Disjoint Temporal Walks Under Waiting Time Constraints

Allen Ibiapina[1], Raul Lopes[2(✉)], Andrea Marino[3], and Ana Silva[3]

[1] IRIF, Université Paris Cité, Paris, France
ibiapina@irif.fr

[2] LIRMM, Université de Montpellier, CNRS, Montpellier, France
rtlopes@protonmail.com

[3] Dipartimento di Statistica, Informatica, Applicazioni, Università degli Studi di Firenze, Firenze, Italy
{andrea.marino,anashirley.ferreiradasilva}@unifi.it

Abstract. A temporal (directed) graph is a (directed) graph whose edge availability varies over time, and temporal walks are sequences of adjacent edges respecting these availabilities. A third component of this scenario imposes time constraints on the vertices, known as "waiting time constraints." Drawing a parallel with a transportation network, these constraints can be envisioned as the minimum and maximum time a person would need or be willing to stay at the same location. It is known that computing a temporal walk between a pair of vertices under waiting time constraints can be achieved in polynomial time, even when optimizing certain criteria. In this article, we extend these investigations to the problem of finding k temporal walks such that no two distinct walks overlap in time. Given a multiset $\{(s_1, z_1), \ldots, (s_k, z_k)\}$ of pairs of vertices of a temporal graph $\mathcal{G}$, we consider the problem of finding a set of pairwise temporally disjoint walks $P_1, \ldots, P_k$ such that each P_i is a temporal walk from s_i to z_i. Under a given set of waiting time constraints, when $s_i = s$ and $z_i = z$ for all $i \in [k]$ we show that determining the existence of such walks is W[1]-hard when parameterized by k, and that it is NP-complete to decide the existence of a separator (i.e., a set of vertex occurrences in time intersecting all such walks) of size at most h. In the more general case, when the vertices s_i and z_i are allowed to be disjoint, we show that the walks can be found in XP time parameterized by k when each walk has to wait at least $\omega_L(u) \geq 1$ units of time on each occurrence of each vertex u used by the walk, and that the separator can be found in XP time with parameter h.

Keywords: Temporal graphs · Disjoint walks · Menger's Theorem

1 Introduction

Temporal graphs capture the temporal dependencies among edges, reflecting the intricate nature of interactions and relationships in dynamic real-world systems. They offer a nuanced perspective on complex systems, enabling researchers

I. Finocchi and L. Georgiadis (Eds.): CIAC 2025, LNCS 15680, pp. 87–103, 2025.
https://doi.org/10.1007/978-3-031-92935-9_6

to uncover temporal patterns, analyze dynamic behavior, and predict future interactions. Understanding temporal dynamics is crucial for numerous applications, including real-time recommendation systems, epidemic spread modeling, anomaly detection, and dynamic network optimization [14,22,27,29].

Existing research has focused on various applications of temporal graphs (see e.g. [14]). In particular, multi-agent path finding problems (MAPF) is an area that has grown considerably lately and is used mostly by the AI and robotics communities (see e.g. [31,32]). In this scenario, a set of agents located at certain starting points of a network want to arrive at their targets by a sequence of actions (movements on the edges of the network) avoiding collisions (no two agents can meet at the same point at the same time). This problem is related to Menger's Theorem [25] if starting and arrival points of all agents are the same, or to the so called Vertex Disjoint Paths problem if their starting and arrival points are distinct. On static graphs, the former problem is polynomial-time solvable, while the latter is NP-complete [18] and FPTparameterized by the number of paths in the undirected case [30], and NP-complete even for $k = 2$ paths in the directed case [11]. They have also been extensively studied in the temporal graphs context [2,15,19–21,26], but never taking into account waiting time constraints.

Waiting time constraints arise naturally in various scenarios [1,3,8,33,34]. For instance, in a transportation network a passenger may need some time on a node in order to transfer between vehicles (e.g. inside an airport). Similarly, in the study of a contagious disease, an infected person can only spread a disease up to some days after the infection. When constructing walks in such temporal networks, the former scenario imposes a lower bound on the time spent on a vertex, while the latter imposes an upper bound.

Given vertices s, z of a temporal graph $\mathcal{G}$, a *temporal* s, z*-walk* is a temporal walk from s to z in $\mathcal{G}$ (the formal definition of temporal walks with waiting time constraints is given in Sect. 2). It is known that computing *one* temporal walk between a pair of vertices under waiting time constraints can be done in polynomial time [1,3–5]. In this article, we explore the problem of computing *multiple* disjoint walks under waiting time constraints, where the walks may or may not share sources and destinations, and of finding sets of temporal vertices intersecting all temporal s, z-walks.

Given waiting time constraints, a temporal graph $\mathcal{G}$, vertices s, z, and integers k, and h, we prove that: (1) deciding the existence of k temporal s, z-walks in $\mathcal{G}$ sharing no vertex at the same time is W[1]-hard when parameterized by k; and (2) it is NP-complete to decide if $\mathcal{G}$ contains a set of temporal vertices (i.e. pairs (x, t) with x where is a vertex and t is a timestep) of size at most h whose removal breaks all temporal s, z-walks.

On the positive side, again considering a given set of waiting time constraints, given pairs of vertices $\{(s_1, z_1), \ldots, (s_k, z_k)\}$ of $\mathcal{G}$, and an integer k, we can decide in XP time parameterized by k if $\mathcal{G}$ contains a set $P_1, \ldots, P_k$ of temporal walks such that each P_i with $i \in [k]$ is a temporal s_i, z_i-walk and no two distinct P_i, P_j share a temporal vertex of $\mathcal{G}$. For the separator problem, we show that deciding if $\mathcal{G}$ contains a set of temporal vertices of size at most h intersecting all temporal s_i, z_i-walks with $i \in [k]$ is XP with parameter h.

Problem Definition. A *temporal (directed) graph* with *lifetime* τ is a pair $\mathcal{G} = (G, \lambda)$ where G is a multi-(di)graph and λ is a function that assigns to each edge e of G a value in $[\tau]$, called the *starting* time of e. We say that G is the *underlying (directed) graph* of $\mathcal{G}$. We call an integer $p \in [\tau]$ a *timestep* and an element $(v, t) \in V(G) \times [\tau]$ a *temporal vertex*. An s, z-walk in G is called a *non-strict temporal s, z-walk* if its edge labels are non-decreasing and a *strict temporal s, z-walk* if its edge labels are strictly increasing. Observe that the definition of strict temporal walks can be ambiguous as such walks can arise either by making the edges have a duration or by enforcing a waiting time on the vertices. We solve this ambiguity by adding more structure to the scenario, as has been done in [3,5].

In addition to temporal (directed) graphs, we also consider a triple of integer functions $(\varepsilon, \omega_L, \omega_U)$ representing the *travel times* of the edges, the *minimum waiting time* of each vertex, and the *maximum waiting time* of each vertex, respectively. Given a pair of vertices $s, z \in V(G)$, a temporal s, z-walk *respecting* $(\varepsilon, \omega_L, \omega_U)$ is an s, z-walk in G such that the edges are traversed in a time-consistent way, i.e. the arrival time $\lambda(e) + \varepsilon(e)$ of each edge e is smaller or equal to the starting time of the next one; and the time spent[1] on each vertex u is at least $\omega_L(u)$ and at most $\omega_U(u)$. If any of these functions, say $f : X \to \mathbb{N}$, is constant, we write $f \equiv c$. In such cases, we may replace f directly by c within the triple.

Two temporal s, z-walks respecting $(\varepsilon, \omega_L, \omega_U)$ are *vertex disjoint* if they do not share any internal vertex, and *t-vertex disjoint* if they do not share any internal temporal vertex. In other words, they do not pass by the same vertex *at the same time*.

Now, given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$, a triple of functions $(\varepsilon, \omega_L, \omega_U)$, and a multiset of pairs of vertices $\{(s_1, z_1), \ldots, (s_k, z_k)\}$, we define:

- Waiting t-Vertex Disjoint Temporal Walks (wtvd): are there P_1, $\ldots$, P_k such that P_i is a temporal s_i, z_i-walk respecting $(\varepsilon, \omega_L, \omega_U)$, for each $i \in [k]$, and such that no two of them share an internal temporal vertex?
- Waiting t-Vertex Separator (wtvs): given additionally an integer h, are there h temporal vertices that intersect every temporal s_i, z_i-walk respecting $(\varepsilon, \omega_L, \omega_U)$, for every $i \in [k]$?

If in particular $s_i = s$ and $z_i = z$ for every $i \in [k]$, we write s, z-wtvd and s, z-wtvs. As previously said, we show our negative results using these particular cases and our positive results using the general ones.

Our Contribution. Table 1 summarizes our results and existing results closely related to the problems investigated here. We first comment on our results concerning walks. In Theorem 4 we prove that s, z-wtvd is W[1]-hard parameterized by k even for fixed waiting time constraints (a, b, c) where $b < c$. In [20], the

[1] Time between the arrival of an edge and the starting time of the next edge.

Table 1. W[1]-hardness results refer to the natural parameter (number of walks). All the results in Column II hold for both the strict and the non-strict model. For the poly results in B.I, Menger's Theorem holds, i.e. these problems are equivalent to finding suitable separators, as discussed in the related works section. For B.II, Menger's Theorem does not hold and an example is given in Fig. 1. We also show that WTVS is NP-c even if τ is fixed (Theorem 6) and argue that it can can be trivially solved in XP time when parameterized by h.

	(I) TEMPORAL GRAPHS W/O WAITING TIME CONSTR.	(II) TEMPORAL GRAPHS WITH WAITING TIME CONSTR.
(A) DISJOINT WALKS	NP-c, W[1]-hard, and XP [20,21]	NP-c, W[1]-hard (follow from A.I and B.II) XP (Theorem 7)
(B) DISJOINT s,z -WALKS	poly [15,26]	NP-c even if τ fixed (Theorem 5) W[1]-hard (Theorem 4) XP(Theorem 7)

authors investigate WTVD with $(0,0,\tau)$ and $(1,0,\tau)$ (i.e., when no waiting time constraints are imposed on the vertices), also proving W[1]-hardness parameterized by k. Our result is stronger as it works for the single source and destination case. However, we need to impose the waiting time constraints as the s,z-WTVD becomes polynomial without them. More specifically, in Theorem 1 we prove that s,z-WTVD and s,z-WTVS with constraints $(\varepsilon,0,\tau)$ can be solved in polynomial time even for general ε and that in fact a version of Menger's Theorem holds. The case $(\varepsilon,0,0)$ follows similarly in Theorem 2.

Theorem 1. *Let s,z be vertices of a temporal directed graph $\mathcal{G}$. With respect to $(\varepsilon,0,\tau)$, the maximum number of pairwise t-vertex disjoint temporal s,z-walks in $\mathcal{G}$ is equal to the minimum size of a temporal s,z-separator. Additionally, a maximum size set of t-vertex disjoint walks and and a minimum size separator can be found in polynomial time.*

Theorem 2. *Let s,z be vertices of a temporal directed graph $\mathcal{G}$. With respect to $(\varepsilon,0,\tau)$, the maximum number of pairwise t-vertex disjoint temporal s,z-walks in $\mathcal{G}$ is equal to the minimum size of a temporal s,z-separator. Additionally, a maximum size set of t-vertex disjoint walks and a minimum size separator can be found in polynomial time.*

We remark that s,z-WTVD and s,z-WTVS for $(0,0,\tau)$ and $(1,0,\tau)$ have already been solved in [15] and [26], respectively.

Additionally, we prove that s,z-WTVD with constraints $(0,1,\tau)$ is NP-complete for fixed $\tau \geq 9$. Observe that this case can be interpreted as strict temporal walks as well, where the strictness arises from the fact that we must stay at least one timestep on each vertex. In contrast, as already mentioned, not only the strict case $(1,0,\tau)$ is solvable in polynomial time but also a version of Menger's Theorem holds [26], and this is not true for the strict case $(0,1,\tau)$,

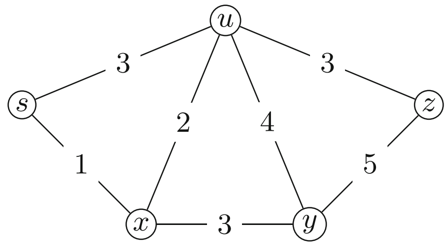

Fig. 1. In the example, there are two t-vertex disjoint temporal s, z-walks respecting $(1, 0, \tau)$: $(s, 3, u, 4, y, 5, z)$ and $(s, 1, x, 2, u, 3, z)$. However, there are no two t-vertex disjoint temporal s, z-walks respecting $(0, 1, \tau)$. Indeed, there are only three temporal s, z-walks respecting $(0, 1, \tau)$, namely $P_1 = (s, 3, u, 4, y, 5, z)$, $P_2 = (s, 1, x, 2, u, 3, z)$, and $P_3 = (s, 1, x, 3, y, 5, z)$. As $\varepsilon \equiv 0$, P_1, P_2, and P_3 make busy respectively the following sets of temporal vertices: $\{(u, 3), (u, 4), (y, 4), (y, 5)\}$, $\{(x, 1), (x, 2), (u, 2), (u, 3)\}$, and $\{(x, 1), (x, 2), (x, 3), (y, 3), (y, 4), (y, 5)\}$. We have that P_1, P_2 intersect on $(u, 3)$, P_1, P_3 intersect on $(y, 5)$, and P_2, P_3 intersect on $(x, 1)$. On the other hand, it is easy to check that no single temporal vertex intersects all three walks. Note that $\{(x, 2), (u, 4)\}$ is a separator both for $(1, 0, \tau)$ and $(0, 1, \tau)$.

as shown in the example in Fig. 1.[2] The divergence between strict cases $(0, 1, \tau)$ and $(1, 0, \tau)$ cannot be perceived when studying existing works due to the way strict walks are usually defined, considering ε to be positive only implicitly and without specifying which temporal vertices are busy. Given such a difference in tractability between the $(0, 1, \tau)$ and $(1, 0, \tau)$ cases, we advocate for a more precise definition of "strict" temporal walks.

On the positive side, in Theorem 7 we prove that WTVD can be solved in XP time when parameterized by k for every triple $(\varepsilon, \omega_L, \omega_U)$ such that ω_L is a positive function. Observe that this is a reasonable condition as the case $(0, 0, \tau)$ includes the disjoint paths problem on static directed graphs, which is known to be NP-complete even when $k = 2$ pairs of vertices [11]. Thus our result generalizes the existent algorithms to find *one* s, z-walk with waiting time constraints is polynomial-time [1,3–5]. Our XP algorithm also complements the result given in [20,21], as they prove that the cases $(0, 0, \tau)$ and $(1, 0, \tau)$ on *undirected* temporal graphs can also be solved in XP time when parameterized by k.

Finally, concerning separators, in Theorem 6 we prove that s, z-WTVS with constraint (a, b, c), for every fixed a, b, c where $a + b > 0$ and $c \geq 1 + 2a + 3b$, is NP-complete even for fixed lifetime. The case $(\varepsilon, 0, 0)$ instead is easily solved in polynomial time using a strategy similar to the one used to solve $(\varepsilon, 0, \tau)$. Observe that in particular we get that the separators problem is also hard for the $(0, 1, \tau)$ case. On the positive side, in Theorem 3 we also show that WTVS is XP when parameterized by the solution size, as it suffices to test, for every

[2] This example is a modification of the one presented in [19] to show that Menger's Theorem does not hold considering vertex disjoint temporal walks, i.e. temporal walks not passing by a same vertex.

possible subset S with h temporal vertices, whether there exists a temporal s_i, z_i-walk respecting $(\varepsilon, \omega_L, \omega_U)$ in $\mathcal{G} - S$.

Theorem 3. WTVS *can be solved in time* $\mathcal{O}((n \cdot \tau)^h \cdot k \cdot m^2)$.

This strategy cannot be immediately applied to the input temporal graph since by deleting a set of temporal vertices we generate a temporal graph whose vertices are *not* permanent (i.e., *not* active at all times), and the existing algorithms to test for temporal walks consider that all vertices are permanent. Thus to obtain the XP we modify the input to its temporal line graph [24]. Up to our knowledge, these are the first results concerning t-vertex separators with waiting time constraints.

Related Works. Some articles have focused on finding a *restless* walk, i.e. setting upper bounds on the waiting time [7,8,33,34]. Finding such walks can be seen as a way for finding robust walks. Concerning this, different definitions of robustness have been given in [6,12,13]. It is worth mentioning that finding *one* temporal s, z-*path*[3] respecting $(\varepsilon, \omega_L, \omega_U)$ is NP-hard even in very restrictive cases [7,8,34].

Concerning disjoint walks, we refer to Table 1. Concerning t-vertex disjoint walks in temporal graphs (column II), when no waiting time constraints are considered, in [26] they have proved that Menger's Theorem holds and that both s, z-WTVD and s, z-WTVS are polynomial-time solvable for the $(1, 0, \tau)$ case. This result extends also to the $(0, 0, \tau)$ case, as shown in [15]. If k s, z-paths instead of k s, z-walks are required, and still no waiting time constraints are considered, the s, z-WTVD problem is hard for every $k \geq 3$ and polynomial otherwise [15]. As it is hard for static graphs, WTVD continues to be hard for temporal graphs even without considering waiting time constraints. In particular, [20,21] provided hardness results even in very restricted cases, namely when the underlying graph is a star, studying also the path version of the problem, i.e. requiring the walks to be paths. They provided an XP algorithm which works for undirected temporal graphs. While their negative results, reported in A.I of Table 1, extends to A.II, their XP algorithm does not, as it works for undirected temporal graphs and is not able to deal with waiting time constraints. Our negative results for WTVD complement theirs as we prove that WTVD is hard even in the case of s, z-WTVD. As far as we know, our XP algorithm is the first positive result for WTVD. Furthermore, our results are the first ones for s, z-WTVD (see cell B.II) and s, z-WTVS.

For the sake of completeness, we mention that the seminal articles [2,19] have considered the case of vertex disjoint temporal walks, providing hardness results even without taking into account waiting time constraints (see [15] for a summary). As in this case Menger's Theorem does not hold in general [17,19], the complexity of computing vertex s, z-separators, i.e. disconnecting s and z by deleting vertices rather than temporal vertices as in s, z-WTVS, is independent from the one of computing vertex disjoint s, z-walks. Computing vertex s, z-separators is NP-complete [19] and W[1]-hard when parameterized by the

[3] A path is a walk with no vertex repetition.

separator size k [23,35]. Interestingly, [28] has focused on vertex separation for restless walks. Finally, a further notion of disjointness for walks has been considered in [16].

Organization. The remainder of this paper is organized as follows. In Sect. 2, we formally present the definitions and argue that the problems on temporal directed graphs are harder. In Sect. 3, we present our negative results for s, z-WTVD, providing a proof of a particular case of Theorem 4. In Sect. 4, we present the NP-completeness proof for s, z-WTVS. In Sect. 5, we present our XP algorithm for WTVD. Proofs omitted due to space restrictions can be found in the full version of this paper.

2 Preliminaries

We refer the reader to [9] for basic background in parameterized complexity. Given $n \in \mathbb{N}$, we let $[n]$ denote the set $\{x \in \mathbb{N} : x \leq n\}$ and $[n]_0$ denote the set $[n] \cup \{0\}$. Given an edge e with *endpoints* uv, we sometimes write simply $e = uv$. This is an abuse of language since we are dealing with a multigraph, which means that multiple edges with the same endpoints might exist. Nevertheless, we adopt this notation only if no ambiguity might arise. As we are going to deal with undirected and directed graphs, we use the term "edges" also to refer to the arcs of a directed graph and write uv to refer to an arc from u to v. As we will see shortly, the problems investigated here defined on temporal graphs can be reduced to the same problems defined on temporal directed graphs. Thus we present positive results for the directed case, while the negative results are presented for the undirected case.

For clarity, we define in a more formal way some of the concepts presented in the introduction. A *time constraint* is a triple of functions $(\varepsilon, \omega_L, \omega_U)$, where $\varepsilon : E(G) \to [\tau]_0$ represents the *travel time* of each edge, $\omega_L : V(G) \to [\tau]_0$ represents the *minimum waiting time*, and $\omega_U : V(G) \to [\tau]_0$ represents the *maximum waiting time* of each vertex, with $\omega_L(v) \leq \omega_U(v)$ for every $v \in V(G)$. The *arrival time* of e is then equal to $\lambda(e) + \varepsilon(e)$; it is denoted by $\texttt{arr}(e)$. We suppose throughout the text that $\tau \geq \texttt{arr}(e)$ for every $e \in E(G)$.

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$, a time constraint $(\varepsilon, \omega_L, \omega_U)$, and a pair of vertices $s, z \in V(G)$, a temporal s, z-walk *respecting* $(\varepsilon, \omega_L, \omega_U)$ is a sequence $W = (s = v_1, e_1, v_2, e_2 \ldots, e_{p-1}, v_p = t)$ of alternating vertices and edges of G whose edges are traversed in a time-consistent way and whose time spent on each vertex respects the minimum and the maximum waiting time. Formally, W satisfies the following constraints:

- For every $i \in [p-1]$, we have $e_i = v_i v_{i+1}$;
- For every $i \in \{2, \ldots, p-1\}$, we have $\texttt{arr}(e_{i-1}) + \omega_L(v_i) \leq \lambda(e_i) \leq \texttt{arr}(e_{i-1}) + \omega_U(v_i)$. We say that W *waits* on v_i from $\texttt{arr}(e_{i-1})$ to $\lambda(e_i)$.

If $v_1, \ldots, v_p$ are all distinct, then we say that P is a temporal s, z-path *respecting* $(\varepsilon, \omega_L, \omega_U)$. From hereon, we only consider walks respecting $(\varepsilon, \omega_L, \omega_U)$, so we refrain from repeating it. We say that W contains the vertices $v_1, \ldots, v_p$ and the edges $e_1, \ldots, e_{p-1}$; we denote such sets by $V(W)$ and $E(W)$, respectively. Additionally, for each $i \in [p-1]$, we say that W *departs from* v_i *at time* $\lambda(e_i)$ and *arrives at* v_{i+1} *at time* $\texttt{arr}(e_i)$. Finally, the set of temporal vertices of W is given by

$$V^T(W) = \begin{array}{l} \{(v_1, \lambda(e_1)), (v_p, \texttt{arr}(e_{p-1}))\} \cup \\ \{(v_i, t) \mid i \in \{2, \ldots, p-1\} \text{ and } \texttt{arr}(e_{i-1}) \leq t \leq \lambda(e_i)\}. \end{array}$$

Given two temporal s, z-walks P and Q, we say that P and Q are *vertex disjoint* if they do not pass through a same vertex, i.e. $V(P) \cap V(Q) = \{s, z\}$, while we say that they are *t-vertex disjoint* if they do not pass through a same temporal vertex, i.e. $V^T(P) \cap V^T(Q) \subseteq \{s, z\} \times [\tau]$. Finally, a set $S \subseteq (V(G) \backslash \{s, z\}) \times [\tau]$ is a temporal t-vertex s, z-separator (respecting $(\varepsilon, \omega_L, \omega_U)$) if it intersects every temporal s, z-walk.

In the case that we have a pair of temporal graphs $\mathcal{G} = (G, \lambda)$ and $\mathcal{G}' = (G, \lambda')$ sharing the same base graph G, we may use a subscript (i.e., $V^T_{\mathcal{G}}$) to explictly refer to the context in which G is being used. For instance, a walk P in G might have distinct sets of temporal vertices in $\mathcal{G}$ and $\mathcal{G}'$, denoted by $V^T_{\mathcal{G}}$ and $V^T_{\mathcal{G}'}$, respectively.

In what follows, we analyse the complexity of all related problems when considering the possible values of $(\varepsilon, \omega_L, \omega_U)$. Given constant values t, ℓ, u with $\ell \leq u$, we call the (t, ℓ, u) *case* the problems related to the functions $\varepsilon \equiv t$, $\omega_L \equiv \ell$, and $\omega_U \equiv u$.

Given a temporal graph $\mathcal{G} = (G, \lambda)$ and a time constraint $(\varepsilon, \omega_L, \omega_U)$, we denote the maximum number of t-vertex disjoint temporal s, z-walks in $\mathcal{G}$ respecting $(\varepsilon, \omega_L, \omega_U)$ by $\texttt{tw}_{\mathcal{G}}^{(\varepsilon, \omega_L, \omega_U)}(s, z)$; and we denote the minimum size of a temporal t-vertex s, z-separator respecting $(\varepsilon, \omega_L, \omega_U)$ by $\texttt{tc}_{\mathcal{G}}^{(\varepsilon, \omega_L, \omega_U)}(s, z)$. We omit the subscript and/or superscript, whenever it is clear from the context.

The problem of computing those metrics can be easily reduced from the undirected to the directed case. Given an undirected temporal graph $(\mathcal{G}, \lambda)$, it suffices to generate a temporal directed graph $(\mathcal{G}', \lambda')$ by substituting every edge e of $\mathcal{G}$ by a pair of arcs in $\mathcal{G}'$ with the same endpoints, in both directions, and active at the same times as e. For brevity, we refer to $\mathcal{G}'$ as the *directed version* of $\mathcal{G}$.

3 Max Disjoint Walks is Hard

In this section we present our negative results concerning s, z-WTVD. In particular, we prove that deciding the existence of k t-vertex disjoint temporal s, z-walks respecting $(\varepsilon, \omega_L, \omega_U)$ is W[1]-hard when parameterized by k and para-NP-complete when parameterized by the lifetime. We start with the former. In order to maintain a brief presentation of the main results of this article, we give a proof of the next result for the particular $(1, 1, \tau)$ case while the complete version can be found in the full version.

Theorem 4. *Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and a pair of non-adjacent vertices $s, z \in V(G)$, for every fixed a, b, c such that $b < c$, deciding whether $\mathsf{tw}^{(a,b,c)}(s, z) \geq k$ is $\mathsf{W}[1]$-hard when parameterized by k.*

Proof. (Proof for the $(1, 1, \tau)$ *case.*] We construct an undirected temporal graph and the result for directed follows from constructing its directed version. We make a reduction from k-Independent Set parameterized by k. Let (G, k) be an instance of such problem and let $v^1, \ldots, v^n$ be an order of $V(G)$. We first construct an auxiliary temporal graph (H, λ'). Observe Fig. 2 to follow the construction. For each $v^\ell \in V(G)$, add vertices v_i^ℓ, $i \in [4]$ to H; denote the set $\{v_p^\ell \mid \ell \in [n]\}$ by V_p for each $p \in [4]$. Also, join v_1^ℓ to v_2^ℓ using a path with n new internal vertices, $w_1^\ell, \cdots, w_n^\ell$. Denote this path by P^ℓ, and denote the set $\{w_i^\ell \mid \ell, i \in [n]\}$ by W. Finally, let $\lambda'(w_i^\ell w_{i+1}^\ell) = 2(\ell + i + 1)$ for $i \in [n-1]$, $\lambda'(v_1^\ell w_1^\ell) = 2(\ell + 1)$, and $\lambda'(w_n^\ell v_2^\ell) = 2(\ell + n + 2)$.

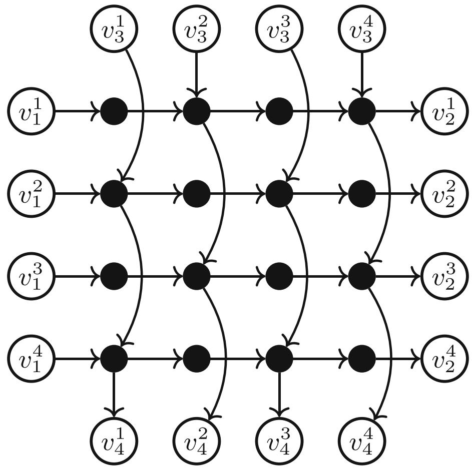

Fig. 2. Graph H related to the cycle on vertices $\{v^1, \cdots, v^4\}$ ordered as they appear in the cycle. Function λ' is constructed in a way that: each directed walk is a temporal walk respecting $(1, 1, \tau)$; and each intersection between a horizontal and a vertical walk is reflected in the intersection between the related sets of temporal vertices. Even though H is *undirected*, the edges are oriented in order to emphasize the order in which the edges are traversed in the temporal walks.

Now, we use the topology of G to define more paths using the vertices in W. The idea is that the ℓ-th vertex on each path P^i, where $i \in N(v^\ell)$, will represent the edge between v^ℓ and v^i. Formally, let $\ell \in [n]$ and let $N_G(v^\ell) = \{v^{\ell_1}, \ldots, v^{\ell_q}\}$ be the neighborhood of v^ℓ with $\ell_1 < \ell_2 < \cdots < \ell_q$. Then, add edges to form the path $(v_3^\ell, w_\ell^{\ell_1}, w_\ell^{\ell_2}, \ldots, w_\ell^{\ell_q}, v_4^\ell)$ and define $\lambda'(w_\ell^{\ell_p} w_\ell^{\ell_{p+1}}) = 2(\ell_p + \ell) + 1$, $\lambda'(v_3^\ell w_\ell^{\ell_1}) = 1$ and $\lambda'(w_\ell^{\ell_q} v_4^\ell) = 2(\ell_q + \ell) + 1$. Denote such temporal path by Q^ℓ. Observe that P^ℓ and Q^ℓ are temporal walks respecting $(1, 1, \tau)$. We prove that the following properties hold for the temporal graph (H, λ'):

1. If P is a temporal walk respecting $(1, 1, \tau)$ in (H, λ') and $e \in E(P) \cap E(P^\ell)$, then all the edges appearing in P after e are contained in P^ℓ; and

2. If Q is a temporal v_3^ℓ, v_4^ℓ-walk respecting $(1, 1, \tau)$ in (H, λ'), then Q and P^{ℓ_i} are t-vertex disjoint if and only if $v^{\ell_i} \notin N_G(v_\ell)$.

Now, we construct the final instance. Observe Fig. 3 to follow the construction. The idea is that all the copies of H on a given row of the construction will function to select one of the vertices of the independent set, with the last vertex being selected by the first column. Formally, for each row $i \in [k-1]$, create $k-i$ copies of H, each denoted by $H_{i,j}$ where $j \in [k-i]$. For each structure named X in H, we denote its copy in $H_{i,j}$ by $X(i, j)$. For instance, path P^ℓ is denoted by $P^\ell(i, j)$ and vertex v_1^ℓ is denoted by $v_1^\ell(i, j)$. Given sets $V_p(i, j)$ and $V_q(i', j')$, by *identification* of these sets we mean identifying pairs of vertices related to the same vertex of G; namely identifying $v_p^\ell(i, j)$ with $v_q^\ell(i', j')$ for every $\ell \in [n]$. We now identify some of these sets as follows.

- For each row i and each column $j \in [k-i-1]$, identify $V_2(i, j)$ and $V_1(i, j+1)$;
- For each column j and each row $i \in [k-j-1]$, identify $V_3(i, j)$ and $V_4(i+1, j)$; and
- Finally, for the diagonal identifications, do: for each $i \in [k-2]$, identify $V_3(i, k-i)$ with $V_2(i+1, k-(i+1))$.

Finally, we add vertices $s, s_1, \dots, s_k$, $z, z_1, \dots, z_k$, make s adjacent to every vertex s_j and z adjacent to every vertex z_j, $j \in [k]$. Also, for each $i \in [k-1]$, make s_i adjacent to every vertex in $V_1(i, 1)$ and make s_k adjacent to every vertex in $V_3(k-1, 1)$. Similarly, for each $j \in [k-1]$, make z_j adjacent to every vertex in $V_4(1, j)$ and make z_k adjacent to every vertex in $V_2(1, k-1)$. Denote the obtained graph by G' and observe that such graph is a simple graph. We define a timefunction λ in the following way. We number the copies of H from left to right and from bottom to top, starting from 0; denote by $t(i, j)$ the numbering of $H_{i,j}$. In this way, we have $t(k-1, 1) = 0$ and $t(1, k-1) = \binom{k}{2} - 1$. Now, let T be the lifetime of (H, λ'). For every $i \in [k-1]$, $j \in [k-i]$, and every edge $e \in E(H_{i,j})$, let $\lambda(e(i, j)) = \lambda'(e) + t(i, j)T + 3$. Also, for every $i \in [k]$, let $\lambda'(ss_i) = 1$ and $\lambda'(s_i v) = 3$ for every $v \in N(s_i) \setminus \{s\}$. Finally, let $T' = \binom{k}{2}T + 3$; hence T' is bigger than $\lambda'(e)$ for every e whose timefunction was assigned up to now. For each $j \in [k]$ define $\lambda'(z_j z) = T' + 2$, and for each $v \in N(z_j) \setminus \{z\}$, define $\lambda'(v z_j) = T'$. Observe that the directed paths in Fig. 3 are temporal paths. Also recall that the depicted directions are for presentation purposes only, the graph being constructed is undirected.

Because inside $H_{i,j}$, this time function is just a shifting of λ', this means that Properties 1 and 2 continue to hold. Moreover, given $i, j, i', j' \in [k]$, if $t(i', j') < t(i, j)$, then all the edges of $H_{i,j}$ are present later than any edge in $H_{i',j'}$. We call this the *increasing property*. To see that this reduction works, intuitively one needs to observe that each of the k temporal s, z-walks passes by a distinct s_i, then goes all the way to the diagonal using exactly the row related to the same v^{ℓ_i}, then enters the column related to v^{ℓ_i} until it reaches z_{k-i+1} and z. The set $\{v^{\ell_1}, \cdots, v^{\ell_k}\}$ is an independent set if and only if the related walks are t-vertex disjoint. $\square$

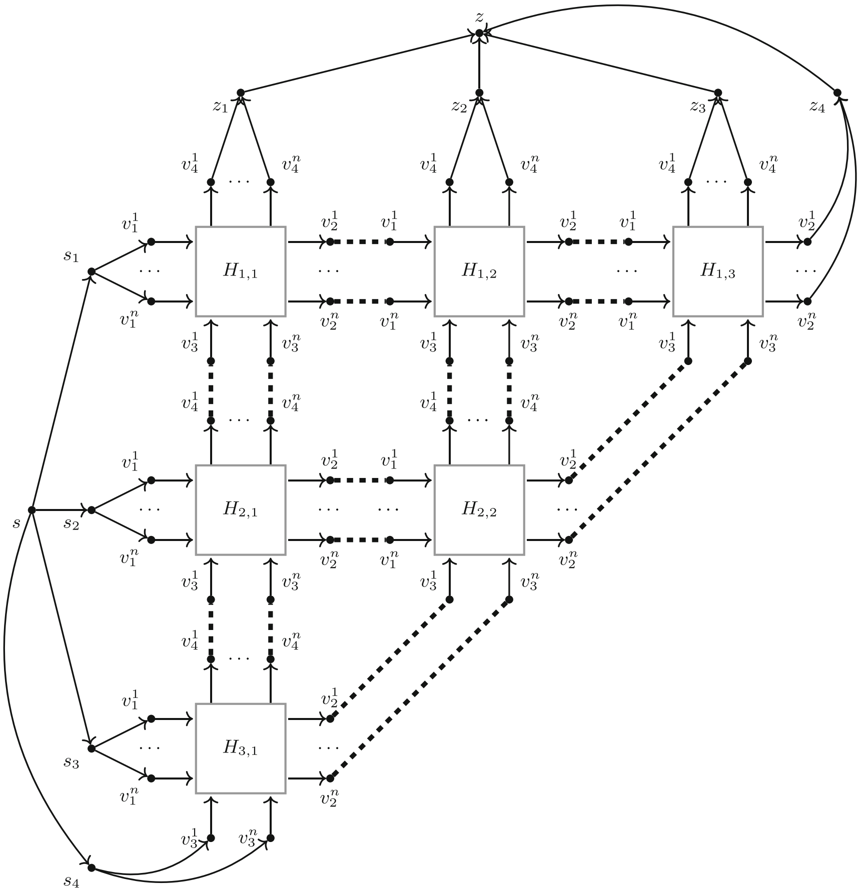

Fig. 3. Final construction for $k = 4$. We write only v_p^ℓ instead of $v_p^\ell(i,j)$ so the figure becomes cleaner. Thick dotted edges mean that the linked pair of vertices are going to be identified. The labels on the edges are omitted, but they are created in a way that the directed paths are temporal paths respecting $(1, 1, \tau)$.

Now we prove that s, z-WTVD is para-NP-complete when parameterized by the lifetime.

Theorem 5. *Let $\mathcal{G} = (G, \lambda)$ be a temporal graph with lifetime τ, s, z be a pair of vertices of G, and k be a positive integer. Deciding whether $\mathsf{tw}_{\mathcal{G}}^{(0,1,\tau)}(s, z) \geq k$ is NP-complete, even if τ is fixed, $\tau \geq 9$.*

Proof. We make a reduction from $3-\mathsf{SAT}$. Given a CNF formula ϕ on m clauses and n vertices, we construct a temporal graph $\mathcal{G}$ with two vertices $s, z \in V(\mathcal{G})$ such that ϕ is satisfiable if and only if there are at least $n + m$ t-vertex disjoint temporal s, z-walks respecting $(0, 1, \tau)$. We consider the variation where each variable appears at most twice positively and at most twice negatively, which is known to be NP-complete [10].

So let $\{c_1, \ldots, c_m\}$ be the set of clauses of ϕ, and $\{x_1, \ldots, x_n\}$, the set of variables. First add to $\mathcal{G}$ vertices s and z. Then, for each variable x_i do as follows (observe Fig. 4 to follow the construction). Add 4 new vertices, $\{s_i, z_i, a_i, x_i\}$, together with the edges ss_i active at time 1, z_iz active at time 9, s_ia_i active at time 5, $s_i x_i$ active at time 3, $a_i z_i$ active at time 5, $x_i z_i$ active at time 7, and two edges with endpoints $a_i x_i$, one active at time 4 and the other at time 6. We observe that if one wants to construct a digraph instead, then do so in a way that $(s_i, 5, a_i, 6, x_i, 7, z_i)$ and $(s_i, 3, x_i, 4, a_i, 5, z_i)$ are paths.

Denote by X_i the set of vertices $\{s_i, z_i, a_i, x_i\}$, and by H_i the temporal subgraph induced by $X_i \cup \{s, z\}$. Now, for each $j \in [m]$, add two new vertices s'_j and z'_j to $\mathcal{G}$, making s'_j adjacent to s at time 1 and z'_j adjacent to z at time 9. Finally, for each x_i appearing in c_j, add $e_{i,j} = s'_j x_i$ and $f_{i,j} = x_i z'_j$ and define their time-function as follows:

- If this is the 1st appearance of x_i, then let $\lambda(e_{i,j}) = 2$ and $\lambda(f_{i,j}) = 3$;
- If this is the 2nd appearance of x_i, then let $\lambda(e_{i,j}) = 4$ and $\lambda(f_{i,j}) = 5$;
- If this is the 1st appearance of $\neg x_i$, then let $\lambda(e_{i,j}) = 5$ and $\lambda(f_{i,j}) = 6$; and
- If this is the 2nd appearance of $\neg x_i$, then let $\lambda(e_{i,j}) = 7$ and $\lambda(f_{i,j}) = 8$.

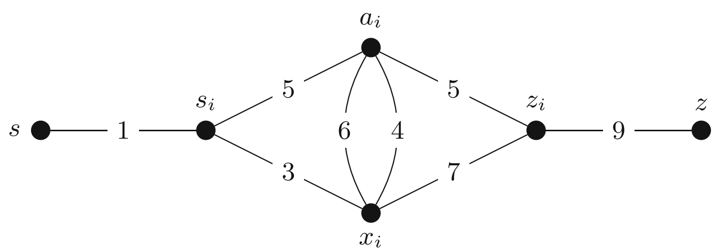

Fig. 4. Variable gadget in the proof that deciding $\mathtt{tw}^{(0,1,\tau)}{}_{\mathcal{G}}(s, z) \geq k$ is NP-complete.

The idea for the proof of correctness is that we need to get a distinct walk for each variable and each clause. Since every valid walk must go through some vertex x_i, the occurrence of x_i between timesteps 2 and 5 assigns the truth value, while its occurrence between timesteps 5 and 8 assigns the false value. First suppose that ϕ is satisfiable and consider a satisfying assignment of ϕ. For each $i \in [n]$, if x_i is true, then let P_i be the path $(s, 1, s_i, 5, a_i, 6, x_i, 7, z_i, 9, z)$; otherwise, let $P_i = (s, 1, s_i, 3, x_i, 4, a_i, 5, z_i, 9, z)$. Now, for each $j \in [m]$, let x_i be the variable related to the true literal in c_j. We analyse the cases below:

- If x_i appears for the 1st time in c_j, then let $P'_j = (s, 1, s'_j, 2, x_i, 3, z'_j, 9, z)$;
- If x_i appears for the 2nd time in c_j, then let $P'_j = (s, 1, s'_j, 4, x_i, 5, z'_j, 9, z)$;
- If $\neg x_i$ appears for the 1st time in c_j, then let $P'_j = (s, 1, s'_j, 5, x_i, 6, z'_j, 9, z)$; and
- If $\neg x_i$ appears for the 2nd time in c_j, then let $P'_j = (s, 1, s'_j, 7, x_i, 8, z'_j, 9, z)$.

We now prove that $P_1, \ldots, P_n, P'_1, \ldots, P'_m$ are t-vertex disjoint (observe that they clearly respect $(0, 1, \tau)$). Paths $P_1, \ldots, P_n$ are vertex disjoint since each P_i is contained in H_i and $X_i \cap X_j = \emptyset$ for every $i \neq j$. Hence they are also t-vertex disjoint. Now consider $j, \ell \in [m]$ with $j \neq \ell$. If P'_j and P'_ℓ intersect on a vertex, then it is because the literal validating P'_j is the same literal validating

P'_ℓ, say $\ell_i \in \{x_i, \neg x_i\}$. Hence, they are vertex disjoint with any other path in $\{P'_1, \ldots, P'_m\} \setminus \{P'_j, P'_\ell\}$, and they are t-vertex disjoint among themselves by construction, since one of them contains the 1st occurrence of ℓ_i, while the other contains the 2nd occurrence of ℓ_i. Finally, if x_i is true, then P_i contains $\{(x_i, 6), (x_i, 7)\}$, while the paths in $P'_1, \ldots, P'_m$ containing x_i are the ones defined in the first two items and, hence, do not intersect P_i (recall that the definition of P'_j is according to a true literal appearing in P'_j). A similar argument can be applied in case P_i is false to conclude that $V^T(P_i) \cap V^T(P'_j) = \emptyset$ for every $i \in [n]$ and $j \in [m]$.

Now, suppose that $\mathcal{G}$ has $n+m$ t-vertex disjoint temporal s,z-walks respecting $(0, 1, \tau)$. Observe that, since s has exactly $n+m$ temporal edges incident to it (as well as z), such walks must start with such edges. For each $i \in [n]$, call P_i the walk starting with $(ss_i, 1)$, and for each $j \in [m]$, call P'_j the walk starting with $(ss'_j, 1)$. For each $i \in [n]$, set x_i to false if $(s_i x_i, 3)$ is in P_i, and to true otherwise. We want to argue that this is a satisfying assignment. Suppose otherwise and let $j \in [m]$ be such that all literals of P'_j are set to false. Also, let $\ell_i \in \{x_i, \neg x_i\}$ be the literal of c_j corresponding to the edge being used in P'_j, i.e., such that $s'_j x_i \in E(P'_j)$. If $\ell_i = x_i$, then x_i is set to false, and hence $(s_i x_i, 3)$ is in P_i. As P_i respects $(0, 1, \tau)$, we get that $\{(x_i, 3), (x_i, 4)\} \subseteq V^T(P_i)$. Since P'_j also respects $(0, 1, \tau)$, we get that either $(z'_j x_i, 2) \in E^T(P'_j)$, in which case $(x_i, 3) \in V^T(P'_j) \cap V^T(P_i)$, or $(z'_j x_i, 4) \in E^T(P'_j)$ and hence $(x_i, 4) \in V^T(P'_j) \cap V^T(P_i)$. In both cases we get a contradiction as P_i and P'_j are t-vertex disjoint. And if $\ell_i = \neg x_i$, then x_i is set to true, and hence $(s_i x_i, 3) \notin E^T(P_i)$. In this case, we must have $\{(s_i a_i, 5), (a_i x_i, 6)\} \subseteq E^T(P_i)$, since P_i respects $(0, 1, \tau)$ and hence cannot use $(a_i z_i, 5)$ after $(s_i a_i, 5)$. Note that this implies that $\{(x_i, 6), (x_i, 7)\} \subseteq V^T(P_i)$. Now, if P'_j contains $(s'_j x_i, 5)$, then, since P'_j respects $(0, 1, \tau)$, we get that $(x_i, 6) \in V^T(P'_j) \cap V^T(P_i)$, a contradiction. And if P'_j contains $(s'_j x_i, 7)$, we get again a contradiction as $(x_i, 7) \in V^T(P'_j) \cap V^T(P_i)$. □

4 Min Separator is NP-Complete

In this section we provide our negative result for s,z-WTVS. We give a reduction from Vertex Cover, which is known to be NP-complete. In this problem, given a graph G and a positive integer k, the goal is to decide if there is a $C \subseteq V(G)$ of size at most k such that every edge of G has at least one endpoint in C. Such a set is called a *vertex cover of* G.

Theorem 6. *Let $\mathcal{G} = (G, \lambda)$ be a temporal graph with lifetime τ, s, z be a pair of vertices of G, and k be a positive integer. For every fixed a, b, c with $a + b > 0$ and $c \geq 1 + 2a + 3b$, deciding whether $\mathtt{tc}_{\mathcal{G}}^{(a,b,c)}(s, z) \leq k$ is NP-complete, even for fixed τ.*

Proof. The idea is that we spread the timesteps in order to make space for the time constraints. Let (G, k) be an instance of Vertex Cover. Define $\mathcal{G}$ as follows. Let $V(\mathcal{G})$ be equal to $V(G) \cup F \cup \{s, z\}$, where $F = \{f_{uv}, f_{vu} \mid uv \in$

$E(G)\}$. Then, for each vertex $u \in V(G)$, add two edges from s to u, one active in timestep 1 and the other in timestep $2+b$, and two edges from u to z, one active in timestep $2+a+b$ and the other in timestep $2+3a+3b$. Also, for each edge $uv \in E(G)$, add an edge from u to f_{uv} and from v to f_{vu} both active in timestep $1+a+b$, and an edge from f_{uv} to v and from f_{vu} to u both active in timestep $2+2a+2b$. Observe that the lifetime of $\mathcal{G}$ must be equal to at least the maximum label of an edge plus a; so define τ to be equal to $2+4a+3b$. Additionally, as we will see, all important walks will wait on a vertex for at most $1+2a+3b$; hence the condition on c. Now we prove that G has a vertex cover of size at most k if and only if $\mathcal{G}$ has a temporal t-vertex s,z-separator respecting (a,b,c) of size at most $n+k$, where $n = |V(G)|$. See Fig. 5 for an illustration of this construction for the particular case $(0,1,\tau)$. For the general case, one can think of timesteps 2, 4, and 5 in the figure as $1+a+b$, $2+2a+2b$, and $2+3a+3b$, respectively. Additionally, timestep 3 is split into two: timestep $2+b$ containing the edges of timestep 3 incident to s and timestep $2+a+b$ containing the edges incident to z.

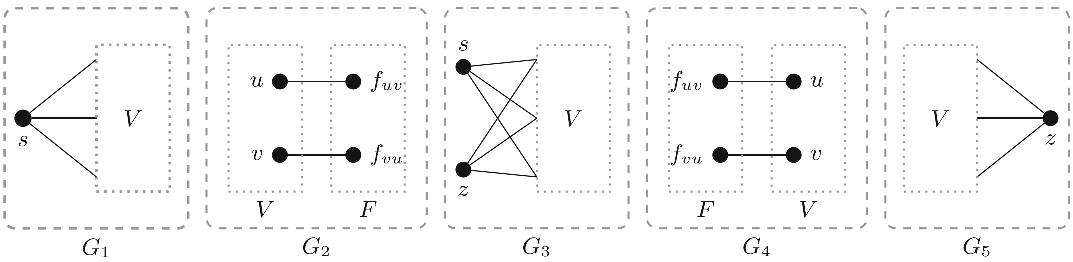

Fig. 5. Reduction from VERTEX COVER. We denote $V(G)$ by V.

First suppose that $S \subseteq V(G)$ is a vertex cover of size at most k, and define $S' = \{(u, 2+a+b) \mid u \notin S\} \cup \{(u, 1+a+b), (u, 2+2a+2b) \mid u \in S\}$. Note that S' has size at most $n+k$ as each vertex has at least one copy in S' and the vertices in S have two. To see that S' is a temporal t-vertex s,t-separator, let P be any temporal s,t-walk respecting (a,b,c). If $V(P) \cap F = \emptyset$, then P is one of the following paths, for some $u \in V(G)$: $P_1 = (s,1,u,2+a+b,z)$, $P_2 = (s,2+b,u,2+3a+3b,z)$, or $P_3 = (s,1,u,2+3a+3b,z)$. If $u \notin S$, then S' clearly intersects each of these paths as it contains $(u,2+a+b)$. Otherwise, if $u \in S$, then: S' intersects P_1 and P_3 as it contains $(u,1+a+b)$; and S' intersects P_2 too as it contains $(u,2+2a+2b)$.

Now, suppose that $f_{uv} \in V(P) \cap F$ and observe that, since each vertex in F has degree exactly two, we must have $P = (s,1,u,1+a+b,f_{uv},2+2a+2b,v,2+3a+3b,z)$. As S is a vertex cover, we get that $\{u,v\} \cap S \neq \emptyset$. If $u \in S$, then $(u,1+a+b) \in S' \cap V^T(P)$. And if $v \in S$, then $(v,2+2a+2b) \in S' \cap V^T(P)$. In each situation we get that S' intersects $V^T(P)$; therefore S' is a temporal t-vertex s,z-separator.

Now let S' be temporal t-vertex s,z-separator respecting (a,b,c) of size at most $n+k$. Recall the existence of paths P_1, P_2, P_3 in the previous paragraph.

Observe that this implies that S' must contain at least one copy of u for every $u \in V(G)$. Additionally, if S' contains exactly one copy of u, then observe that it must contain $(u, 2+a+b)$ as this is the only t-vertex in $V^T(P_1) \cap V^T(P_2)$. Indeed, P_1 leaves u at time $2+a+b$, while P_2 arrives in u at time $2+a+b$. Now, let U be the set of vertices such that S' contains only $(u, 2+a+b)$; formally, $U = \{u \in V(G) \mid S' \cap (\{u\} \times [\tau]) = \{(u, 2+a+b)\}\}$. Define $S = V(G) \setminus U$. We argue that S is a vertex cover of size at most k. First note that, because S' contains at least two copies of each $u \in S$ and since S' has size at most $n+k$, we get that S' contains exactly two copies of each $u \in S$ and $|S| \leq k$. Note that this also implies that S' does not contain any copy of a vertex in F. Now, consider any edge $uv \in E(G)$ and let $P'_1 = (s, 1, u, 1{+}a{+}b, f_{uv}, 2{+}2a{+}2b, v, 2{+}3a{+}3b, z)$. Observe that P'_1 does not contain $(u, 2+a+b)$ as it leaves earlier from u; also, it does not contain $(v, 2+a+b)$ as it arrives in v at time $2+3a+2b$ which is bigger than $2+a+b$ as $a+b > 0$. Therefore, if $u \in U$, then v cannot be in U as in such case P'_1 would not be intersected by S'. In other words, at least one between u and v must be in S, as we wanted to prove. □

5 XP Algorithm for Disjoint Walks

In this section we show our XP algorithm for WTVD in temporal directed graphs, where the paths respect a triple of appropriate functions $(\varepsilon, \omega_L, \omega_U)$. The proof happens in two steps: first, we reduce the problem to the case where $\varepsilon \equiv 0$ and every edge of the temporal directed graph is active exactly once. Then, we apply another reduction that allows us to solve the problem by deciding if a given digraph H contains a path from a special vertex s^* to another special vertex z^*.

The proof is roughly inspired in the XP algorithm for linkages in acyclic digraphs given in [11], in which H contains one vertex for every k-tuple of vertices of the input digraph and the edges of H are chosen in such a way that one can only leave a vertex v of H if v cannot be reached by any other vertex contained in the paths constructed so far. In the end, a path P in H from $(s_1, \dots, s_k)$ to $(z_1, \dots, z_k)$ can be easily translated into a set of walks from s_i to t_i by noticing that each edge of P is associated with the change of exactly one coordinate of the current k-tuple. A fundamental property of this algorithm is that, in acyclic digraphs, any k-tuple of vertices $(v_1, \dots, v_k)$ contains a vertex v_i that cannot be reached by any other vertex of the k-tuple. This is used to ensure that the constructed walks are indeed disjoint.

When working with temporal graphs this property is not guaranteed to hold: it is possible that every vertex of a k-tuple contains an incoming temporal edge from another vertex of the same k-tuple, even when $\omega_L(v) \geq 1$ for every vertex v of the temporal directed graph. Thus to ensure that we can only build t-vertex disjoint walks, we focus on k-tuples of edges instead of vertices and impose more advanced rules to the system that is used to decide when to add edges to H.

Theorem 7. *Let $\mathcal{G} = (D, \lambda)$ be a temporal directed graph with lifetime τ, $(\varepsilon, \omega_L, \omega_U)$ be a time constraint where $\omega_L(u) \geq 1$ for every $u \in V(D)$, and $\{(s_1, z_1),$*

$\ldots, (s_k, z_k)\}$ be a set of vertex pairs of D. Then in time $|E(D)|^{\mathcal{O}(k)}$ one can decide whether there exists t-vertex disjoint temporal walks $P_1, \ldots, P_k$ such that each P_i is a temporal s_i, z_i-walk in (D, λ) respecting $(\varepsilon, \omega_L, \omega_U)$.

Acknowledgements. This was partially funded by Italian PNRR CN4 Centro Nazionale per la Mobilità Sostenibile, NextGeneration EU - CUP, B13C22001000001. MUR of Italy, under PRIN Project n. 2022ME9Z78 - NextGRAAL: Next-generation algorithms for constrained GRAph visuALization, PRIN PNRR Project n. P2022NZPJA - DLT-FRUIT: A user centered framework for facilitating DLTs FRUITion, and French project ELIT (ANR-20-CE48-0008-01).

References

1. Bentert, M., Himmel, A.-S., Nichterlein, A., Niedermeier, R.: Efficient computation of optimal temporal walks under waiting-time constraints. Appl. Netw. Sci. **5**(1), 1–26 (2020). https://doi.org/10.1007/s41109-020-00311-0
2. Berman, K.A.: Vulnerability of scheduled networks and a generalization of Menger's Theorem. Netw.: Int. J. **28**(3), 125–134 (1996)
3. Brunelli, F.: Algorithmic aspects of reachability in temporal graphs. (Aspects algorithmiques de l'atteignabilité dans les graphes). Ph.D. thesis, Paris Cité University, France (2023)
4. Brunelli, F., Viennot, L.: Minimum-cost temporal walks under waiting-time constraints in linear time. CoRR **abs/2211.12136** (2022)
5. Brunelli, F., Viennot, L.: Computing temporal reachability under waiting-time constraints in linear time. In: 2nd Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2023. LIPIcs, vol. 257, pp. 4:1–4:11 (2023)
6. Casteigts, A., Corsini, T., Hocquard, H., Labourel, A.: Robustness of distances and diameter in a fragile network. In: 1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 221, pp. 9:1–9:16 (2022)
7. Casteigts, A., Himmel, A., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. In: 31st International Symposium on Algorithms and Computation, ISAAC 2020. LIPIcs, vol. 181, pp. 30:1–30:18 (2020)
8. Casteigts, A., Himmel, A., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. Algorithmica **83**(9), 2754–2802 (2021)
9. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D.: Parameterized Algorithms. Springer, Cham (2015)
10. Darmann, A., Döcker, J.: On simplified np-complete variants of monotone 3-sat. Discret. Appl. Math. **292**, 45–58 (2021)
11. Fortune, S., Hopcroft, J.E., Wyllie, J.: The directed subgraph homeomorphism problem. Theoret. Comput. Sci. **10**, 111–121 (1980)
12. Füchsle, E., Molter, H., Niedermeier, R., Renken, M.: Delay-robust routes in temporal graphs. In: 39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022. LIPIcs, vol. 219, pp. 30:1–30:15 (2022)
13. Füchsle, E., Molter, H., Niedermeier, R., Renken, M.: Temporal connectivity: coping with foreseen and unforeseen delays. In: 1st Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2022. LIPIcs, vol. 221, pp. 17:1–17:17 (2022)

14. Holme, P.: Modern temporal network theory: a colloquium. Eur. Phys. J. B **88**(9), 1–30 (2015). https://doi.org/10.1140/epjb/e2015-60657-4
15. Ibiapina, A., Lopes, R., Marino, A., Silva, A.: Menger's theorem for temporal paths (not walks). CoRR **abs/2206.15251** (2022)
16. Ibiapina, A., Silva, A.: Snapshot disjointness in temporal graphs. arXiv preprint arXiv:2302.06653 (2023)
17. Ibiapina, A., Silva, A.: Mengerian graphs: characterization and recognition. J. Comput. Syst. Sci. **139**, 103467 (2024)
18. Karp, R.M.: On the computational complexity of combinatorial problems. Networks **5**(1), 45–68 (1975)
19. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. **64**, 820–842 (2002)
20. Klobas, N., Mertzios, G.B., Molter, H., Niedermeier, R., Zschoche, P.: Interference-free walks in time: temporally disjoint paths. Auton. Agents Multi Agent Syst. **37**(1), 1 (2023)
21. Kunz, P., Molter, H., Zehavi, M.: In which graph structures can we efficiently find temporally disjoint paths and walks? In: IJCAI, pp. 180–188. ijcai.org (2023)
22. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Soc. Netw. Anal. Min. **8**(1), 1–29 (2018). https://doi.org/10.1007/s13278-018-0537-7
23. Maack, N., Molter, H., Niedermeier, R., Renken, M.: On finding separators in temporal split and permutation graphs. J. Comput. Syst. Sci. **135**, 1–14 (2023)
24. Marino, A., Silva, A.: Paths and Connectivity in Temporal Graphs. Editora do IMPA, 1 edn. (2023)
25. Menger, K.: Zur allgemeinen kurventheorie. Fundam. Math. **10**(1), 96–115 (1927)
26. Mertzios, G.B., Michail, O., Spirakis, P.G.: Temporal network optimization subject to connectivity constraints. Algorithmica **81**(4), 1416–1449 (2019)
27. Michail, O.: An introduction to temporal graphs: an algorithmic perspective. Internet Math. **12**(4), 239–280 (2016)
28. Molter, H.: The complexity of finding temporal separators under waiting time constraints. Inf. Process. Lett. **175**, 106229 (2022)
29. Nicosia, V., Tang, J., Mascolo, C., Musolesi, M., Russo, G., Latora, V.: Graph metrics for temporal networks. In: Temporal networks, pp. 15–40. Springer (2013)
30. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Comb. Theory, Ser. B **63**(1), 65–110 (1995)
31. Stern, R.: Multi-agent path finding–an overview. In: Artificial Intelligence: 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures, pp. 96–115 (2019)
32. Stern, R., et al.: Multi-agent pathfinding: definitions, variants, and benchmarks. In: Twelfth Annual Symposium on Combinatorial Search (2019)
33. Thejaswi, S., Gionis, A.: Restless reachability in temporal graphs. CoRR **abs/2010.08423** (2020)
34. Zschoche, P.: Restless temporal path parameterized above lower bounds. In: 40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023. LIPIcs, vol. 254, pp. 55:1–55:16 (2023)
35. Zschoche, P., Fluschnik, T., Molter, H., Niedermeier, R.: The complexity of finding small separators in temporal graphs. J. Comput. Syst. Sci. **107**, 72–92 (2020)