



# Exact and approximation algorithms for covering timeline in temporal graphs

Riccardo Dondi<sup>1</sup> · Alexandru Popa<sup>2</sup>

Received: 25 May 2023 / Accepted: 8 April 2024  
© The Author(s) 2024

## Abstract

We consider a variant of vertex cover on temporal graphs that has been recently defined for summarization of timeline activities in temporal graphs. The problem has been proved to be NP-hard, even for several restrictions of the time domain and vertex degree. We present novel algorithmic contributions for the problem and we give an approximation algorithm of factor  $O(T \log n)$ , on a temporal graph of  $T$  timestamps and  $n$  vertices. We focus then on the NP-hard restriction of the problem, where at most one temporal edge is defined in each timestamp. For this restriction we present a  $4(T - 1)$  approximation algorithm and a parameterized algorithm (a reduction to kernel) for parameter the cost, called span, of the solution.

**Keywords** Timeline cover · Temporal graph · NP-hard problem · Approximation algorithm · FPT algorithm

## 1 Introduction

Novel representations of entity interactions have been considered in network science and graph literature, in order to take into account their dynamics and heterogeneity. This has led to the definition of novel graph models, a notable example being *temporal networks* (Holme, 2015; Michail, 2016; Holme & Saramäki, 2019). Temporal networks or temporal graphs represent how interactions (or edges) evolve in a discrete time domain for a given set of entities (or vertices) (Holme, 2015; Kempe et al., 2002). The time domain consists of a sequence of timestamps and, for each timestamp of the considered time domain, a temporal graph defines a static graph (also called snapshot) on the same vertex set. Thus a temporal

---

Alexandru Popa has contributed equally to this work.

---

A preliminary version of this paper appeared in Dondi and Popa (2023).

---

✉ Riccardo Dondi  
riccardo.dondi@unibg.it

✉ Alexandru Popa  
alexandru.popa@fmi.unibuc.ro

<sup>1</sup> Dipartimento di Lettere, Filosofia, Comunicazione, Università degli Studi di Bergamo, Bergamo, Italy

<sup>2</sup> Department of Computer Science, University of Bucharest, Bucharest, Romania

graph can be seen as sequence of static graphs, one for each timestamp, over the same vertex set, while the edge sets can change from one timestamp to the other. In particular, an edge observed in a timestamp is called a *temporal edge*.

The main aspects of temporal graphs considered in the literature have been finding paths and studying their connectivity (Kempe et al., 2002; Wu et al., 2014, 2016; Erlebach et al., 2021; Zschoche et al., 2020; Fluschnik et al., 2020; Akrida et al., 2021; Bumpus & Meeks, 2023; Marino & Silva, 2021), but other problems have been studied, for example dense subgraph discovery (Rozenshtein et al., 2020; Dondi & Hosseinzadeh, 2021). A fundamental problem in computer science that has been recently considered for temporal graphs is Vertex Cover (Akrida et al., 2020; Rozenshtein et al., 2021). In this contribution we will consider a variant of Vertex Cover on temporal graphs that has been introduced for interaction timeline summarization, a problem called MinTCover (Rozenshtein et al., 2021). The problem considers a sequence of observed interactions between entities (for example, users of a social platform) and aims to explain these interactions with intervals of entity activities that, following a parsimony approach, have minimum length. From a graph theory perspective, the MinTCover problem is defined over a temporal graph, whose vertices represents entities and whose temporal edges represent interactions between entities. MinTCover then asks, for each vertex, for the definition of a temporal interval where the vertex is considered to be active such that each temporal edge  $e$  is covered, that is at least one of its endpoints has an interval activity that includes the timestamp where  $e$  is present. The length of an interval activity of a vertex is called the *span* of the vertex. The objective function of MinTCover asks to minimize the sum of the vertex spans.

Notice that an alternative definition of the span of a vertex could be the number of timestamps where the vertex is active. This definition leads to a problem that is more similar to Vertex Cover (when the time domain consists of a single timestamp the problem is exactly Vertex Cover). On the other hand, the definition we consider, defined in Rozenshtein et al. (2021), can be seen as a temporal variant of Vertex Cover above a guarantee, a problem that has been studied in the literature (Garg & Philip, 2016). In Vertex Cover above a guarantee, given a graph  $G$  the problem asks whether there exists a solution consisting of at most  $k + l(G)$  vertices, where  $l(G)$  is a lower bound on the size of a minimum vertex cover of  $G$ , like the size of a maximum matching or the minimum value of a linear programming relaxation of Vertex Cover. This leads to a challenging problem, as  $k$  is smaller than the size of a vertex cover and, in most of the cases, also than  $l(G)$ . Thus for example the strategy of branching on the endpoints of an edge, in order to compute which one is in a vertex cover, cannot be applied. Similarly, in our case, since the span of a vertex active in a single timestamp is equal to 0, we have that a vertex that contributes 0 to the value of the objective function may cover temporal edges. Thus also in this case strategies like that of branching on the endpoints of a (temporal) edge cannot be used. This has lead to challenging analysis of the problem, for example for its parameterized tractability (Froese et al., 2023; Dondi & Lafond, 2023), where the approaches for Vertex Cover cannot be applied.

MinTCover is NP-hard (Rozenshtein et al., 2021), also for some restrictions: when each timestamp contains at most one temporal edge (Dondi, 2023), a restriction we denote by 1-MinTCover, when each vertex has degree at most two in each timestamp and the temporal graph is defined over three timestamps (Dondi, 2023), and when the temporal graph is defined over two timestamps (Froese et al., 2023).

MinTCover has been considered also in the parameterized complexity framework. When the temporal graph is defined over two timestamps, MinTCover admits a parameterized algorithm for parameter the span of a solution (Froese et al., 2023).

Two results given in Froese et al. (2023) can be applied also for the approximation complexity of MinTCover. A lower bound on the approximability of the problem can be proved by observing that the reduction from the Odd Cycle Transversal problem<sup>1</sup> to MinTCover on two timestamps presented in Froese et al. (2023) is indeed an approximation preserving reduction. Since assuming the Unique Games Conjecture Odd Cycle Transversal is known to be not approximable within constant factor [?], the following result holds.

**Theorem 1** (Froese et al., 2023) *MinTCover cannot be approximated within constant factor, assuming the Unique Games Conjecture, even on temporal graphs defined over two timestamps.*

The authors of Froese et al. (2023) give a parameterized algorithm, when the problem is parameterized by the span of a solution and the temporal graph is defined over two timestamps. This parameterized algorithm is based on a parameterized reduction from MinTCover on a time domain of two timestamps to the Almost 2-SAT problem,<sup>2</sup>. This reduction can be easily modified so that it is an approximation preserving reduction from MinTCover on two timestamps to Almost 2-SAT. Since Almost 2-SAT is approximable within factor  $O(\sqrt{\log n})$ , for a graph having  $n$  vertices (Agarwal et al., 2005), the following result holds.

**Theorem 2** (Froese et al., 2023) *MinTCover on two timestamps can be approximated within factor  $O(\sqrt{\log n})$ , for a temporal graph having  $n$  vertices.*

### Related Works.

In Rozenshtein et al. (2021) different variants of the MinTCover problem have been introduced, considering the cases that vertex activity is defined in one interval (as for MinTCover) or in more than one interval (but in a bounded number), and that the objective function asks for the minimization of (1) the sum of vertex spans (as for MinTCover) or (2) the maximum activity span of a vertex.

The computational complexity of the variants of the MinTCover problem has been analyzed recently. Unlike MinTCover, when the vertex activity is defined in one interval and the objective function is the minimization of the maximum activity span of a vertex, the problem admits a polynomial-time algorithm (Rozenshtein et al., 2021). The variants where the vertex activity is defined as  $k \geq 2$  intervals are NP-hard for both objective functions considered (Rozenshtein et al., 2021) and they also do not admit any polynomial-time approximation algorithm, as deciding whether there exists a solution of span equal to 0 is already an NP-complete problem (Rozenshtein et al., 2021).

In Froese et al. (2023) the parameterized complexity of the variants of MinTCover has been explored, considering as parameters the number of vertices of the temporal graph, the length of the time domain, the number of intervals of vertex activity and the span of a solution.

Two other variants of Vertex Cover in temporal graphs have been considered in Akrida et al. (2020); Hamm et al. (2022). A first variant, given a temporal graph, asks for the minimum number of pairs  $(u, t)$ , where  $u$  is a vertex and  $t$  is a timestamp, such that each non-temporal edge  $e = \{a, b\}$  is temporally covered, that is there exists a timestamp  $t$  where  $e$  is present and at least one of  $(a, t)$  and  $(b, t)$  belongs to the cover. A second variant asks for each temporal edge to be temporally covered at least once for every interval of a given length  $\Delta$ .

<sup>1</sup> We recall that the Odd Cycle Transversal problem, given a graph, asks for the removal of the minimum number of edges such that the resulting graph is bipartite.

<sup>2</sup> We recall that the Almost 2-SAT given a formula consisting of clauses on two literals, asks for the removal of the minimum number of clauses so that the resulting formula is satisfiable.

### Our results.

In this paper, we give algorithmic contributions for  $\text{MinTCover}$ , in particular for its approximability, and for  $1\text{-MinTCover}$  (the restriction of  $\text{MinTCover}$  where at most one temporal edge is present in each timestamp, a restriction also known to be NP-hard (Dondi, 2023)). First, we present in Sect. 3 a randomized polynomial-time approximation algorithm for  $\text{MinTCover}$  of factor  $O(T \log n)$ , for a temporal graph defined over  $T$  timestamps and  $n$  vertices. Then in Sect. 4 we focus on  $1\text{-MinTCover}$  and we provide a parameterized algorithm that reduces an instance to a polynomial kernel, where the parameter is the span of the solution, and a polynomial-time approximation algorithm of factor  $4(T - 1)$ . In the next section we introduce the main concepts related to  $\text{MinTCover}$  and we formally define the  $\text{MinTCover}$  problem.

## 2 Preliminaries

We start this section by introducing the definition of discrete time domain over which a temporal graph is defined. A temporal graph is defined over a sequence  $\mathcal{T}$  of timestamps between 1 and  $T$ , denoted by

$$\mathcal{T} = [1 \dots, T].$$

In what follows,  $T$  denotes the number of timestamps over which the temporal graph is defined.

An interval  $I = [i, j]$ , with  $1 \leq i \leq j \leq T$ , is the sequence of timestamps with value between  $i$  and  $j$ ; notice that if  $i = j$ , then  $I = [i, j] = [i, i]$  denotes an interval consisting of a single timestamp.

We present now the definition of temporal graph, where recall that the vertex set is not changing in the time domain (an example is presented in Fig. 1).

**Definition 1** A temporal graph  $G$  is a triple  $G = (V, E, \mathcal{T})$ , where  $V, E, \mathcal{T}$  are defined as follows:

1.  $V$  is a set of  $n$  vertices,
2.  $\mathcal{T} = [1, 2, \dots, T]$  is a time domain consisting of  $T$  timestamps
3.  $E \subseteq V \times V \times \mathcal{T}$  is a set  $m$  temporal edges, where a temporal edge of  $G$  is a triple  $\{u, v, t\}$ , with  $u, v \in V$  and  $t \in \mathcal{T}$ .

Given an interval  $I$  of  $\mathcal{T}$ ,  $E_I$  denotes the set of active edges in the timestamps of  $I$ , that is:

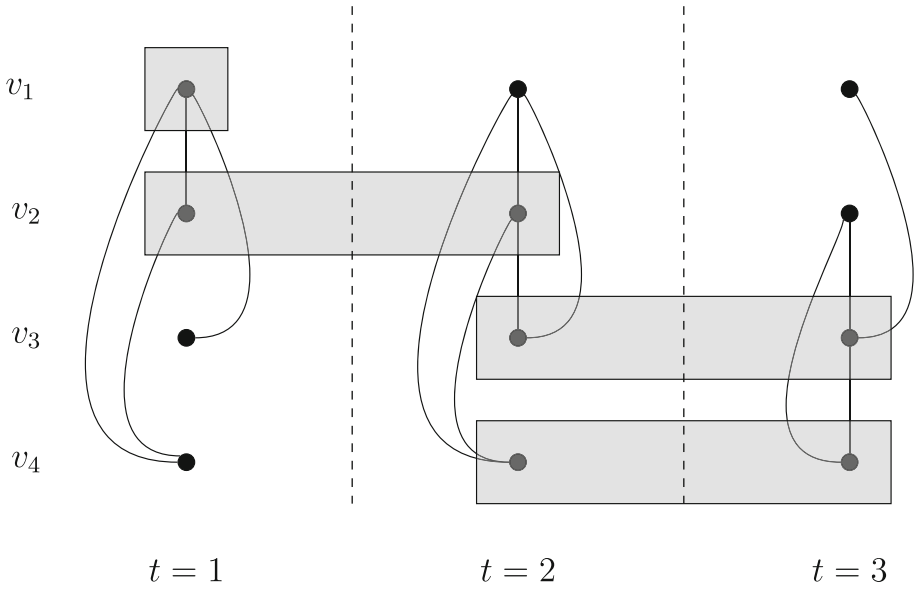
$$E_I = \{\{u, v, t\} \mid \{u, v, t\} \in E \wedge t \in I\}.$$

$E_t$  denotes the set of active edges in timestamp  $t$ .

Given a vertex  $v \in V$  and  $l_v, r_v \in \mathcal{T}$ , with  $l_v < r_v$ , an activity interval of  $v$  is defined as an interval

$$I(v) = [l_v, r_v]$$

of the time domain where  $v$  is considered active, while in any timestamp not in  $I(v)$ ,  $v$  is considered inactive. Notice that if  $I(v) = [l_v, r_v]$  is an activity interval of  $v$ , there may exist temporal edges  $\{u, v, t\}$ , with  $t < l_v$  or  $t > r_v$ . For example in Fig. 1  $I(v_1) = [1, 1]$ , while there exist temporal edges outside  $I(v_1)$ , more precisely  $\{v_1, v_2, 2\}$ ,  $\{v_1, v_3, 2\}$ ,  $\{v_1, v_4, 2\}$  and  $\{v_1, v_3, 3\}$ .



**Fig. 1** An example of MinTCover on a temporal graph  $G$  consisting of vertex set  $\{v_1, v_2, v_3, v_4\}$  and time domain  $T = [1, 2, 3]$ . For each timestamp  $t$ , we represent the temporal edges defined in that timestamp. For  $t = 1$ , the temporal edges are  $\{v_1, v_2, 1\}$ ,  $\{v_1, v_3, 1\}$ ,  $\{v_1, v_4, 1\}$ ,  $\{v_2, v_4, 1\}$ . The activity interval of each vertex is represented in gray: vertex  $v_1$  is active in interval  $[1, 1]$ , with span equal to 0, vertex  $v_2$  is active in interval  $[1, 2]$ , with span equal to 1, vertices  $v_3$  and  $v_4$  are active in interval  $[2, 3]$ , each one with span equal to 1. Hence the overall span is equal to 3

An activity timeline  $\mathcal{A}$  is a set of activity interval, defined as:

$$\mathcal{A} = \{I_{\mathcal{A}}(v) : v \in V\},$$

that is it contains an activity interval, denoted by  $I_{\mathcal{A}}(v)$ , for each vertex in  $v \in V$ .

Given a temporal graph  $G = (V, E, T)$ , a timeline  $\mathcal{A}$  covers  $G = (V, E, T)$  if for each temporal edge  $\{u, v, t\} \in E$ ,  $t$  belongs to  $I_{\mathcal{A}}(u)$  or to  $I_{\mathcal{A}}(v)$ .

The *span*  $s(I(v))$  of an interval  $I(v) = [l_v, r_v]$ , for some  $v \in V$ , is defined as follows:

$$s(I(v)) = |r_v - l_v|.$$

Notice that for an interval  $I(v) = [l_v, r_v]$  consisting of a single timestamp, that is where  $l_v = r_v$ , it holds that  $s(I(v)) = 0$ . The overall span of an activity timeline  $\mathcal{A}$  is equal to

$$s(\mathcal{A}) = \sum_{I_{\mathcal{A}}(v) \in \mathcal{A}} s(I_{\mathcal{A}}(v)).$$

Now, we are ready to define the problem we are interested into (see the example of Fig. 1).

**Problem 1 (MinTCover)**

**Input:** A temporal graph  $G = (V, E, T)$ .

**Output:** An activity timeline of minimum span that covers  $G$ .

We introduce now two definitions of degree of a vertex, *local degree*, defined in each timestamp, and *global degree*, defined in the whole temporal graph.

Given a temporal graph  $G = (V, E, T)$  and a vertex  $v \in V$ , the local degree of  $v$  in a timestamp  $t$ , denoted by  $\Delta_L(v, t)$ , is equal to the number of temporal edges incident in  $v$  at timestamp  $t$ , that is

$$\Delta_L(v, t) = |\{\{u, v, t\} : v \in V\}|.$$

The local degree  $\Delta_L$  of  $G$  is the maximum over  $v$  and  $t$  of  $\Delta_L(v, t)$ . The global degree  $\Delta(v)$  of a vertex  $v \in V$  is the number of temporal edges incident in  $v$  in the overall time domain, that is

$$\Delta(v) = \sum_{t=1}^T \Delta_L(v, t).$$

The global degree  $\Delta$  of  $G$  is the maximum over  $v$  of  $\Delta(v)$ .

Consider a set  $V_1 \subseteq V$  of vertices and an activity timeline  $\mathcal{A}_1$  for  $V_1$ . Given a set  $V_2 \subseteq V$ , an activity timeline  $\mathcal{A}_2$  for  $V_2$  is in *agreement* with  $\mathcal{A}_1$  if, for each  $v \in V_1 \cap V_2$ , it holds that

$$I_{\mathcal{A}_1}(v) = I_{\mathcal{A}_2}(v)$$

that is the activity interval of  $v$  in  $\mathcal{A}_1$  is identical to the activity interval of  $v$  in  $\mathcal{A}_2$ . Furthermore, when  $\mathcal{A}_2$  is in agreement with  $\mathcal{A}_1$ , we define as

$$\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$$

the activity timeline of  $V_1 \cup V_2$  that is in agreement with both  $\mathcal{A}_1$  and  $\mathcal{A}_2$ .

Given a temporal graph  $G = (V, E, T)$ , it is possible to compute in polynomial time whether there exists a solution of MinTCover on  $G$  of span 0.

**Lemma 1** (Rozenstein et al., 2021) *Let  $G$  be an instance of MinTCover. We can compute in polynomial time whether there exists a solution of MinTCover on  $G$  that has span equal to 0.*

**Proof** In Rozenstein et al. (2021) it is shown that, given a temporal graph  $G = (V, E, T)$  and a value  $h \in \mathbb{N}$ , it is possible to compute in polynomial time whether there exists an activity timeline  $\mathcal{A}$  that covers  $G$  such that, for each  $v \in V$ ,  $|I_{\mathcal{A}}(v)| \leq h$ . When  $h = 0$ , then  $|I_{\mathcal{A}}(v)| = 0$  for each  $v \in V$ , hence it is possible to check in polynomial time whether there exists an activity timeline of span equal to 0, thus the lemma holds.  $\square$

Given a temporal graph  $G = (V, E, T)$ , we can associate a labeled static graph, called *union graph*, where all the temporal edges are represented.

**Definition 2** Given a temporal graph  $G = (V, E, T)$ , the union graph associated with  $G$  is a labeled graph  $G_U = (V, E_U, \lambda)$ , where  $E_U = \{\{u, v\} : \{u, v, t\} \in E, \text{ for some } t \in T\}$  and  $\lambda \subseteq (E_U \times T)$  is a labeling of the edges in  $E_U$  defined as  $\lambda(\{u, v\}) = \{t : \{u, v, t\} \in E, \text{ for some } t \in T\}$ .

In the paper we consider a variant of MinTCover, denoted by 1-MinTCover, when there exists at most one active temporal edge in each timestamp.

## 2.1 Preprocessing a temporal graph

We present a preprocessing procedure of a temporal graph and the corresponding union graph that allows to remove some easy to cover parts. The preprocessing consists of two phases, we start by describing the first phase.

Phase 1: while there exists a vertex  $u$  with global degree 1, remove  $u$  and the single temporal edge  $\{u, v, t\}$  incident in  $u$  from  $G$  (and  $G_U$ );  $u$  is defined to be active in  $t$  (that is  $u$  covers  $\{u, v, t\}$ ) with span 0.

**Lemma 2** Consider a temporal graph  $G$  and the temporal graph  $G'$  obtained after Phase 1. Then there exists a solution of MinTCover on  $G$  having span at most  $k$  if and only if there exists a solution of MinTCover on  $G'$  having span at most  $k$ .

**Proof** Consider a solution of MinTCover on  $G'$  having span equal to  $h \leq k$ . Since the subgraph removed by Phase 1 can be covered with span 0, it follows that there exists a solution of MinTCover on  $G$  having span equal to  $h \leq k$ .

Consider a solution  $\mathcal{A}$  of MinTCover on  $G$  having span equal to  $h \leq k$ . Consider the subgraph  $G'$  of  $G$ , induced by the set  $V'$  of vertices not removed by Phase 1. Then  $\mathcal{A}$  defines a span equal to  $z \leq h$  for  $G'$ , thus concluding the proof.  $\square$

Next, we present the second phase of preprocessing.

Phase 2: While there exists a connected component  $C$  of  $G_U$  that is a simple cycle such that each edge in  $C$  has exactly one label, remove  $C$  from  $G_U$  (and the vertices and temporal edges corresponding to  $C$  from  $G$ ) and compute a solution of MinTCover for  $C$  having span equal to 0, by covering each temporal edge with one endpoint of span 0.

**Lemma 3** Consider a temporal graph  $G$  and the temporal graph  $G'$  obtained after Phase 2. Then there exists a solution of MinTCover on  $G$  having span at most  $k$  if and only if there exists a solution of MinTCover on  $G'$  having span at most  $k$ .

**Proof** Consider a solution  $\mathcal{A}'$  of MinTCover on  $G'$  having span equal to  $h \leq k$ . Consider the set  $\mathcal{C}$  of connected components of  $G$  that are simple cycles such that each of their edges has exactly one label. For each  $C \in \mathcal{C}$  with  $C = v_{C,1}, v_{C,2}, \dots, v_{C,z}, v_{C,1}$ , we define an activity timeline  $\mathcal{A}_C$  that has span 0 and covers  $C$  as follows:  $v_{C,j}$ ,  $1 \leq j \leq z-1$ , covers the temporal edge  $\{v_{C,j}, v_{C,j+1}, t_j\}$ , vertex  $v_{C,z}$  covers the the temporal edge  $\{v_{C,z}, v_{C,1}, t_z\}$ . Since each vertex of  $C$  is active in a single time stamp in  $\mathcal{A}_C$ , then this activity timeline has a span of 0. Then

$$\mathcal{A} = \mathcal{A}' \bigcup_{C \in \mathcal{C}} \mathcal{A}_C$$

is a solution of MinTCover on  $G$  having span equal to  $h \leq k$ .

For the other direction, as for the proof of Lemma 2, a solution  $\mathcal{A}$  of MinTCover on  $G$  having span equal to  $h \leq k$  defines an activity timeline that covers  $G'$  and has span at most  $h$ , thus concluding the proof.  $\square$

### 3 An approximation algorithm for MinTCover

In this section we present an approximation algorithm for MinTCover. First, we recall that by Lemma 1 it is possible to compute in polynomial time whether there exists a solution of MinTCover having span 0, so in what follows we assume that an optimal solution of MinTCover on instance  $G$  requires a span greater than 0. We give an  $O(T \log n)$  randomized approximation algorithm for the MinTCover problem (recall that  $n$  is the number of vertices of  $G$  and  $T$  is the number of timestamps).

The approximation algorithm consists of two phases:

1. First (Sect. 3.1) it computes an approximated solution that covers the temporal edges having at least three occurrences
2. It then computes an approximated solution of the remaining part of the temporal graph (Sect. 3.2).

### 3.1 Temporal edges with at least three occurrences

In the first phase, the approximation algorithm considers the following subgraph  $G'_U = (V, E'_U)$  of the union graph  $G_U$  (associated with  $G$ ) that contains edges with at least three labels.  $E'_U$  is then defined as follows:

$$E'_U = \{\{u, v\} : \exists\{u, v, t_1\}, \{u, v, t_2\}, \{u, v, t_3\} \in E, \text{ with } 1 \leq t_1 < t_2 < t_3 \leq T\}.$$

Now, consider  $G'_U = (V, E'_U)$  and compute in polynomial time a vertex cover  $V'_U \subseteq V$  of  $G'_U$ , by applying a factor 2 approximation algorithm for Vertex Cover (for example with the approximation algorithm given in Karakostas (2009)). We define an activity  $\mathcal{A}_1$  for the of vertices in  $V'_U$ , defining the following activity intervals:

$$I_{\mathcal{A}_1}(v) = [1, T], \text{ for each } v \in V'_U.$$

We prove now the following result on  $\mathcal{A}_1$ .

**Lemma 4** *Consider the set of vertices  $V'_U$  and an optimal solution  $\mathcal{A}^*$  of MinTCover on instance  $G$ . Then, it holds that (1) every temporal edge with an endpoint in  $V'_U$  is covered by  $\mathcal{A}_1$  and (2) it holds that*

$$s(\mathcal{A}_1) \leq 2(T - 1)s(\mathcal{A}^*).$$

**Proof** (1) The first part of the lemma follows by construction, since  $V'_U$  covers each vertex of  $G'_U$  and each vertex in  $V'_U$  is defined to be active by  $\mathcal{A}_1$  in each timestamp, thus  $\mathcal{A}_1$  covers every temporal edge of  $G$  with an endpoint in  $V'_U$ .

(2) Consider a minimum vertex cover  $V_U^*$  of  $G'_U$ . By construction of the approximation algorithm it holds that

$$s(\mathcal{A}_1) \leq (T - 1)|V'_U| \leq 2(T - 1)|V_U^*|. \quad (1)$$

Next, we claim that

$$|V_U^*| \leq s(\mathcal{A}^*). \quad (2)$$

Define the set  $V_a$  of vertices:

$$V_a = \{x \in V : x \text{ has a span greater than } 0 \text{ in } \mathcal{A}^*\}.$$

By construction, for each  $\{u, v\} \in E'_U$ , there exist at least three temporal edges  $\{u, v, t\}$  in  $G$ , thus at least one of  $u, v$  has a span greater than 0 in a solution of MinTCover on instance  $G$ . Then  $V_a$  is a vertex cover of  $G'_U$ , thus, since  $V_U^*$  is a minimum vertex cover of  $G'_U$ , it holds that

$$|V_U^*| \leq |V_a|.$$

and Eq. 2 holds. Combining Eqs. 1 and 2, we can conclude that

$$s(\mathcal{A}_1) \leq (T - 1)|V'_U| \leq 2(T - 1)|V_U^*| \leq 2(T - 1)s(\mathcal{A}^*).$$

thus also the second part of the lemma holds.  $\square$



$$\begin{aligned}
& \text{minimize } \sum_{v \in V} \left( \sum_{t=1}^T x_v^t - 1 \right) \\
& \text{subject to } \sum_{t=1}^T x_v^t \geq 1 && \forall v \in V \\
& x_v^t + x_u^t \geq 1 && \forall \{u, v, t\} \in E \\
& x_v^t \in \{0, 1\} && \forall v \in V, \forall t \in \{1, 2, \dots, T\}
\end{aligned}$$

**Fig. 2** ILP formulation for the timeline cover problem for the Min-NC-TCover problem

Now, after the computation of  $\mathcal{A}_1$ , the approximation algorithm removes from  $G$  the vertices in  $V'_U$  and the temporal edges covered by  $V'_U$  (hence all the temporal edges incident in one vertex of  $V'_U$ ). Notice that in the resulting temporal graph every two vertices  $u, v$  are connected by at most two temporal edges, otherwise one of  $u$  and  $v$  is in  $V'_U$  and the temporal edges that connect  $u$  and  $v$  are removed. Hence the number of temporal edges in the resulting temporal graph is bounded by  $2\binom{n}{2} \leq n^2$ . We present now the second phase of the approximation algorithm.

### 3.2 Temporal edges with at most two occurrences

We consider now a temporal graph  $G$ , where each pair of vertices is connected by at most two temporal edges. We give an approximation algorithm for this case. Notice that the number of temporal edges is bounded by  $2\binom{n}{2} \leq n^2$ .

The approximation algorithm is based on the randomized rounding technique and it is inspired by the approximation algorithm for Set Cover (Hochbaum, 1982).

First of all, we present an ILP formulation to model the following variant of the problem, called Minimum Non-Consecutive Timeline Cover (Min-NC-TCover), where: (1) each vertex can be active in several non necessarily consecutive timestamps and (2) if each vertex is active in  $x$  timestamps,  $1 \leq x \leq |T|$ , it has a span of  $x - 1$ , hence it contributes  $x - 1$  to the objective function. Notice that since Min-NC-TCover is less restrictive than MinTCover, the optimum of Min-NC-TCover on a temporal graph  $G$  is not greater than the optimum of MinTCover on the same instance. Indeed any solution of MinTCover on  $G$  is also a solution of Min-NC-TCover on the same instance. Furthermore, notice that Min-NC-TCover is an NP-hard problem, since on a temporal graph defined on two timestamps the two problems are identical and MinTCover is known to be NP-hard in this case (Froese et al., 2023).

We use a randomized rounding algorithm to find an  $O(\log n)$  approximation solution to the Min-NC-TCover problem. Then, we transform the solution for the Min-NC-TCover into a solution for the MinTCover problem increasing the cost within a factor of at most  $T - 1$ . Thus, we obtain a  $O(T \log n)$  approximation for the MinTCover problem. The integer program formulation of the Min-NC-TCover problem is presented in Fig. 2. Notice that in the formulation a variable  $x_v^t$ , for  $v \in V$  and  $t \in \{1, 2, \dots, T\}$ , is equal to 1 when vertex  $v$  is active in timestamp  $t$ , 0 when it is not active. Furthermore, notice that we can assume that every vertex is active in at least one timestamp (if it is active in no timestamp, we can make it active in one timestamp without increasing the value of the objective function), thus constraint (4) in Fig. 2 is always satisfied.

The  $O(T \log n)$  approximation for the MinTCover problem is presented in Algorithm 1. The algorithm solves an ILP relaxation from Fig. 2, then it defines a solution for MinTCover

by rounding the variables and by defining each vertex active in an interval that includes the minimum and maximum timestamp delimited by the rounded variables.

---

**Algorithm 1**  $O(T \log n)$  approximation algorithm for the MinTCover problem.

---

1. Solve the LP relaxation of the ILP formulation from Fig. 2, where we relax constraint (6) to be  $0 \leq x_v^t \leq 1$ .
  2. Define  $c = 4n^2$ .
  3. For every variable  $x_v^t$ , define a boolean variable  $X_v^t$ , initialized to 0.
  4. Repeat  $\log_4 c$  times the following point (point 5), independently for each variable
  5. For every variable  $x_v^t$ , assign  $X_v^t$  to 1 with probability  $x_v^t$ , that is  $P[X_v^t = 1] = x_v^t$ .
  6. For every vertex  $v$  such that there exist at least two variables  $X_v^t = X_v^t = 1$ , let  $t_{min}$  be the smallest  $t$  such that  $X_v^t = 1$  and  $t_{max}$  be the maximum  $t$  such that  $X_v^t = 1$ . We make the vertex  $v$  active in interval  $[t_{min}, t_{max}]$ .
- 

We prove now the correctness and the approximation ratio of Algorithm 1.

**Lemma 5** *With probability at least  $\frac{1}{2}$ , Algorithm 1 outputs a feasible solution for the MinTCover problem that has a span of at most  $O(T \log n)$  of the span of an optimal solution.*

**Proof** First, we consider the correctness of the Algorithm 1, in particular that each temporal edge is covered, then we consider the approximation factor.

Consider a temporal edge  $\{u, v, t\}$ , we bound the probability that the edge is not covered after one step of rounding:

$$\begin{aligned} Pr(\{u, v, t\} \text{ is not covered}) &= Pr(x_u^t = 0 \text{ and } x_v^t = 0) \\ &= Pr(x_u^t = 0)Pr(x_v^t = 0) \\ &= (1 - Pr(x_u^t = 1))(1 - Pr(x_v^t = 1)) \\ &= (1 - x_u^t)(1 - x_v^t) \leq 1/4 \end{aligned}$$

The last step follows from the fact that  $(1 - x_u^t)(1 - x_v^t)$  is maximized when  $x_u^t = x_v^t = 1/2$ . Then, after  $\log_4 c$  rounds it holds that

$$\begin{aligned} Pr(\{u, v, t\} \text{ is not covered after } \log_4 c \text{ rounds}) &= \\ \prod_{i=1}^{\log_4 c} Pr(\{u, v, t\} \text{ is not covered in round } i) &\leq \left(\frac{1}{4}\right)^{\log_4 c} = \frac{1}{c} \end{aligned}$$

Consider the overall set  $E$  of temporal edges. Recall that it holds that  $|E| \leq n^2$ , since each pair of vertices is connected by at most two temporal edges. Using the union bound we show that the probability that after  $\log_4 c$  rounds there exist uncovered temporal edges is less than  $1/4$ .

$$Pr((\exists \{u, v, t\} \in E \text{ not covered after } \log_4 c \text{ rounds})) \tag{7}$$

$$\leq \sum_{\{u, v, t\} \in E} Pr(\{u, v, t\} \text{ is not covered after } \log_4 c \text{ rounds}) \leq |E| \frac{1}{c} \leq \frac{1}{4} \tag{8}$$

We prove now the approximation ratio of Algorithm 1. We denote by  $OPT_f$  the fractional optimum of the ILP from the Fig. 2. Notice that by the first constraint of the ILP it holds that

$\sum_{t=1}^T x_v^t \geq 1$ , hence  $\sum_{t=1}^T x_v^t - 1 \geq 0$ . After step 4 of Algorithm 1 we have that the expected span of our solution:

$$\mathbb{E}[\sum_{v \in V} (\sum_{t=1}^T X_v^t - 1)] = \sum_{v \in V} (\sum_{t=1}^T \mathbb{E}[X_v^t] - 1) \leq \sum_{v \in V} (\sum_{t=1}^T \log_4 c \Pr(x_v^t = 1) - 1) \tag{9}$$

$$= \sum_{v \in V} (\sum_{t=1}^T \log_4 c x_v^t - 1) \leq \log_4 c \sum_{v \in V} (\sum_{t=1}^T x_v^t - 1) \leq \log_4 c \cdot OPT_f \tag{10}$$

Now, using the Markov inequality, we have that:

$$\Pr[\text{Cost of Algorithm 1} \geq 4 \log_4 c \cdot OPT_f] \leq 1/4 \tag{11}$$

Thus, using union bound we have that the probability that solution provided by Algorithm 1 is not valid (Eq. 8) or does not have the desired approximation ratio (Eq. 11) is less than or equal to 1/2. If at the end of step 4, we do not obtain a solution of Min-NC-TCover with the aforementioned properties, we repeat the algorithm (in expectation we need 2 repetitions).

In step 5, we transform the solution  $\mathcal{C}$  of the Min-NC-TCover problem into a solution  $\mathcal{A}_2$  of the MinTCover problem by defining, for each vertex  $v \in V$ , an interval  $I_{\mathcal{A}_2}(v)$  of minimum span that includes all the timestamps where  $v$  is defined to be active in  $\mathcal{C}$ . It follows that if a vertex  $v$  is active in a timestamp  $t$  in  $\mathcal{C}$  then it is active in a timestamp  $t$  in  $\mathcal{A}_2$ . Thus, since the solution  $\mathcal{C}$  covers all the temporal edges of  $G$ , then also the solution  $\mathcal{A}_2$  produced by the Algorithm 1 covers all the temporal edges of  $G$  and is a valid solution to the MinTCover problem.

If we denote by  $ALG$  the span of the solution  $\mathcal{A}_2$  returned by the Algorithm 1 and by  $ALG(v)$  the number of timestamps in which the vertex  $v$  is active in the solution returned by Algorithm 1, we have  $ALG = \sum_{v \in V} (ALG(v) - 1)$ . However, from step 5 of Algorithm 1 we have that if there exists only one  $t$  such that  $X_v^t = 1$ , then also  $ALG(v) = 1$ . Thus, we have that  $\mathbb{E}[ALG(v) - 1] \leq (T - 1)(\sum_{t=1}^T \mathbb{E}[X_v^t] - 1)$ . Therefore:

$$\mathbb{E}[ALG] = \sum_{v \in V} \mathbb{E}[(ALG(v) - 1)] \leq (T - 1) \sum_{v \in V} (\sum_{t=1}^T \mathbb{E}[X_v^t] - 1) \leq (T - 1) \log_4 c \cdot OPT_f$$

where the last inequality follows from Inequality 10. We have that the expected approximation ratio of Algorithm 1 is  $O(T \log n)$ . □

We can prove now that the overall algorithm has an approximation factor  $O(T \log n)$ .

**Theorem 3** *MinTCover can be approximated within factor  $O(T \log n)$  in polynomial time.*

**Proof** First, notice that both algorithms have polynomial time complexity. Now, let  $\mathcal{A}_1$  be the activity assignment of the first phase of the algorithm and let  $\mathcal{A}_2$  be the activity assignment of the second phase of the algorithm. Let  $\mathcal{A}_f = \mathcal{A}_1 + \mathcal{A}_2$  be the solution consisting of the assignment of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . Notice that, since all the vertices defined to be active by  $\mathcal{A}_1$  are removed,  $\mathcal{A}_f$  is well-defined.

Let  $\mathcal{A}^*$  be an optimal solution of MinTCover on instance  $G$ , recall that by Lemma 4 it holds that

$$s(\mathcal{A}_1) \leq 2(T - 1)s(\mathcal{A}^*),$$

while by Lemma 5 it holds that

$$s(\mathcal{A}_2) \leq O(T \log n s(\mathcal{A}^*)).$$

Summing up the two inequalities we have that

$$s(\mathcal{A}_f) \leq s(\mathcal{A}_1) + s(\mathcal{A}_2) \leq 2(T - 1)s(\mathcal{A}^*) + O(T \log n)s(\mathcal{A}^*) \leq O(T \log n)s(\mathcal{A}^*),$$

thus concluding the proof.  $\square$

We have presented a randomized rounding approximation algorithm. The algorithm can be derandomized applying the conditional probability method. The approach is based on choosing deterministically the value of variables  $X_v^t$  so that the conditional probability that the obtained solution is not a feasible solution of span  $O(T \log n)$  is a value below 1.

## 4 Algorithms for 1-MinTCover

In this section we study 1-MinTCover, the variant of the MinTCover problem in which the temporal graph from each timestamp contains a single temporal edge. For this variant we give a fixed-parameter algorithm for parameter the span of the solution, more precisely a reduction to a polynomial kernel, and an approximation algorithm with approximation factor  $4(T - 1)$ . We start by proving some properties of 1-MinTCover that are useful for both algorithms.

Consider a solution  $\mathcal{A}$  of 1-MinTCover on a temporal graph  $G$ . We assume that the instance has been preprocessed as described in Sect. 2.1, hence the union graph  $G_U$  does not contain a disjoint simple cycle whose edges have exactly one label and there is no vertex having global degree one. Moreover, we assume that the union graph  $G_U$  is connected, otherwise we can solve 1-MinTCover on each connected component independently.

### 4.1 1-MinTCover properties

In this subsection we focus mainly on the vertices that have global degree greater than two in  $G$  and we prove some properties on these vertices. Denote by  $\mathcal{D} \subseteq V$  the set of vertices in  $G$  having global degree greater than two (that is those vertices having at least three incident temporal edges). We start by proving the following result.

**Lemma 6** *Let  $G$  be an instance of 1-MinTCover that admits a solution of span  $k$ . Let  $\mathcal{D} \subseteq V$  be the set of vertices in  $G$  having global degree greater than two. Then, (1)  $|\mathcal{D}| \leq 2k$ ; (2) There are at most  $6k$  temporal edges incident to vertices of  $\mathcal{D}$ .*

**Proof** Recall that  $|V| = n$  and  $|E| = m$ . Notice that in an instance of 1-MinTCover, since there exists at most one temporal edge in each timestamp, the following property holds: a vertex that covers  $z$  temporal edges,  $1 \leq z \leq m$ , has a span of at least  $z - 1$ . Since  $G$  admits a solution of 1-MinTCover of span  $k$ , it follows that there can be at most  $n + k$  temporal edges in  $G$ .

(1) Recall that each vertex of  $G$  has global degree at least two, since we have applied the preprocessing from Sect. 2.1. Thus the sum of the vertex degrees (we count twice each temporal edge), is bounded as follows (recall that  $\Delta(v)$  denotes the global degree of vertex  $v$ ):

$$\sum_{v \in V} \Delta(v) \leq 2(n + k). \quad (12)$$

Notice that

$$\sum_{v \in V} \Delta(v) = \sum_{v \in \mathcal{D}} \Delta(v) + \sum_{v \in V \setminus \mathcal{D}} \Delta(v).$$

We know that the vertices  $v \in \mathcal{D}$  are the only vertices in  $G$  that have  $\Delta(v) \geq 3$ , and thus the vertices  $v \in V \setminus \mathcal{D}$  have degree precisely 2 (since we have eliminated the degree one vertices using the preprocessing from Sect. 2.1). Therefore, we have that  $\sum_{v \in V \setminus \mathcal{D}} \Delta(v) = 2(n - |\mathcal{D}|)$ .

Assume towards a contradiction that  $|\mathcal{D}| > 2k$ . Then, since each vertex in  $\mathcal{D}$  has global degree at least three, we have that:

$$\sum_{v \in \mathcal{D}} \Delta(v) + \sum_{v \in V \setminus \mathcal{D}} \Delta(v) \geq 2n + |\mathcal{D}| > 2(n + k),$$

thus leading to a contradiction with Inequality 12. Thus  $|\mathcal{D}| \leq 2k$ .

(2) Since by Inequality 12, it holds that  $\sum_{v \in V} \Delta(v) = \sum_{v \in \mathcal{D}} \Delta(v) + \sum_{v \in V \setminus \mathcal{D}} \Delta(v) \leq 2(n + k)$ , it follows that

$$\sum_{v \in \mathcal{D}} \Delta(v) \leq 2(n + k) - \sum_{v \in V \setminus \mathcal{D}} \Delta(v). \tag{13}$$

Since each vertex of  $V \setminus \mathcal{D}$  has degree exactly 2, it follows that

$$\sum_{v \in V \setminus \mathcal{D}} \Delta(v) = 2|V \setminus \mathcal{D}|. \tag{14}$$

From Eqs. 13 and 14, we have that

$$\sum_{v \in \mathcal{D}} \Delta(v) \leq 2(n + k) - 2|V \setminus \mathcal{D}|. \tag{15}$$

Since we have proved that  $|\mathcal{D}| \leq 2k$ , thus  $|V \setminus \mathcal{D}| \geq n - 2k$ , from Eq. 15 it follows that

$$\sum_{v \in \mathcal{D}} \Delta(v) \leq 2(n + k) - 2|V \setminus \mathcal{D}| \leq 2(n + k) - 2(n - 2k) = 6k$$

thus concluding the proof. □

Now, we prove that the union graph  $G'_U$  obtained from  $G_U$  by removing the vertices in  $\mathcal{D}$  consists of disjoint paths.

**Lemma 7** *Let  $G'$  be the temporal graph obtained by removing the vertices in  $\mathcal{D}$ . Then, the union graph  $G'_U$  associated with  $G'$  consists of a set of disjoint paths where each edge has a single label.*

**Proof** First notice that, since  $G'$  is obtained by removing the vertices in  $\mathcal{D}$ , there cannot be vertices in  $G'_U$  of degree larger than 2, thus each connected component is either a path or a simple cycle. Furthermore, notice that if two vertices of  $G'_U$  are connected by an edge with different labels, then since these vertices have global degree two in  $G$ , they induce a connected component in  $G$ , thus they would have been removed by Phase 2 of Preprocessing of Sect. 2.1.

Now, assume that there exists a simple cycle; then either is a simple cycle not connected to any vertex of  $\mathcal{D}$ , but then it would have been removed by Phase 2 of preprocessing, or there exists a vertex  $v$  of the cycle connected to a vertex of  $\mathcal{D}$ . But then  $v$  would have degree larger than 2 in  $G_U$  and it would be in  $\mathcal{D}$ , and thus it cannot belong to  $G'$ . □

## 4.2 A polynomial kernel

We now present an algorithm that computes a polynomial kernel for 1-MinTCover for parameter  $k$  (the span of the solution), based on Lemma 6 and Lemma 7. Informally, since  $|\mathcal{D}| \leq 2k$  and the temporal edges incident in some vertex of  $\mathcal{D}$  are at most  $6k$  (see Lemma 6), we have to shrink only the number of vertices of degree two.

As a preliminary step, consider the graph  $G$  and compute the set  $\mathcal{S}$  of connected components of  $G_U$  induced by vertices of degree two. Notice that, by Lemma 7, each connected component in  $\mathcal{S}$  is a path in  $G_U$ . Furthermore, for each  $C \in \mathcal{S}$ , denote by  $\{v_{C,1}, w_{C,1}, t_{C,1}\}$  and  $\{v_{C,2}, w_{C,2}, t_{C,2}\}$  the two temporal edges of  $G$  connecting vertices  $v_{C,1}, v_{C,2} \in \mathcal{D}$  with vertices  $w_{C,1}, w_{C,2}$ , respectively, of  $C$ .

Given a path  $C$  in  $\mathcal{S}$ , compute the minimum span, denoted by  $s(C)$ , of a timeline activity for the vertices of  $C$  (notice that the timeline activity is defined only for the vertices of  $C$ , not for  $v_{C,1}$  and  $v_{C,2}$ ) that covers each temporal edge of  $C$  plus both temporal edges  $\{v_{C,1}, w_{C,1}, t_{C,1}\}$  and  $\{v_{C,2}, w_{C,2}, t_{C,2}\}$ .  $s(C)$  can be computed in polynomial time by first finding, for each vertex  $w$  of  $C$ , the span  $s(w, C)$  of a timeline activity that covers each temporal edge of  $C$  plus both temporal edges  $\{v_{C,1}, w_{C,1}, t_{C,1}\}$  and  $\{v_{C,2}, w_{C,2}, t_{C,2}\}$ , where  $w$  covers the two temporal edges incident in it, while each other vertex of  $C$  covers exactly one temporal edge incident in it.  $s(C)$  is then the minimum, over vertices  $w$  of  $C$ , of  $s(C, w)$ .

Now, we present the details of the reduction to kernel (Algorithm 2). Starting from  $G = (V, E, T)$ , we compute a temporal graph  $G' = (V', E', T)$  by applying Algorithm 2 on each connected component  $C$  of  $\mathcal{S}$ .

---

**Algorithm 2** The algorithm that computes a polynomial kernel of 1-MinTCover for path  $C$  of  $\mathcal{S}$ .

---

Consider a path  $C$  of  $\mathcal{S}$ , connected to vertices of  $\mathcal{D}$  with temporal edges  $\{v_{C,1}, w_{C,1}, t_{C,1}\}$  and  $\{v_{C,2}, w_{C,2}, t_{C,2}\}$  (possibly  $w_{C,1} = w_{C,2}$ ).

$C$  is replaced by a corresponding path  $P(C)$  in  $G'$  as follows:

1. If  $C$  has a single vertex  $w_C$ , then  $P(C)$  consists of a vertex  $z_C$  and temporal edges  $\{v_{C,1}, z_C, t_{C,1}\}$ ,  $\{v_{C,2}, z_C, t_{C,2}\}$ ,
  2. If  $C$  has at least two vertices, then  $P(C)$  consists of vertices  $z_{C,1}$  and  $z_{C,2}$  and temporal edges  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$ ,  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$ , plus if  $t_{C,1} > t_{C,2}$ , temporal edge  $\{z_{C,1}, z_{C,2}, t_{C,1} + s(C)\}$ , if  $t_{C,1} \leq t_{C,2}$ , temporal edge  $\{z_{C,1}, z_{C,2}, t_{C,2} + s(C)\}$ .
- 

Algorithm 2 replaces each path  $C$  with a corresponding path  $P(C)$ . In particular, it shrinks each long  $C$  (consisting of at least two vertices), by replacing it with a path  $P(C)$  consisting of exactly two vertices. If the vertices of  $C$  cover all the temporal edges incident in them with a span  $s(C)$ , then also the vertices of  $P(C)$  covers all the temporal edges incident in them with span  $s(C)$ .

We prove the correctness of Algorithm 2 in the following lemma.

**Lemma 8** *Let  $G = (V, E, T)$  be a temporal graph input of 1-MinTCover and let  $G^* = (V^*, E^*, T^*)$  be the temporal graph computed by Algorithm 2 on input  $G$ . Then  $|V^*| \leq 8k$  and  $|E^*| \leq 9k$ . Moreover, 1-MinTCover admits a solution of span  $k$  on instance  $G$  if and only if 1-MinTCover admits a solution of span  $k$  on instance  $G^*$ .*

**Proof** First, we prove that  $|V^*| \leq 8k$  and  $|E^*| \leq 9k$ . We start by considering  $V^*$ . By Lemma 6 we have that  $|\mathcal{D}| \leq 2k$ . Now, consider the vertices of degree two in  $G^*$ . By construction each

of such vertices is adjacent to a vertex of  $\mathcal{D}$ . Since by Lemma 6, there exist at most  $6k$  temporal edges of  $G$  incident in a vertex of  $\mathcal{D}$  and the same property holds for  $G^*$ , it follows that there are at most  $6k$  vertices having degree two in  $G'$ , thus

$$|V^*| \leq 2k + 6k = 8k.$$

As for the set  $E^*$ , there are at most  $6k$  temporal edges incident in vertices of  $\mathcal{D}$ . Moreover, by construction each vertex of  $V^* \setminus \mathcal{D}$  is connected with at most another vertex of  $V^* \setminus \mathcal{D}$ . Since  $|V^* \setminus \mathcal{D}| \leq 6k$ , there are at most  $3k$  temporal edges connecting vertices of  $V^* \setminus \mathcal{D}$ . It follows that

$$|E^*| \leq 6k + 3k = 9k.$$

Now, we prove the correctness of Algorithm 2. Assume that  $\mathcal{A}$  is a solution of 1-MinTCover on instance  $G$ . Then, define a solution  $\mathcal{A}^*$  of 1-MinTCover on instance  $G^*$  as follows:

1. For each  $v \in \mathcal{D}$ , then  $I_{\mathcal{A}^*}(v) = I_{\mathcal{A}}(v)$
2. For each  $P(C)$  in  $G^*$ , if  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$  is covered by  $v_{C,1}$  or if  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$  is covered by  $v_{C,2}$ , then compute an activity timeline  $\mathcal{A}_C^*$  of span 0, that covers the temporal edges of  $P(C)$  and possibly one uncovered temporal edge of  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$ ,  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$
3. For each  $P(C)$  in  $G^*$ , if  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$  is not covered by  $v_{C,1}$  and if  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$  is not covered by  $v_{C,2}$ , then compute an activity timeline  $\mathcal{A}_C^*$  of span  $s(C)$  that covers the temporal edges of  $P(C)$  as follows: if  $t_{C,1} > t_{C,2}$ , then  $z_{C,1}$  covers its two incident temporal edges, else  $z_{C,2}$  covers its two incident temporal edges; the other vertex of  $P(C)$  covers the uncovered temporal edge of  $P(C)$  with span equal to 0.

By construction it holds that:

$$s(\mathcal{A}^*) = s(\mathcal{A}).$$

Now, consider a solution  $\mathcal{A}^*$  of 1-MinTCover on instance  $G^*$ . Then, define a solution  $\mathcal{A}$  of 1-MinTCover on instance  $G$  as follows:

1. For each  $v \in \mathcal{D}$ , then  $I_{\mathcal{A}}(v) = I_{\mathcal{A}^*}(v)$
2. For each  $P(C)$  in  $G^*$ , if  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$  is covered by  $v_{C,1}$  or if  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$  is covered by  $v_{C,2}$ , then compute an activity timeline  $\mathcal{A}_C$  of span 0 that covers all the uncovered temporal edges incident in  $C$ .
3. For each  $P(C)$  in  $G^*$ , if  $\{v_{C,1}, z_{C,1}, t_{C,1}\}$  is not covered by  $v_{C,1}$  and  $\{v_{C,2}, z_{C,2}, t_{C,2}\}$  is not covered by  $v_{C,2}$ , then compute an activity timeline  $\mathcal{A}_C$  that covers each temporal edge incident in vertices of  $C$  and that has span equal to  $s(C)$ .

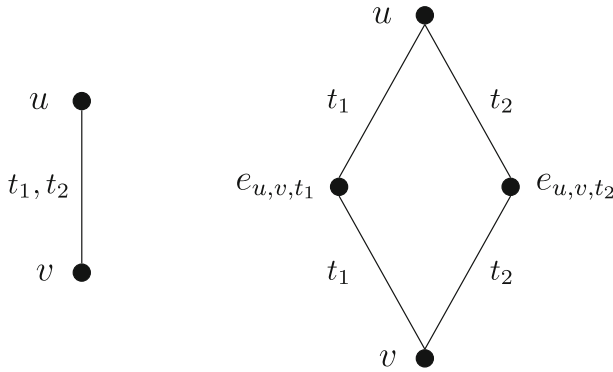
By construction it holds that:

$$s(\mathcal{A}) \leq s(\mathcal{A}^*)$$

thus concluding the proof. □

While we are able to bound the number of temporal edges, the timestamps may have values that are not bounded by a function of  $k$ . We conclude this subsection by considering the time domain  $T^*$  and showing how to modify it so that each timestamp is bounded by a function of  $k$ . If  $T^* = [1, \dots, T^*]$ , where  $|T^*| > 9k(k + 1)$ , we redefine  $T^*$  as follows:

1. Order the timestamps in increasing order



**Fig. 3** An example of subdivision of an edge: two vertices  $u$  and  $v$  connected by an edge  $\{u, v\}$  with labels  $t_1$  and  $t_2$  (left); the subdivision of edge  $\{u, v\}$  (right)

- From  $i = 1$  to  $T^* - 1$ , if for two consecutive timestamps  $t_i^*, t_{i+1}^*$  it holds that  $t_{i+1}^* - t_i^* > k + 1$ , then update each timestamp  $t_j^*$ , with  $i + 1 \leq j \leq T^*$ , as follows:

$$t_j^* = t_j^* - (t_{i+1}^* - t_i^*) + (k + 1)$$

At the end of this procedure each pair of consecutive timestamps have difference at most  $k + 1$ . Since by Lemma 8 we have that  $|E^*| \leq 9k$ , it follows  $|T^*| \leq 9k(k + 1)$ . Furthermore, if consider two timestamps  $t_a^*, t_b^*$  with  $t_b^* \geq t_a^*$ . If  $t_b^* - t_a^* = h \leq k$  before the updating, then  $t_b^* - t_a^* = h$  after the updating. If  $t_b^* - t_a^* = h > k$  before the updating, then  $t_b^* - t_a^* = k + 1$  after the updating, thus no vertex can be defined active in an interval of  $G^*$  that includes  $[t_a^*, t_b^*]$ .

### 4.3 An approximation algorithm

In this subsection we present a polynomial time  $4(T - 1)$ -approximation algorithm (Algorithm 3) for the 1-MinTCover problem. We start by showing that starting from a union graph  $G_U$  where edges can have multiple labels, we can compute a corresponding simple union graph  $G_{U,s}$  where each edge has a single label and such that  $G_U$  has a feedback vertex set of size  $k$  if and only if  $G_{U,s}$  has a feedback vertex set of size  $k$ . We recall that a feedback vertex set of a graph is a subset of the vertex set that, after its removal, leaves the graph acyclic. For union graph  $G_U$  (which formally is a labeled multigraph), the definition of feedback vertex set is the same as for graphs, but in this case there can be cycles of length 2 between two vertices, when they are connected by two parallel edges with different labels.

Consider the union graph  $G_U$ , then compute a graph  $G_{U,s}$  by subdividing each edge of  $G_U$  with more than one label in a set of edges as follows (an example in Fig. 3): for each  $\{u, v\} \in E_U$  labeled by  $t$ , (1) a new vertex  $e_{u,v,t}$  is added to  $G_{U,s}$ ; (2) edge  $\{u, v\}$  is removed and (3) edges  $\{e_{u,v,t}, u\}$  and  $\{e_{u,v,t}, v\}$ , labeled by  $t$ , are added to  $G_{U,s}$ .

The vertices  $e_{u,v,t}$  added to  $G_{U,s}$  are called subdivision vertices.

**Lemma 9** Consider a union graph  $G_U$  and the corresponding simple union graph  $G_{U,s}$ . Then  $G_U$  has a feedback vertex set of size  $k$  if and only if  $G_{U,s}$  has a feedback vertex set of size  $k$ .

**Proof** Consider a feedback vertex set  $F$  of  $G_U$ . We claim that  $F$  is also a feedback vertex set of  $G_{U,s}$ . Indeed, assume that there exists a cycle  $C'$  in  $G_{U,s}$  that does not contain any



vertex of  $F$ . Consider the set of vertices  $A$  of  $C'$  which are not subdivision vertices. Then in  $G_U$  the vertices corresponding to  $A$  induces a cycle (possibly of length 2), thus  $F$  would not be a feedback vertex set.

Assume now, that  $F'$  is a feedback vertex set of  $G_{U,s}$ , we construct a feedback vertex set  $F''$  of  $G_{U,s}$  that does not contain subdivision vertices as follows: if one subdivision vertex  $e_{u,v,t}$  belongs to  $F'$ , then replace  $e_{u,v,t}$  with one of  $u$  or  $v$  that is not already in  $F'$ . Notice that we can assume that at most one  $u$  and  $v$  is in  $F'$ , otherwise  $e_{u,v,t}$  is an isolated vertex after the removal of  $u$  and  $v$  and thus it can be removed from  $F'$ . We claim that  $F''$  is a feedback vertex set of  $G_{U,s}$ . Assume this is not the case, then after the removal of  $F''$  there is a cycle  $C'$  in  $G_{U,s}$ . However, notice that  $C'$  must contain a subdivision vertex  $e_{u,v,t}$ , otherwise  $C'$  is a cycle also after the removal of  $F'$ , leading to a contradiction. Since  $e_{u,v,t}$  belongs to  $F' \setminus F''$ , by construction  $F''$  contains one of  $u, v$ , hence  $e_{u,v,t}$  is a leaf after the removal of  $F''$ . It follows that, after the removal of  $F''$ ,  $C'$  is not a cycle in  $G_{U,s}$ . Now, the set of vertices of  $G_U$  corresponding to  $F''$  is also a feedback vertex set of  $G_U$ , and has size equal to  $|F'|$ , thus concluding the proof.  $\square$

---

### Algorithm 3 $4(T - 1)$ -approximation algorithm for 1-MinTCover

---

1. Construct the union graph  $G_U = (V, E)$  of  $G$  and the corresponding simple union graph  $G_{U,s}$ .
  2. Compute a 2-approximate feedback vertex set  $F$  of  $G_{U,s}$  and  $G_U$  (see the approximation algorithm in Becker and Geiger (1996); Bafna et al. (1999))
  3. Make each vertex  $v \in F$  active in the time interval  $[1, T]$ .
  4. Cover the graph  $G - F$  using the first reduction rule from Sect. 2.1, since  $G - F$  is acyclic.
- 

Algorithm 3 considers the union graph  $G_U$  and its associated simple union graph  $G_{U,s}$ , computes a feedback vertex set of  $G_{U,s}$  and, for each vertex in the feedback vertex set, makes it active in the whole time domain. The key to the analysis of Algorithm 3 is the following structural lemma that relates the size of an optimal solution of 1-MinTCover to the size of a minimum feedback vertex set of  $G_{U,s}$ .

**Lemma 10** *Let  $G$  be a temporal graph, input of 1-MinTCover. Let  $G_U$  be the corresponding union graph and  $G_{U,s}$  be the corresponding simple union graph. If an optimal solution of 1-MinTCover on  $G$  has a span of  $k$ , then a feedback vertex set of  $G_U$  and  $G_{U,s}$  has at most  $2k$  vertices.*

**Proof** Consider the set  $\mathcal{D}$  of vertices having degree larger than three. By Lemma 6, it follows that  $|\mathcal{D}| \leq 2k$ . Moreover, after the removal of the vertices in  $\mathcal{D}$  (see Lemma 7), we obtain a graph consisting of disjoint paths. Thus  $\mathcal{D}$  is a feedback vertex set of  $G_U$  consisting of at most  $2k$  vertices. By Lemma 9,  $\mathcal{D}$  is also a feedback vertex set of  $G_{U,s}$ , thus concluding the proof.  $\square$

We analyze now the correctness and the performance of Algorithm 3.

**Theorem 4** *Algorithm 3 is a polynomial  $4(T - 1)$ -approximation algorithm for the 1-MinTCover problem.*

**Proof** Algorithm 3 produces a valid cover of  $G$  in polynomial time, since  $F$  is a feedback vertex set of  $G_{U,s}$  and  $G_{U,s} \setminus F$  is acyclic and thus, we can iteratively apply the first rule from Sect. 2.1 and obtain a timeline activity of span 0 for  $G - F$ .

We analyze now the approximation factor of Algorithm 3. First, observe that the global span of the activity timeline returned by the algorithm is

$$ALG \leq 2(T - 1)|F^*|,$$

where  $F^*$  is a minimum feedback vertex set in the graph  $G_{U,s}$ . From Lemma 10 we have that

$$|F^*| \leq 2 \cdot OPT,$$

where  $OPT$  is the minimum span of any activity timeline on the graph  $G$ . Thus, it holds that

$$ALG \leq 2(T - 1)|F^*| \leq 2(T - 1) 2 OPT \leq 4(T - 1)OPT,$$

thus concluding the proof.  $\square$

## 5 Conclusion

In this paper we have presented algorithmic contributions on MinTCover, a problem recently introduced for covering temporal graphs. We have presented an approximation algorithm of factor  $O(T \log n)$  and for 1-MinTCover, the restriction where at most one temporal edge is defined in each timestamp, we have presented a reduction to a polynomial kernel and a  $4(T - 1)$  approximation algorithm.

There are several interesting open problems related to MinTCover. There is a gap in the approximation complexity of the problem. In particular is it possible to obtain an approximation algorithm whose factor does not depend on  $T$  for MinTCover and 1-MinTCover? Another interesting open problem is whether it is possible to obtain a fixed-parameter tractable algorithm with parameter the span of the solution.

**Funding** Open access funding provided by Università degli studi di Bergamo within the CRUI-CARE Agreement. This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, Project Number PN-III-P1-1.1-TE-2021-0253, within PNCDI III.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Agarwal, A., Charikar, M., Makarychev, K., & Makarychev, Y. (2005). O( $\sqrt{\log n}$ ) approximation algorithms for Min UnCut, Min 2CNF Deletion, and directed cut problems. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing* (pp. 573–581). <https://doi.org/10.1145/1060590.1060675>
- Akrida, E. C., Mertzios, G. B., Spirakis, P. G., & Raptopoulos, C. L. (2021). The temporal explorer who returns to the base. *Journal of Computer and System Sciences*, 120, 179–193. <https://doi.org/10.1016/j.jcss.2021.04.001>
- Akrida, E. C., Mertzios, G. B., Spirakis, P. G., & Zamaraev, V. (2020). Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107, 108–123. <https://doi.org/10.1016/j.jcss.2019.08.002>
- Bafna, V., Berman, P., & Fujito, T. (1999). A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12(3), 289–297. <https://doi.org/10.1137/S0895480196305124>
- Becker, A., & Geiger, D. (1996). Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artificial Intelligence*, 83(1), 167–188. [https://doi.org/10.1016/0004-3702\(95\)00004-6](https://doi.org/10.1016/0004-3702(95)00004-6)
- Bumpus, B. M., & Meeks, K. (2023). Edge exploration of temporal graphs. *Algorithmica*, 85(3), 688–716. <https://doi.org/10.1007/s00453-022-01018-7>
- Dondi, R., & Lafond, M. (2023). An FPT algorithm for temporal graph untangling. In *18th International Symposium on Parameterized and Exact Computation (IPEC 2023)*. <https://doi.org/10.4230/LIPICS.IPEC.2023.12>. Schloss-Dagstuhl-Leibniz Zentrum für Informatik
- Dondi, R., & Popa, A. (2023). Timeline cover in temporal graphs: Exact and approximation algorithms. In *International Workshop on Combinatorial Algorithms* (pp. 173–184). [https://doi.org/10.1007/978-3-031-34347-6\\_15](https://doi.org/10.1007/978-3-031-34347-6_15)
- Dondi, R. (2023). Untangling temporal graphs of bounded degree. *Theoretical Computer Science*, 969, 114040. <https://doi.org/10.1016/J.TCS.2023.114040>
- Dondi, R., & Hosseinzadeh, M. M. (2021). Dense sub-networks discovery in temporal networks. *SN Computer Science*, 2(3), 158. <https://doi.org/10.1007/s42979-021-00593-w>
- Erlbach, T., Hoffmann, M., & Kammer, F. (2021). On temporal graph exploration. *Journal of Computer and System Sciences*, 119, 1–18. <https://doi.org/10.1016/j.jcss.2021.01.005>
- Fluschnik, T., Molter, H., Niedermeier, R., Renken, M., & Zschoche, P. (2020). Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806, 197–218. <https://doi.org/10.1016/j.tcs.2019.03.031>
- Froese, V., Kunz, P., & Zschoche, P. (2023). Disentangling the computational complexity of network untangling. *Theory of Computing Systems*. <https://doi.org/10.1007/s00224-023-10150-y>
- Garg, S., & Philip, G. (2016). Raising the bar for vertex cover: Fixed-parameter tractability above a higher guarantee. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1152–1166). <https://doi.org/10.1137/1.9781611974331.CH80>
- Hamm, T., Klobas, N., Mertzios, G. B., & Spirakis, P. G. (2022). The complexity of temporal vertex cover in small-degree graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence* (vol. 36, pp. 10193–10201). <https://doi.org/10.1609/aaai.v36i9.21259>
- Hochbaum, D. S. (1982). Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3), 555–556. <https://doi.org/10.1137/0211045>
- Holme, P. (2015). Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(9), 234. <https://doi.org/10.1140/epjb/e2015-60657-4>
- Holme, P., & Saramäki, J. (2019). A map of approaches to temporal networks. *Temporal Network Theory*. [https://doi.org/10.1007/978-3-030-23495-9\\_1](https://doi.org/10.1007/978-3-030-23495-9_1)
- Karakostas, G. (2009). A better approximation ratio for the vertex cover problem. *ACM Transactions on Algorithms*, 5(4), 41–1418. <https://doi.org/10.1145/1597036.1597045>
- Kempe, D., Kleinberg, J. M., & Kumar, A. (2002). Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4), 820–842. <https://doi.org/10.1006/jcss.2002.1829>
- Marino, A., & Silva, A. (2021). Königsberg sightseeing: Eulerian walks in temporal graphs. In *International Workshop on Combinatorial Algorithms* (pp. 485–500). [https://doi.org/10.1007/978-3-030-79987-8\\_34](https://doi.org/10.1007/978-3-030-79987-8_34)
- Michail, O. (2016). An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4), 239–280. <https://doi.org/10.1080/15427951.2016.1177801>
- Rozenshtein, P., Bonchi, F., Gionis, A., Sozio, M., & Tatti, N. (2020). Finding events in temporal networks: Segmentation meets densest subgraph discovery. *Knowledge and Information Systems*, 62(4), 1611–1639. <https://doi.org/10.1007/s10115-019-01403-9>

- Rozenshtein, P., Tatti, N., & Gionis, A. (2021). The network-untangling problem: From interactions to activity timelines. *Data Mining and Knowledge Discovery*, 35(1), 213–247. <https://doi.org/10.1007/s10618-020-00717-5>
- Wu, H., Cheng, J., Huang, S., Ke, Y., Lu, Y., & Xu, Y. (2014). Path problems in temporal graphs. *Proceedings of the VLDB Endowment* 7(9), 721–732 <https://doi.org/10.14778/2732939.2732945>
- Wu, H., Cheng, J., Ke, Y., Huang, S., Huang, Y., & Wu, H. (2016). Efficient algorithms for temporal path computation. *IEEE Transactions on Knowledge and Data Engineering*, 28(11), 2927–2942. <https://doi.org/10.1109/TKDE.2016.2594065>
- Zschoche, P., Fluschnik, T., Molter, H., & Niedermeier, R. (2020). The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107, 72–92. <https://doi.org/10.1016/j.jcss.2019.07.006>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.