $H_1$ and $H_2$: $H_1$ (resp. $H_2$) contains the vertices of $H$ that are in the same part of $T \setminus \overrightarrow{a}$ as $u_{i+1}$ (resp. $u_{i+2}$). Now, since $H$ is a cycle, there is an edge $u_ju_{j+1}$ such that (without loss of generality) $u_j \in H_1$, $u_{j+1} \in H_2$ and $(j, j+1) \neq (i+1, i+2)$. This implies that the path $Q_{u_ju_{j+1}}$ has to use $\overrightarrow{a}$ in $\mathcal{T}$, and thus $Q_{u_{i+1}u_{i+2}}$ and $Q_{u_ju_{j+1}}$ share a common vertex. Hence, Lemma 12 implies that there is a temporal path from $u_{j+1}$ to $u_{i+1}$ or from $u_{i+2}$ to $u_j$. However, since $j \neq i+3$ ($u_j \in H_1$ and $u_{i+3} \in H_2$) and $j+1 \neq i$ ($u_{j+1} \in H_2$ and $u_i \in H_1$), both temporal paths would induce a chord in $H$, a contradiction. ◀

▶ **Lemma 16.** *The connectivity graph $G$ does not contain any hole of size 6.*

**Proof.** Assume by contradiction that there is a hole on six vertices $u_1, \ldots, u_6$. We know that $Q_{u_1u_2}$ and $Q_{u_4u_5}$ are vertex-disjoint (since otherwise, by Lemma 12, at least one of the chords $u_1u_4$ or $u_2u_5$ would exist). The $u_i$'s are leaves of $T_H$, so $Q_{u_1u_2}$ and $Q_{u_1u_6}$, being paths with a common leaf in the same subtree, share at least one common vertex other than $u_1$ (its neighbour in $T_H$), let $v$ be the last (with respect to the orientation of $T$) vertex in their common subpath. Now, $Q_{u_5u_6}$ has a common vertex with both $Q_{u_1u_2}$ (by Lemma 15) and $Q_{u_1u_6}$ (the neighbour of $u_6$ in $T_H$), so it has to contain $v$ by the Helly property of subtrees of a tree. By the same reasoning, $Q_{u_4u_5}$ and $Q_{u_5u_6}$ share at least one common vertex other

than $u_5$ (its neighbour in $T_H$), let $w$ be the last vertex in their common subpath. The Helly property of subtrees of a tree again implies that both $Q_{u_2u_3}$ and $Q_{u_3u_4}$ have to contain $w$, since they pairwise intersect with $Q_{u_4u_5}$. But this means that $Q_{u_2u_3}$ and $Q_{u_5u_6}$ share both $v$ and $w$ as common vertices, and so by Lemma 12 there is at least one of the two chords $u_2u_5$ or $u_3u_6$, a contradiction. ◀

We can now prove that there is no even hole in $G$:

▶ **Lemma 17.** *The connectivity graph $G$ does not contain any even hole.*

**Proof.** Assume by contradiction that $G$ contains an even hole $H$ on $k \geq 8$ vertices ($k = 6$ is impossible by Lemma 16). We know by Lemma 15 that both $Q_{u_3u_4}$ and $Q_{u_{k-1}u_k}$ intersect $Q_{u_1u_2}$, but do not intersect each other (otherwise, by Lemma 12, at least one of the edges $u_3u_k$ or $u_4u_{k-1}$ would exist, and both would be chords since $k \geq 8$), so there is an arc $\overrightarrow{a}$ in $T$ that separates them. Removing $\overrightarrow{a}$ from $T$ partitions the vertices of $H$ into two sets $H_1$ and $H_2$: $H_1$ (resp. $H_2$) contains the vertices of $H$ that are in the same part of $T \setminus \overrightarrow{a}$ as $u_3$ (resp. $u_k$). Now, since $H$ is a cycle, there is an edge $u_ju_{j+1}$ such that (without loss of generality) $u_j \in H_1$ and $u_{j+1} \in H_2$. This implies that the path $Q_{u_ju_{j+1}}$ has to use $\overrightarrow{a}$ in $\mathcal{T}$, and thus $Q_{u_1u_2}$ and $Q_{u_ju_{j+1}}$, both containing $\overrightarrow{a}$, share a common vertex. Hence, Lemma 12 implies that there is a temporal path from $u_{j+1}$ to $u_2$ or from $u_1$ to $u_j$. However, since $j \neq k$ ($u_j \in H_1$ and $u_k \in H_2$) and $j + 1 \neq 3$ ($u_{j+1} \in H_2$ and $u_3 \in H_1$), by Lemma 12 both temporal paths would induce a chord in $H$, a contradiction. ◀

## 4.2 The case of anti-holes

In this subsection, we will show that the connectivity graph $G$ does not contain any anti-hole. For an anti-hole $H$, let its vertices be circularly ordered as $u_1, u_2, \ldots, u_k$ as they are encountered while traversing the complement of $H$ (which is a hole). Let $ODD(H)$ (resp. $EVEN(H)$) denote the set of vertices with odd (resp. even) indices.

▶ **Lemma 18.** *The connectivity graph $G$ does not contain any anti-hole.*

**Proof.** Throughout this proof, recall that $T$ is a tree, in particular, if two vertices are temporally connected, then there is a unique temporal path from one to the other. Assume by contradiction that $G$ contains an anti-hole $H$ with $k$ vertices. If $k = 5$, then $H$ is a hole, which contradicts Lemma 14; hence, assume $k \geq 6$.

When $k$ is odd, let $F_1 = ODD(H) \setminus \{u_k\}, F_2 = EVEN(H)$. When $k$ is even, let $F_1 = ODD(H), F_2 = EVEN(H)$. Observe that $|F_1| = |F_2| \geq 3$ and both sets induce (vertex-disjoint) cliques in $G$. By Lemma 10, there are temporal paths $P_1$ and $P_2$ in $\mathcal{T}$ containing $F_1$ and $F_2$, respectively, which we can assume are minimal vertex-inclusion-wise (so that, for each $i \in \{1, 2\}$, both end-vertices of $P_i$ lie in $F_i$). For $i \in \{1, 2\}$, let $v_i$ and $w_i$ denote the source and sink of $P_i$, respectively. We have two cases.

**Case 1: $V(P_1) \cap V(P_2) = \emptyset$.** Let $Q$ be the shortest temporal path that contains vertices from both $P_1$ and $P_2$ (note that, since every vertex in $F_1$ is temporally connected to at least one vertex in $F_2$, $Q$ necessarily exists). Let $p_1, p_2$ be the end-vertices of $Q$ that lie on $P_1$ and $P_2$, respectively. Since for each $i \in \{1, 2\}$ and $Z \in \{F_1, F_2\}$, $N_G(w_i) \cap Z \neq \emptyset$, $Q$ is oriented from $p_1$ to $p_2$, or vice versa. Without loss of generality, assume that $Q$ is oriented from $p_2$ to $p_1$. Then, necessarily $p_2 = w_2$, since otherwise $w_2$ is not temporally connected with any vertex of $F_1$, a contradiction. By a similar argument, we have $p_1 = v_1$. Now, consider the clique induced by $N_G(v_2) \cap F_1$. Due to Lemma 10, all vertices of $N_G(v_2) \cap F_1$ and $v_2$ itself

are contained in a temporal path, which also necessarily contains $w_2$. Hence all of $F_2$ ($P_2$, even) is in a temporal path containing $v_1$, since the path has to go through $v_1$ to reach other vertices of $F_1$, and so $F_2 \cup \{v_1\}$ forms a clique. This is a contradiction as $v_1$ necessarily has at least one non-neighbor in $F_2$.

**Case 2: $V(P_1) \cap V(P_2) \neq \emptyset$.** Let $Q$ denote the maximal vertex-inclusion-wise path that is common to both $P_1$ and $P_2$, *i.e.*, the path induced by the set $V(P_1) \cap V(P_2)$. Note that $Q$ does not contain any vertex from $H$, since a vertex of $H$ in $Q$ would be temporally connected to every other vertex of $F_1 \cup F_2$, a contradiction. Let $p$ denote source of $Q$ and for each $i \in \{1, 2\}$ let $Q_i$ (resp. $Q'_i$) be the subpath of $P_i$ between $p$ and $w_i$ (resp. $p$ and $v_i$).

Note that no vertex of $Q'_1 \setminus \{p\}$ can be in a directed path with any vertex of $Q'_2 \setminus \{p\}$. Similarly, no vertex of $Q_1 \setminus Q$ can be in a directed path with any vertex of $Q_2 \setminus Q$. Thus, the two subgraphs of the connectivity graph $G$ induced by the vertices of $(V(Q_1) \cup V(Q_2)) \setminus V(Q)$ and $(V(Q'_1) \cup V(Q'_2)) \setminus \{p\}$ each induce the complement of a complete bipartite graph. As $H$ does not contain any complement of a 4-cycle as an induced subgraph, this implies that there are exactly three vertices of $H$ in each of these two subsets of vertices (since $Q$ does not contain any vertex of $H$). In particular, $H$ has size either 6 or 7.

Without loss of generality, we assume that $Q'_1$ contains only one vertex of $H$, which must be $v_1$. Thus, there are two vertices of $H$ in $Q'_2$: $v_2$ and another vertex, say, $v'_2$. Since $F_1$ and $F_2$ both have size 3, the vertices of $H$ in $Q_1$ are $w_1$ and (say) $w'_1$, and the only vertex of $H$ in $Q_2$ is $w_2$. Now, observe that if $v_2$ is contained in a temporal path with $w_1$, then $v_2$, $v'_2$, $w'_1$ and $w_1$ are in a common temporal path. This is not possible, since in $H$, there is either one or two non-edges among these four vertices (depending on whether $H$ has size 7 or 6). Thus, $w_1$ and $v_2$ are in no common temporal path. Since $v_2$ has no non-neighbour in $H$ other than $v_1$ and $w_1$, $v_2$ and $w'_1$ are in a common temporal path, that also contains $v'_2$. Thus, $\{v_2, v'_2, w'_1\}$ form a clique in $H$. Similarly, $\{v'_2, w'_1, w_1\}$ also form a clique in $H$. If $H$ had size 6, $v'_2$ and $w'_1$ would need to be non-neighbours in $H$ (since $w_1$ already has two non-neighbours in $H$), a contradiction. Thus, $H$ has size 7, and the two non-neighbours in $H$ of $u_7$ (the vertex of $H$ not in $F_1 \cup F_2$) are $v'_2$ and $w'_1$ (since they are the only ones without two non-neighbours in $H$). But $u_7$ has to be temporally connected to all of $v_1$, $v_2$, $w_1$ and $w_2$, so $u_7$ has to be in $Q$. But any temporal path from $v_2$ to a vertex of $Q$ has to contain $v'_2$, and so $u_7$ and $v'_2$ are temporally connected, a contradiction. This completes the proof. ◀

## 4.3 Completion of the proof of Theorem 2

Lemmas 14, 17, and 18 imply that the connectivity graph of a temporal oriented tree is weakly chordal. Note that this cannot be strengthened to chordal, as there are temporal oriented trees whose connectivity graphs contain induced 4-cycles: let $\lambda(\overrightarrow{s_1c}) = \lambda(\overrightarrow{s_2c}) = 1$ and $\lambda(\overrightarrow{ct_1}) = \lambda(\overrightarrow{ct_2}) = 2$, the vertices $s_1$, $t_1$, $s_2$ and $t_2$ induce a $C_4$ in the connectivity graph. Corollary 11 implies the correspondence between a minimum temporal path cover of a temporal oriented tree and a minimum clique cover of the corresponding connectivity graph. We then conclude using Theorem 7 for the algorithm. Observation 9, Corollary 11 and Theorem 7 together give the Dilworth property.

# 5 Temporally Disjoint Path Cover on temporal oriented trees

We provide a reduction from Unary Bin Packing to prove the following (proof deferred to the full version [9] due to space constraints).

▶ **Theorem 3.** *Temporally Disjoint Path Cover is NP-hard on temporal oriented trees.*

We also show the following.

▶ **Proposition 19 (*).** *There are temporal oriented trees (whose underlying digraph is a star) that do not satisfy the TD-Dilworth property.*

## 6 Subclasses of temporal oriented trees

▶ **Theorem 4.** *Temporal Path Cover and Temporally Disjoint Path Cover can be solved in time:*
**(a)** $\mathcal{O}(\ell n)$ *on temporal oriented lines;*
**(b)** $\mathcal{O}(\ell n^2)$ *on temporal rooted directed trees;*
*where $\ell$ is the maximum number of labels per arc and $n$ is the number of vertices. Furthermore, both classes satisfy the TD-Dilworth property.*

**Proof.** (*a*) Let $\mathcal{P} = (P, \lambda)$ be a temporal oriented line, and let $v$ be a leaf of $P$. We construct $\mathcal{C}$ as follows. Assume that $v$ is incident with an in-arc $\overrightarrow{uv}$. We construct a maximum-length temporal path that covers $v$. Set $(b, c) = (u, v)$, $\ell = \max \lambda(\overrightarrow{uv})$, and apply the following routine: while $b$ is incident with an in-arc $\overrightarrow{ab}$, if there is a time label smaller than $\ell$ in $\lambda(\overrightarrow{ab})$, add $\overrightarrow{ab}$ to the path, update $(b, c) = (a, b)$ and $\ell = \max\{k \in \lambda(\overrightarrow{ab}) \mid k < \ell\}$. When the routine stops, add the path to $\mathcal{C}$, remove its vertices from $P$, and start again on a new leaf (or return $\mathcal{C}$ if $P$ is empty). If $v$ was incident with an out-arc, we would do the same but with out-arcs, start with the smallest possible time label, and update $\ell = \min\{k \in \lambda(\overrightarrow{ab}) \mid k > \ell\}$.

This algorithm computes its output in time $\mathcal{O}(\ell n)$: every arc is visited at most once, but we need to parse the time labels in order to see whether we can keep on extending the path or not. Furthermore, the set of leaves $v$ where we start the routine are a temporal antichain: assume on the contrary that $v_1$ and $v_2$ are such vertices that are temporally connected, and assume without loss of generality that there is a path from $v_1$ to $v_2$ in the underlying oriented path; in this case, our algorithm would have added $v_1$ to the path that started being computed at $v_2$, a contradiction. Hence, $\mathcal{C}$ is a temporally disjoint path cover with the same size as a temporal antichain, proving that it is minimum-size and that temporal oriented lines satisfy the TD-Dilworth property.

(*b*) We give an algorithm that solves Temporal Path Cover on a temporal rooted directed tree $\mathcal{T} = (T, \lambda)$. First, we sort the vertices of $T$ with respect to their topological distance from the root in $T$ (with the highest distances first). Then, we construct a maximum-length temporal path covering the first uncovered vertex (which will be a sink of that path), and repeat until $\mathcal{T}$ is fully covered.

Note that this algorithm outputs $\mathcal{C}$ which is clearly a temporal path cover: every vertex is covered by some path of $\mathcal{C}$. Furthermore, it is an adaptation of the algorithm for temporal oriented lines: instead of successive leaves, we construct the paths from successive leaves with highest topological distance from the root. We will show that $\mathcal{C}$ is minimum-size, and later we will explain how to modify the algorithm in order to obtain a minimum-size temporally disjoint path cover.

Let $S$ be the set of sinks of paths of $\mathcal{C}$. First, let $v_i$ and $v_j$ be two vertices of $S$ (without loss of generality, assume that $v_i$ was covered by the algorithm after $v_j$). They cannot be temporally connected, since otherwise, the graph being a temporal rooted directed tree, one of them is necessarily the predecessor of the other in a path from the root, and thus the maximum-length temporal path ending in $v_j$ would necessarily contain $v_i$, since there is a temporal path from $v_i$ to $v_j$, and thus $v_i$ would have been covered at this step and cannot

be in $S$. Hence, $S$ is a temporal antichain. Furthermore, since $|\mathcal{C}| = |S|$, $\mathcal{C}$ is minimum-size and $S$ is maximum-size, implying that temporal rooted directed trees satisfy the Dilworth property.

We now modify the algorithm to obtain a minimum-size temporally disjoint path cover. Indeed, we can see that the maximum-length temporal path construction, which is executed for every vertex of $S$, can re-cover some vertices that had already been covered at a previous step. Let $v_i$ and $v_j$ be two vertices of $S$ such that their maximum-length temporal paths constructed by the algorithm $P_i$ and $P_j$ intersect. Since the graph is a temporal rooted directed tree, we can divide $P_i$ and $P_j$ into the following parts, without loss of generality: $P_i = P_i^{\text{top}} \cup (P_i \cap P_j) \cup P_i^{\text{bot}}$ and $P_j = (P_i \cap P_j) \cup P_j^{\text{bot}}$, where $P_i^{\text{top}} \cap P_j = P_i^{\text{bot}} \cap P_j = P_j^{\text{bot}} \cap P_i = \emptyset$ (note that we can have $P_i^{\text{top}} = \emptyset$). In other words, $P_i^{\text{top}}$ (resp. $P_i^{\text{bot}}$) contains the vertices of $P_i \setminus P_j$ that are ancestors (resp. descendants) of $P_i \cap P_j$ in the underlying rooted tree, with the equivalent definition for $P_j^{\text{top}}$ and $P_j^{\text{bot}}$. Hence, we can modify the algorithm by adding a loop that, for each such pair $(P_i, P_j)$, defines these subpaths and then removes $P_i \cap P_j$ from $P_j$. Now, $\mathcal{C}$ will still be a temporal path cover, but the paths will be vertex-disjoint and thus temporally disjoint, and its size will not change. This implies that temporal rooted directed trees satisfy the TD-Dilworth property (contrasting the general temporal oriented trees), and thus the modified algorithm outputs the optimal solution for these two problems. The result of the algorithm and its modification is depicted in Figure 3.

Finally, one can check that the algorithm and its modification compute $\mathcal{C}$ in time $\mathcal{O}(\ell n^2)$. For each vertex in the antichain $S$, we have to construct the maximum-length temporal path. This can be done in time $\mathcal{O}(\ell n)$ by taking at every arc the largest label that allows to extend the path, thus we have to parse all the labels of every arc along the path, which can be of linear-size in the worst case. Since we can have a linear number of antichain vertices, we have a complexity of $\mathcal{O}(\ell n^2)$ to get the temporal path cover. The modification to make it temporally disjoint can be done in $\mathcal{O}(n^2)$ time afterwards, by considering all pairs of intersecting paths starting from the root. ◀

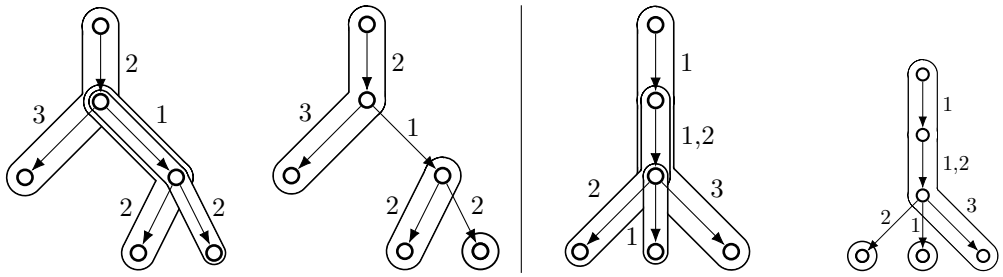



**Figure 3** Minimum-size temporal path covers and temporally disjoint path covers of the same temporal rooted directed tree (on the left, with one label per arc; on the right, with any labels per arc), as computed by our algorithm and its modification in the proof of Theorem 4.

## 7 Algorithms for temporal digraphs of bounded treewidth

Recall that an algorithm is FPT with respect to some parameter $k$ of the input, if it runs in time $f(k)n^{\mathcal{O}(1)}$ for inputs of size $n$, where $f$ is any computable function; the algorithm is XP for this parameter if the running time is in $n^{f(k)}$ [12]. We prove the following theorem.

▶ **Theorem 5.** *There is an algorithm for* Temporally Disjoint Path Cover *on general temporal digraphs that is FPT with respect to the treewidth of the underlying undirected graph and the maximum number of labels per arc. For* Temporal Path Cover *on general temporal digraphs, there is an XP algorithm for the same parameter.*

**Proof (sketch).** To prove the theorem, we use the well-known concept of *nice tree decompositions* [27]. The algorithm is a classic bottom-up dynamic programming. To simplify the algorithm, we replace each arc by a set of parallel arcs, each of them with a distinct time-label. This makes it easier to deal with temporally disjoint paths, as in this representation, one arc cannot be used in two solution paths (for Temporally Disjoint Path Cover).

To give the intuition behind the algorithm, we informally describe the states of the dynamic programming. For a bag $X_v \subseteq V(G)$ corresponding to a node $v$ of the tree decomposition, every state corresponds to a distinct way a potential solution interacts with $X_v$. Primarily, it consists of a set of (solution) subpaths that cover the vertices in $X_v$. Each subpath may consist of several disconnected parts (possibly, a part may have only one vertex). The order in which the vertices of each solution path appear is also specified. We also store the information, for every vertex, whether it is connected via an arc to one vertex (or two vertices) in its solution path, but lying outside the bag $X_v$, and whether this vertex lies in a lower (already handled) or upper (to be handled) part of the tree decomposition.

We show that this information is enough to encode a partial solution, and as there are at most $p = \binom{\mathrm{tw}}{2} \cdot t_{\max}$ arcs in each bag, we deduce that the maximum possible number of states for a bag is at most $2^{O(p \log p)}$ in the case of Temporally Disjoint Path Cover (since every arc may appear in at most one solution path) and $n^{O(p \log p)}$ in the case of Temporal Path Cover (since a specific subpath may appear in any number of solution paths, so we must also encode the number of appearances of each subpath). The running times are essentially dominated by these functions.

Due to space constraints, the details are deferred to the full version [9]. ◀

## 8 Conclusion

We have initiated the study of two natural path covering problems in temporal DAGs, which, in the static case, are related to Dilworth's theorem and are polynomial-time solvable. Both problems become NP-hard for temporal DAGs, even in a very restricted setting. Interestingly, and somewhat unexpectedly, they behave differently on temporal oriented trees: we showed that Temporal Path Cover is polynomial-time solvable on temporal oriented trees (and a temporal version of Dilworth's theorem holds in this setting), while Temporally Disjoint Path Cover remains NP-hard for this class. On the other hand, this distinction is inverted in the parameterized case, where we obtained an FPT algorithm for Temporally Disjoint Path Cover but an XP algorithm for Temporal Path Cover. However, we do not know if our algorithms for treewidth and number of time-steps are optimal. In particular, can we obtain an FPT algorithm for Temporal Path Cover for this parameter? One could also explore other (structural) parameterizations of the problems.

To prove our polynomial-time algorithm for Temporal Path Cover on temporal oriented trees, we have reduced the problem to Clique Cover in a static undirected graph, which turns out to be weakly chordal. This is a powerful technique, and the correspondence between the two problems is quite enlightening for the structure of temporal paths in an oriented tree. Nevertheless, it seems unlikely that this particular technique can be used on temporal digraph classes that are far from trees, as it was essential for the proof that any two vertices are joined by only one path in the underlying tree. However, this general technique could likely be applied in other temporal settings.

We note that many of our results for Temporally Disjoint Path Cover also hold for its stricter vertex-disjoint version (note that a vertex-disjoint version of Temporally Disjoint Paths is studied in [24]), in particular, the NP-hardness result for restricted DAGs and the polynomial-time algorithms for rooted directed trees and oriented lines.

## References


1 Eleni C. Akrida, George B. Mertzios, Sotiris E. Nikoletseas, Christoforos L. Raptopoulos, Paul G. Spirakis, and Viktor Zamaraev. How fast can we reach a target vertex in stochastic temporal graphs? *J. Comput. Syst. Sci.*, 114:65–83, 2020.

2 Eleni C. Akrida, George B. Mertzios, and Paul G. Spirakis. The temporal explorer who returns to the base. In Pinar Heggernes, editor, *Algorithms and Complexity - 11th International Conference, CIAC 2019, Rome, Italy, May 27-29, 2019, Proceedings*, volume 11485 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2019.

3 Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *Int. J. Found. Comput. Sci.*, 14(2):267–285, 2003.

4 Manuel Cáceres. Minimum chain cover in almost linear time. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 31:1–31:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

5 Manuel Cáceres, Massimo Cairo, Brendan Mumey, Romeo Rizzi, and Alexandru I. Tomescu. Sparsifying, shrinking and splicing for minimum path cover in parameterized linear time. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 359–376. SIAM, 2022.

6 Manuel Cáceres, Brendan Mumey, Edin Husić, Romeo Rizzi, Massimo Cairo, Kristoffer Sahlin, and Alexandru I. Tomescu. Safety in multi-assembly via paths appearing in all path covers of a DAG. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(6):3673–3684, 2022.

7 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distributed Syst.*, 27(5):387–408, 2012.

8 Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021.

9 Dibyayan Chakraborty, Antoine Dailly, Florent Foucaud, and Ralf Klasing. Algorithms and complexity for path covers of temporal DAGs. *arXiv preprint arXiv:2403.04589*, 2024.

10 Yangjun Chen and Yibin Chen. On the graph decomposition. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, pages 777–784, 2014.

11 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

12 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4(8). Springer, 2015.

13 Pradipta Kumar Das, Himansu Sekhar Behera, Prabir Kumar Jena, and Bijaya K. Panigrahi. An intelligent multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *Int. J. Autom. Comput.*, 18(6):1032–1044, 2021.

14 Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51:161–166, 1950.

15 Jeff Erickson. *Algorithms*. self-published, 2019. URL: `http://jeffe.cs.illinois.edu/teaching/algorithms/`.

16 D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

17 Delbert R Fulkerson. Note on Dilworth's decomposition theorem for partially ordered sets. In *Proceedings of the American Mathematical Society*, volume 7(4), pages 701–702, 1956.

18 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., NLD, 2004.

19 Ryan B. Hayward. Weakly triangulated graphs. *J. Comb. Theory, Ser. B*, 39(3):200–208, 1985. doi:10.1016/0095-8956(85)90050-4.

20 Ryan B. Hayward, Jeremy P. Spinrad, and R. Sritharan. Improved algorithms for weakly chordal graphs. *ACM Trans. Algorithms*, 3(2):14, 2007. doi:10.1145/1240233.1240237.

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

22 H. V. Jagadish. A compression technique to materialize transitive closure. *ACM Transactions on Database Systems*, 15(4):558–598, 1990.

23 Naoyuki Kamiyama and Yasushi Kawase. On packing arborescences in temporal networks. *Inf. Process. Lett.*, 115(2):321–325, 2015.

24 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002.

25 Nina Klobas, George B. Mertzios, Hendrik Molter, Rolf Niedermeier, and Philipp Zschoche. Interference-free walks in time: temporally disjoint paths. *Autonomous Agents and Multi Agent Systems*, 37(1):1, 2023.

26 Nina Klobas, George B Mertzios, Hendrik Molter, Rolf Niedermeier, and Philipp Zschoche. Interference-free walks in time: Temporally disjoint paths. *Autonomous Agents and Multi-Agent Systems*, 37(1):1, 2023.

27 T. Kloks. *Treewidth, Computations and Approximations*. Springer, 1994.

28 Shimon Kogan and Merav Parter. Faster and unified algorithms for diameter reducing shortcuts and minimum chain covers. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 212–239. SIAM, 2023.

29 Pascal Kunz, Hendrik Molter, and Meirav Zehavi. In which graph structures can we efficiently find temporally disjoint paths and walks? In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 180–188. ijcai.org, 2023.

30 L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(3):253–267, 1972.

31 Veli Mäkinen, Alexandru I. Tomescu, Anna Kuosmanen, Topi Paavilainen, Travis Gagie, and Rayan Chikhi. Sparse dynamic programming on dags with small width. *ACM Trans. Algorithms*, 15(2), February 2019.

32 Loris Marchal, Hanna Nagy, Bertrand Simon, and Frédéric Vivien. Parallel scheduling of DAGs under memory constraints. In *2018 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2018, Vancouver, BC, Canada, May 21-25, 2018*, pages 204–213. IEEE Computer Society, 2018.

33 Andrea Marino and Ana Silva. Eulerian walks in temporal graphs. *Algorithmica*, 85(3):805–830, 2023.

34 George B. Mertzios, Hendrik Molter, Malte Renken, Paul G. Spirakis, and Philipp Zschoche. The complexity of transitively orienting temporal graphs. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 75:1–75:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

35 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. In Christos D. Zaroliagis, Grammati E. Pantziou, and Spyros C. Kontogiannis, editors, *Algorithms, Probability, Networks, and Games - Scientific Papers and Essays Dedicated to Paul G. Spirakis on the Occasion of His 60th Birthday*, volume 9295 of *Lecture Notes in Computer Science*, pages 308–343. Springer, 2015.

36 Simeon C. Ntafos and S. Louis Hakimi. On path cover problems in digraphs and applications to program testing. *IEEE Transactions on Software Engineering*, SE-5(5):520–529, 1979.

37 Jari Saramäki and Petter Holme, editors. *Temporal Network Theory*. Computational Social Sciences. Springer, Germany, October 2019.

38 Jeremy P. Spinrad and R. Sritharan. Algorithms for weakly triangulated graphs. *Discret. Appl. Math.*, 59(2):181–191, 1995. doi:10.1016/0166-218X(93)E0161-Q.

39 Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Pavel Surynek and William Yeoh, editors, *Proceedings of the Twelfth International Symposium on Combinatorial Search, SOCS 2019, Napa, California, 16-17 July 2019*, pages 151–159. AAAI Press, 2019.

40 Dawei Sun, Jingkai Chen, Sayan Mitra, and Chuchu Fan. Multi-agent motion planning from signal temporal logic specifications. *IEEE Robotics Autom. Lett.*, 7(2):3451–3458, 2022.

41 Huanhuan Wu, James Cheng, Yiping Ke, Silu Huang, Yuzhen Huang, and Hejun Wu. Efficient algorithms for temporal path computation. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2927–2942, 2016.

42 Haifeng Xu, Fei Fang, Albert Xin Jiang, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. Solving zero-sum security games in discretized spatio-temporal domains. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1500–1506. AAAI Press, 2014.

43 Yue Yin and Bo An. Efficient resource allocation for protecting coral reef ecosystems. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 531–537. AAAI Press, 2016.

44 Youzhi Zhang, Bo An, Long Tran-Thanh, Zhen Wang, Jiarui Gan, and Nicholas R. Jennings. Optimal escape interdiction on transportation networks. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3936–3944. ijcai.org, 2017.