# Temporal Dominating Set and Temporal Vertex Cover Under the Lense of Degree Restrictions

**Anton Herrmann**
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

**Christian Komusiewicz**
Institute of Computer Science, Friedrich Schiller University Jena, Germany

**Nils Morawietz**
Institute of Computer Science, Friedrich Schiller University Jena, Germany
LaBRI, Université de Bordeaux, France

**Frank Sommer**
Institute of Logic and Computation, TU Wien, Austria

## Abstract

We study the Temporal Dominating Set problem, in which one asks whether a temporal graph $\mathcal{G} = (G_1, \dots, G_T)$ given as a sequence of snapshot graphs, over the same vertex set $V$, has a set $S$ of temporal vertices of size at most $k$ such that each vertex $v$ of $V$ is dominated by some $w \in S$ in the snapshot that contains $w$. Additionally, we consider Temporal Partial Dominating Set, where one asks whether at least $t$ (and not necessarily all) vertices of $V$ can be dominated by $S$ and a further generalization in which the solution may only contain a bounded number of temporal vertices from each snapshot.

We analyze how the complexity of Temporal (Partial) Dominating Set is influenced by the maximum snapshot degree and the structure of the underlying graph, the graph with vertex set $V$ and whose edge set is the union of all snapshot edge sets. For example, we obtain a complexity dichotomy for the maximum snapshot degree and we show that Temporal Partial Dominating Set is fixed-parameter tractable for $\text{tw} + \Delta$, where tw and $\Delta$ denote the treewidth and the maximum degree of the underlying graph of $\mathcal{G}$, respectively. We also show which of our results transfer to the well-studied Temporal Vertex Cover problem. For example, we show that Temporal Vertex Cover is also fixed-parameter tractable for $\text{tw} + \Delta$ which substantially extends the previously known polynomial-time algorithms for the case that the underlying graph is a path or cycle.

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

**Keywords and phrases** NP-hard problem, FPT-algorithm, Treewidth, Color coding

**Digital Object Identifier** 10.4230/LIPIcs.SAND.2025.16

**Funding** *Nils Morawietz*: Partially supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).
*Frank Sommer*: Supported by the Alexander von Humboldt Foundation.

**Acknowledgements** Some of the results of this work are also contained in the first author's Master thesis [23].

## 1 Introduction

Many real-world situations such as friendships in a social network that may change over time are naturally modeled by temporal graphs which have a dynamic set of edges and thus allow to capture changing interactions and relations between the network entities [24, 29]. There are many equivalent ways of formalizing temporal graphs; in this work, we view them as finite sequences $\mathcal{G} := (G_1, \ldots, G_T)$ of (static) graphs, called *snapshots*, where all graphs in the sequence have the same vertex set. The vertices in the snapshots are *temporal vertices* and the graph $G_\downarrow$ obtained by taking the union over all edge sets is the *underlying graph*.

To better understand the algorithmic challenges posed by analyzing and using temporal graphs, different variants of classical graph problems have been lifted to the temporal setting. An example is the famous NP-hard Vertex Cover problem, where one is given an undirected graph $G$ and the task is to find a small vertex cover, that is, a vertex set $S$ such that each edge has an endpoint in $S$. The complexity of Vertex Cover has been studied from many different viewpoints. In particular, it is the main object of research in parameterized algorithmics [22, 6, 25, 32]. A canonical temporal version of Vertex Cover is the Temporal Vertex Cover problem, introduced by Akrida et al. [1]. A temporal vertex is a vertex in a specific snapshot. In Temporal Vertex Cover we are looking for a small *temporal vertex cover*, which is a set $\mathcal{S}$ of temporal vertices such that each edge $e$ of the underlying graph $G_\downarrow$ is covered by some temporal vertex $w$ in $\mathcal{S}$, that is, $w$ is an endpoint of $e$ and $e$ is present in the snapshot of $w$ (see Figure 1 for an example).

Temporal Vertex Cover (TVC)
**Input:** A temporal graph $\mathcal{G}$ and an integer $k$.
**Question:** Is there a temporal vertex cover $\mathcal{S}$ of $\mathcal{G}$ with $|\mathcal{S}| \leq k$

As shown in a series of works that are discussed in more detail below, TVC turned out to be quite hard when compared to Vertex Cover, even for inputs with a heavily restricted structure.

In this work, we want to broaden our algorithmic knowledge for temporal versions of such classic graph problems. To this end, we study a temporal variant of the NP-hard Dominating Set problem, which has a similarly central role in algorithmic research. In Dominating Set the task is to find, for a given graph $G$, a small set of vertices $S$ such that every vertex is *dominated* by $S$ which means that it is either in $S$ or has at least one neighbor in $S$. Interestingly, Dominating Set turns out to be more difficult than Vertex Cover in several aspects such as approximability or parameterized complexity [7, 36]. This motivates the exploration of temporal Dominating Set variants, analogous to those for Vertex Cover, with the goal of analyzing and comparing their complexities.

There are several ways of extending domination to the temporal setting [5]. In analogy to TVC, we consider the version where we say that a vertex $u$ in the underlying graph $G_\downarrow$ is *dominated* by a temporal vertex $v$ if $u$ is contained in the closed neighborhood of $v$ in the snapshot of $v$. A set of temporal vertices $\mathcal{S}$ is then a *temporal dominating set* if each vertex of the underlying graph is dominated by at least one temporal vertex of $\mathcal{S}$ (see Figure 1 for an example). If we now aim to find a small temporal dominating set, we arrive at the following problem.

Temporal Dominating Set (TDS)
**Input:** A temporal graph $\mathcal{G}$ and an integer $k$.
**Question:** Is there a temporal dominating set $\mathcal{S}$ of $\mathcal{G}$ with $|\mathcal{S}| \leq k$?

A natural and well-studied generalization of Dominating Set is Partial Dominating Set where we ask for a small set that dominates many but not necessarily all vertices of the graph. In our temporal setting, this gives the following problem.

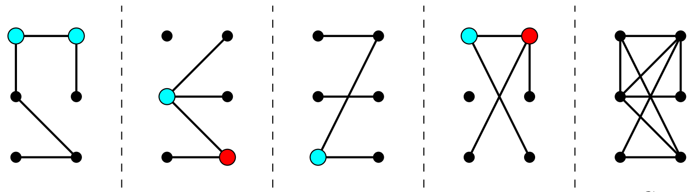


**Figure 1** The five blue temporal vertices are a temporal vertex cover of size 5 and the two red temporal vertices are a temporal dominating set of size 2.

Temporal Partial Dominating Set (TPDS)
**Input:** A temporal graph $\mathcal{G}$ and integers $k, t$.
**Question:** Is there a temporal vertex set $\mathcal{S}$ with $|\mathcal{S}| \leq k$ which dominates at least $t$ vertices of the underlying graph $G_\downarrow$?

Moreover, we consider Temporal (Partial) Dominating Set under *budget constraints* for each snapshot. Here, the solution is allowed to contain at most $b$ temporal vertices from each snapshot, for some given number $b$. We refer to the two resulting problems as Budget-TDS and Budget-TPDS, respectively.

As mentioned above, one aim of our study is to compare the complexity of TDS to TVC. To obtain a full picture, we also consider the partial and budget versions of TVC. In the most general problem, Budget-Temporal Partial Vertex Cover (Budget-TPVC), the question is whether we can cover at least $t$ edges of the underlying graph $G_\downarrow$, by selecting at most $k$ temporal vertices, with at most $b$ from each snapshot. The two intermediate problems, Temporal Partial Vertex Cover (TPVC) and Budget-Temporal Vertex Cover (Budget-TVC) are defined analogously.

We analyze the complexity of all problems with respect to the structure of the underlying graph $G_\downarrow$, the maximum snapshot degree, and from the viewpoint of parameterized complexity. Before stating our results, we give a brief overview of the literature on the considered and related problems.

**Known Results and Further Related Work.** For temporal graphs with only one snapshot, TVC and TDS coincide with Vertex Cover and Dominating Set, respectively. This also holds for the partial variants. Hence, all temporal problems considered in this work inherit all hardness results that are known for their static counterparts. In particular, TVC and TDS are NP-hard even when the underlying graph $G_\downarrow$ has maximum degree 3 [18]. In addition, the following is known: TVC is already NP-hard when the underlying graph is a star graph [1]. The reduction also shows that TVC is W[2]-hard with respect to $k$ in that case [1]. Hence, it is unlikely that the problem can be solved in $f(k) \cdot |\mathcal{G}|^{\mathcal{O}(1)}$ time. On the positive side, TVC is solvable in polynomial time if the underlying graph is a path or cycle [21].

Akrida et al. [1] also introduced Sliding Window Temporal Vertex Cover (SW-TVC) where the input additionally contains an integer $\delta$, and the aim is to cover every edge at least once in every window of $\delta$ consecutive snapshots. In contrast to TVC, SW-TVC is NP-hard if the underlying graph is a path [21]. Moreover, SW-TVC is FPT for $k + \delta$, that is, it can be solved in $f(k + \delta) \cdot |\mathcal{G}|^{\mathcal{O}(1)}$ time [21]. Finally, Hamm et al. [21] considered a partial variant of SW-TVC where "partial" refers to the fact that each edge needs to be covered in only a given subset of the windows. This variant thus differs substantially from TPVC as defined here.

**Table 1** Overview of our results on the classic and parameterized complexity for Temporal Dominating Set and its variants. Herein, tw and $\Delta$ denote the treewidth and the maximum degree of the underlying graph $G_{\downarrow}$, respectively, $\Delta_{\max}$ denotes the maximum degree in any snapshot of the input graph, and $h(\mathcal{G})$ denotes the temporal $h$-index of $\mathcal{G}$.

| | TDS | TPDS | Budget-TDS | Budget-TPDS |
|---|---|---|---|---|
| $G_{\downarrow}$ is a star | NP-h. Corollary 3.2 | NP-h. Corollary 3.2 | NP-h. Corollary 3.2 | NP-h. Corollary 3.2 |
| $G_{\downarrow}$ is a clique | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 |
| $G_{\downarrow}$ is a path | Poly Thm. 4.6 | Poly Thm. 4.6 | NP-h. Thm. 3.7 | NP-h. Thm. 3.7 |
| $\Delta_{\max} \leq 2$ | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 | NP-h. Thm. 3.3 |
| $\Delta_{\max} \leq 1$ | Poly Thm. 3.5 | Poly Thm. 3.5 | NP-h. Thm. 3.7 | NP-h. Thm. 3.7 |
| $\mathrm{tw} + \Delta$ | FPT Thm. 4.6 | FPT Thm. 4.6 | paraNP-h. Thm. 3.7 | paraNP-h. Thm. 3.7 |
| $t$ | - | FPT Thm. 4.2 | - | FPT Thm. 4.2 |
| $k + h(\mathcal{G})$ | FPT Thm. 4.5 | FPT Thm. 4.5 | FPT Thm. 4.5 | FPT Thm. 4.5 |

There are further temporal variants of Vertex Cover and Dominating Set which have been discussed and studied. These include a multistage variant of Vertex Cover [15], a version with activity intervals for the selected vertices [17, 9, 11, 10, 8, 12, 33, 34], a reachability-based variant for Dominating Set [26] and different conditions on how and when a vertex should be dominated [5, 35, 37].

**Our Contribution.** In a nutshell, we show that TDS is considerably harder than its static counterpart. Moreover, we observe that introducing budgets makes it harder to exploit a bounded degree of the underlying graph, and that TDS and TVC are similarly difficult, with some border cases where TVC is easy and TDS is hard. More precisely, we obtain the following results touching on three main aspects of the input structure; see Tables 1 and 2 for an overview for TDS and TVC, respectively.

The first aspect is the global structure of the underlying graph $G_{\downarrow}$. We show that, like TVC, TDS is already NP-hard if $G_{\downarrow}$ is restricted to be a star. Moreover, we show that TVC and TDS are hard if $G_{\downarrow}$ is restricted to be a clique. For $G_{\downarrow}$ restricted to a path, we show that all problem variants without budget constraints are easy but adding budgets to the snapshots makes the problems NP-hard.

The second aspect on which we focus is the maximum snapshot degree $\Delta_{\max}$ which is defined as the maximum of the maximum degrees of all snapshot graphs. Here, we obtain a complexity dichotomy for all problem variants. In particular, we show that TDS is NP-hard for $\Delta_{\max} = 2$ and that for $\Delta_{\max} = 1$ the non-budgeted variants are polynomial-time solvable and the variants with budget are NP-hard. For TVC the situation is different, here $\Delta_{\max} = 3$ leads to hardness for all variants, $\Delta_{\max} = 1$ leads to polynomial-time solvability for all variants, and for $\Delta_{\max} = 2$ we observe that adding budgets makes the problem hard.

The third aspect is the parameterized complexity of the problems. We show that all problem variants without budgets are tractable with respect to the combination of the treewidth of $G_{\downarrow}$ and the maximum degree of $G_{\downarrow}$. For the variants with budget, this problem parameterization is already excluded by the NP-hardness for the case that $G_{\downarrow}$ is a path. For the partial problem variants, we then show fixed-parameter tractability for $t$, the number of

**Table 2** Overview of our results on the classic and parameterized complexity for Temporal Vertex Cover and its variants. Herein, tw and $\Delta$ denote the treewidth and the maximum degree of the underlying graph $G_\downarrow$, respectively, $\Delta_{\max}$ denotes the maximum degree in any snapshot of the input graph, and $h(\mathcal{G})$ denotes the temporal $h$-index of $\mathcal{G}$.

| | TVC | TPVC | Budget-TVC | Budget-TPVC |
|---|---|---|---|---|
| $G_\downarrow$ is a star | NP-h. [1, Thm. 2] | NP-h. [1, Thm. 2] | NP-h. [1, Thm. 2] | NP-h. [1, Thm. 2] |
| $G_\downarrow$ is a clique | NP-h Thm. 3.4 | NP-h Thm. 3.4 | NP-h Thm. 3.4 | NP-h Thm. 3.4 |
| $G_\downarrow$ is a path | Poly [21, Thm. 5] | Poly Thm. 3.6 | NP-h Thm. 3.8 | NP-h Thm. 3.8 |
| $\Delta_{\max} \leq 3$ | NP-h Thm. 3.4 | NP-h Thm. 3.4 | NP-h Thm. 3.4 | NP-h Thm. 3.4 |
| $\Delta_{\max} \leq 2$ | Poly Thm. 3.6 | Poly Thm. 3.6 | NP-h Thm. 3.8 | NP-h Thm. 3.8 |
| $\Delta_{\max} \leq 1$ | Poly Thm. 3.9 | Poly Thm. 3.9 | Poly Thm. 3.9 | Poly Thm. 3.9 |
| tw $+\Delta$ | FPT Prop. 4.7 | FPT Prop. 4.7 | paraNP-h. Thm. 3.8 | paraNP-h. Thm. 3.8 |
| $t$ | - | FPT Corollary 4.3 | - | FPT Corollary 4.3 |
| $k + h(\mathcal{G})$ | FPT Thm. 4.5 | FPT Thm. 4.5 | FPT Thm. 4.5 | FPT Thm. 4.5 |

elements to be covered. This implies also fixed-parameter tractability with respect to $k+\Delta_{\max}$, since instances with $t > k \cdot (\Delta_{\max} + 1)$ are trivial no-instances. Finally, we show that for the latter result, we can replace $\Delta_{\max}$ by the temporal $h$-index of $\mathcal{G}$ which we define to be the largest number $h$ such that $\mathcal{G}$ contains $h$ temporal vertices with degree at least $h$. This parameter is upper-bounded by $\Delta_{\max}$ and potentially much smaller, when we have a skewed degree distribution for the temporal vertices. In our opinion, this parameter can be of interest for other temporal graph problems as well.

Proofs of statements marked with ($\star$) are deferred to the full version.

## 2 Preliminaries

Throughout the work, we denote the set of integers $\{i, i+1, \dots, j-1, j\}$ by $[i, j]$. If $i = 1$, we write $[j]$. Given a set $V$ and an integer $k > 0$, we write $\binom{V}{k}$ for the collection of all subsets of $V$ of size exactly $k$. For the main definitions of classical and parameterized complexity theory, including the definition of treewidth and tree decompositions, we refer to the standard monographs [3, 7].

**Graph notation.** A *(static) graph* $G = (V, E)$ consists of a *vertex set* $V$ and an *edge set* $E \subseteq \binom{V}{2}$. Given a graph $G$, we refer to the set of vertices by $V(G)$ and to the set of edges by $E(G)$. Throughout this work, if $G$ is clear from the context, we let $n := |V|$ and $m := |E|$. For an edge $\{u, v\}$ we may write $uv$. The *neighborhood* of a vertex $v$ is denoted by $N_G(v)$. Further, we call $N_G[v] := N_G(v) \cup \{v\}$ the *closed neighborhood* of $v$. The closed neighborhood $N_G[S]$ of a set $S$ of vertices is defined as the union of the closed neighborhoods over all vertices from the set. The open neighborhood $N_G(S)$ of a set $S$ of vertices is defined as $N_G[S] \setminus S$. If $G$ is clear from the context, we may omit the subscript.

We let $\deg_G(v) := |N_G(v)|$ denote the degree of a vertex $v$. If $\deg_G(v) = 0$, then we say that $v$ is *isolated*. The *maximum degree* of a graph $G$ is $\Delta(G) := \max_{v \in V(G)} \deg_G(v)$. We write $\Delta$ instead of $\Delta(G)$ when $G$ is clear from the context. The *h-index* of a graph $G$ is the largest integer $h$ such that $G$ contains at least $h$ vertices of degree at least $h$ [14]. We call the graph $H = (W, F)$ a *subgraph* of the graph $G = (V, E)$ if $W \subseteq V$ and $F \subseteq E$. The subgraph is called *induced* if $F = E \cap \binom{W}{2}$. The graph $G[W] := (W, E \cap \binom{W}{2})$ is the *subgraph of $G$ induced by $W$*.

A graph $G = (V, E)$ is a *clique* if $E(G) = \binom{V}{2}$. A *path* of length $p$ is a graph of the form $(V = \{v_1, \ldots, v_{p+1}\}, E = \bigcup_{i=1}^{p} v_i v_{i+1})$. We call $v_1$ and $v_{p+1}$ the *endpoints* of the path. A *cycle* of length $p \geq 3$ is a path of length $p - 1$ with an additional edge between the endpoints of the path.

**Temporal graph notation.** We use the following definition of temporal graphs.

▶ **Definition 2.1.** *A* temporal graph $\mathcal{G}$ *is a finite sequence of graphs* $(G_1, \ldots, G_T)$ *which all have the same set of vertices* $V(\mathcal{G}) := V(G_1) = \ldots = V(G_T)$.

The graphs of the sequence are called *snapshots*.We write $E_i$ instead of $E(G_i)$ and $V$ instead of $V(\mathcal{G})$, if the temporal graph $\mathcal{G}$ is clear from context. An edge of a snapshot $G_i$ is a *temporal edge* and the pair $(v, i) \in V \times [T]$ is the *temporal vertex* of $v$ in snapshot $G_i$. Sometimes we say the edge $e$ *appears* in snapshot $G_i$ if $e \in E_i$. Given a temporal graph $\mathcal{G} = (G_1, \ldots, G_T)$, we define the *underlying graph* of $\mathcal{G}$ by $G_\downarrow(\mathcal{G}) := (V_\downarrow := V(\mathcal{G}), E_\downarrow := \bigcup_{i=1}^{T} E_i)$ and simply write $G_\downarrow$ when $\mathcal{G}$ is clear from the context. We let $\Delta = \Delta(G_\downarrow)$. With $\Delta_i(\mathcal{G})$ we denote the maximum degree of the $i$th snapshot, that is, $\Delta_i(\mathcal{G}) := \Delta(G_i)$. Further, we define the maximum degree over all snapshots of a given temporal graph by $\Delta_{\max}(\mathcal{G}) := \max_{i=1}^{T} \Delta_i(\mathcal{G})$ and call $\Delta_{\max}(\mathcal{G})$ the *maximum snapshot degree* of $\mathcal{G}$. We refer to $\Delta_{\max}(\mathcal{G})$ by $\Delta_{\max}$ when $\mathcal{G}$ is clear from the context. We define the $h$-index of a temporal graph as follows.

▶ **Definition 2.2.** *The* $h$-index $h(\mathcal{G})$ of (the temporal graph) $\mathcal{G}$ *is the maximum number* $h(\mathcal{G})$ *such that* $\mathcal{G}$ *contains at least* $h(\mathcal{G})$ *temporal vertices of degree at least* $h(\mathcal{G})$.[1]

An alternative definition for $h(\mathcal{G})$ is the following: Consider the graph $H$ which is the disjoint union of all snapshots of $\mathcal{G}$. Then $h(\mathcal{G})$ is the $h$-index of $H$. If a snapshot $G_i$ has an empty edge set, that is, $E_i = \emptyset$, then we say $G_i$ is an *empty snapshot*. For a temporal vertex $(v, i)$, we define the *neighborhood* by $N_{\mathcal{G}}((v, i)) := N_{G_i}(v)$ and analogously the *closed neighborhood* by $N_{\mathcal{G}}[(v, i)] := N_{G_i}[v]$. The (closed) neighborhood of a set of temporal vertices is defined analogously to the (closed) neighborhood of static vertices. For a set of vertices $W \subseteq V(\mathcal{G})$, we let $\mathcal{G}[W] := (G_1[W], \ldots, G_T[W])$ denote the temporal subgraph of $\mathcal{G}$ *induced by $W$*.

We now give formal definitions for temporal vertex covers and dominating sets that we search for in TDS and TVC and their extensions. Let $\mathcal{G} = (G_1, \ldots, G_T)$ be a temporal graph and let $\mathcal{S} = V(\mathcal{G}) \times [T]$ be the set of temporal vertices of $\mathcal{G}$. We say $\mathcal{S}$ *covers* the edge $uv \in E_\downarrow$, if there exists a timestamp $i \in [T]$ such that $uv \in E_i$ and at least one of $(u, i)$ or $(v, i)$ is contained in $\mathcal{S}$. If $\mathcal{S}$ covers all edges from $E_\downarrow$, then we call $\mathcal{S}$ a *temporal vertex cover* (see Figure 1 for an example). We say a vertex $u \in V_\downarrow = V(\mathcal{G})$ is *dominated* by the temporal vertex $(v, i)$ if $u \in N_{G_i}[v]$. A set of temporal vertices $\mathcal{S} \subseteq V(\mathcal{G}) \times [T]$ is a *temporal dominating set* if each vertex of $V_\downarrow$ is dominated by at least one temporal vertex of $\mathcal{S}$ (see Figure 1 for an example).

[1] Note that our definition of $h$-index of $\mathcal{G}$ differs from the one given by Oettershagen et al. [31]

**Parameterized Complexity Theory.** Parameterized complexity is a two-dimensional framework for describing the computational complexity of decision problems [7]. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where $\Sigma$ is a fixed, finite alphabet. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, $k$ is called the *parameter* and the *size* of the instance is $|x| + k$.

A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$ (called *fixed-parameter algorithm*), a computable function $f : \mathbb{N} \to \mathbb{N}$, and a constant $c$ such that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^c$. The complexity class containing all fixed-parameter tractable problems is called FPT.

To show that a problem is unlikely to have an FPT-algorithm, one makes use of *parameterized reductions*. Let $A, B \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from $A$ to $B$ is an algorithm that, given an instance $(x, k)$ of $A$, outputs an instance $(x', k')$ of $B$ such that $(x, k) \in A$ if and only if $(x', k') \in B$, $k' \leq g(k)$ for some computable function $g$, and the running time is $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function $f$. In particular, if there exists a parameterized reduction from $A$ to $B$ and $A$ is not in FPT, then the parameterized reduction implies that $B$ is also not in FPT. The W-hierarchy consists of the complexity classes $\mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \ldots$; we say that a parameterized problem $A$ is $\mathrm{W}[i]$-hard if every problem from $\mathrm{W}[i]$ can be reduced to $A$ by some parameterized reduction. The standard assumption of parameterized complexity theory is $\mathrm{FPT} \subsetneq \mathrm{W}[1]$. This assumption allows to show that a parameterized problem is not fixed-parameter tractable by providing a parameterized reduction from some $\mathrm{W}[i]$-hard problem.

## 3 Classical Complexity of the Considered Problems

We now present our dichotomy for $\Delta_{\max}$ for all considered problems.

### 3.1 Hardness Results for Unbounded Budget

In this section we study restricted settings for which TDS and therefore all its generalizations are NP-hard. Akrida et al. [1] proved by a reduction from SET COVER that TVC is NP-hard even when the underlying graph $G_\downarrow$ is a star graph. Recall that this reduction [1, Thm. 2] also implies that TVC is W[2]-hard when parameterized by $k$ under these restrictions. With a similar idea, we can also show that TDS is NP-hard if the underlying graph is a star graph.

▶ **Theorem 3.1.** *TDS is NP-hard and* W[2]*-hard when parameterized by $k$ even when the underlying graph $G_\downarrow$ is a star graph.*

**Proof.** We present a polynomial-time reduction from HITTING SET, which is NP-hard and W[2]-hard when parameterized by the size $k$ of the sought solution.

> HITTING SET
> **Input:** A universe $U$, a family $F$ of subsets of $U$ (called *hyperedges*), and an integer $k$.
> **Question:** Is there a *hitting set* of size at most $k$ for $F$, that is, is there a set $S \subseteq U$ of size at most $k$, such that each hyperedge of $F$ contains at least one element of $S$?

Let $I := (U, F, k)$ be an instance of HITTING SET and assume without loss of generality that each hyperedge of $F$ is non-empty, as otherwise, $I$ is a trivial no-instance. Fix some arbitrary order on $U$ and consider the following temporal graph $\mathcal{G}$. The vertex set consists of one vertex $v_e$ for each hyperedge $e \in F$, and one center vertex $c$. Further, we introduce for each element $u_i \in U$ one snapshot $G_i$ in which the center is connected to exactly those vertices $v_e$ for which the hyperedge $e$ contains $u_i$. By construction, $G_\downarrow$ is a star graph. Clearly, this reduction can be done in polynomial time.

*Correctness:* We show that $F$ has a hitting set of size at most $k$ if and only if $\mathcal{G}$ has a temporal dominating set $\mathcal{S}$ of size at most $k$.

($\Rightarrow$) Let $S \subseteq U$ be a hitting set of size at most $k$ for $F$. For each $u_i \in S$, we add the temporal vertex $(c, i)$ to $\mathcal{S}$. For each hyperedge $e \in F$ there exists a $u_i \in S$ such that $u_i \in e$ since $S$ is a hitting set. By construction, the temporal vertex $(c, i)$ dominates the vertex $v_e$ because $cv_e \in E_i$. It follows immediately that the chosen temporal vertices are a temporal dominating set of $\mathcal{G}$.

($\Leftarrow$) Let $\mathcal{S}$ be a temporal dominating set of $\mathcal{G}$ of size at most $k$. Note that we can assume that $\mathcal{S}$ only contains temporal vertices of the center vertex $c$, that is, $\mathcal{S} \subseteq \{(c, i) : i \in [1, |U|]\}$. This is due to the fact that, for each temporal vertex $(v_e, i)$ with $v_e \neq c$ there exists a snapshot $G_j$ such that $cv_e \in E_j$ and consequently the temporal vertex $(v_e, i)$ dominates a subset of the vertices dominated by $(c, j)$. Consider the set $S$ obtained by adding $u_i$ to $S$ if $(c, i)$ is in the temporal dominating set $\mathcal{S}$. For each $e \in F$, the vertex $v_e$ is dominated by some vertex $(c, i) \in \mathcal{S}$. By construction, $u_i \in S$ and $u_i \in e$. Consequently, $S$ is a hitting set of size at most $k$ for $F$. ◀

If we reduce from the NP-hard special case [18] of Hitting Set where each hyperedge has size 2 and each element occurs in at most three hyperedges, the resulting temporal graph has at most three edges in each snapshot, and each edge appears in at most two snapshots, which implies the following.

▶ **Corollary 3.2.** *TDS is NP-hard even when the underlying graph $G_\downarrow$ is a star graph, there are at most three edges in each snapshot, and each edge appears at most two times.*

Due to the NP-hardness for underlying stars, parameterized algorithms for a large number of parameters of the underlying graph are impossible. The next results show that even if the underlying graph is a clique, the problem remains NP-hard even for very restricted number of edges in each snapshot. Note that this also bounds the maximum snapshot degree $\Delta_{\max}$.

▶ **Theorem 3.3.** *TDS is NP-hard even when the underlying graph $G_\downarrow$ is a clique and there are at most two edges in each snapshot.*

**Proof.** We provide a reduction from Exact Cover By 3-Sets (X3C) [18]. The input of the problem consists of a finite set $X = \{x_1, \ldots, x_{3q}\}$ together with a family $\mathcal{F} = \{F_1, \ldots, F_p\}$ of size-3 subsets of $X$ and the question is whether there is a subset $\mathcal{C}$ of $\mathcal{F}$ such that each $x_j$ appears in exactly one set from $\mathcal{C}$ (this implies $|\mathcal{C}| = q$).

Consider a temporal graph $\mathcal{G}$ with vertex set $V(\mathcal{G}) := \{v_j : x_j \in X\}$ and $p + \binom{3q}{2}$ snapshots. For $i \in [p]$ let $j(i) := \min\{s \in [3q] : x_s \in F_i\}$ and define the edge set $E_i := \{v_{j(i)}v_s : s \neq j(i), x_s \in F_i\}$. The purpose of the remaining $\binom{3q}{2}$ snapshots is to ensure that the underlying graph is a clique. This can be done by adding for each pair of elements $x_r, x_s \in X$ one snapshot which only contains the edge $v_r v_s$. By construction, the underlying graph is a clique and there are at most two edges in each snapshot. Clearly, this reduction can be done in polynomial time.

*Correctness:* We show that $(X, \mathcal{F})$ is a yes-instance of X3C if and only if $(\mathcal{G}, q)$ is a yes-instance of TDS.

($\Rightarrow$) Suppose each element from $X$ appears in exactly one set from $\mathcal{C} \subseteq \mathcal{F}$. Recall that this implies that $\mathcal{C}$ has size $q$. Consider the following temporal vertex set $\mathcal{S}$. For each $F_i \in \mathcal{C}$ we add the temporal vertex $(v_{j(i)}, i)$ to $\mathcal{S}$. Clearly, $(v_{j(i)}, i)$ dominates exactly the vertices from $V(\mathcal{G})$ which correspond to the elements from $F_i$. Since every element of $X$ is contained in a set from $\mathcal{C}$, it follows that every vertex from $V(\mathcal{G})$ is dominated.

($\Leftarrow$) Suppose that $\mathcal{S}$ is a temporal dominating set of size $q$ for $\mathcal{G}$. Since each snapshot contains at most two edges and $|V(\mathcal{G})| = 3q$, it is clear that each temporal vertex from $\mathcal{S}$ dominates exactly three vertices which are not dominated by any other temporal vertex from $\mathcal{S}$. By construction, it follows that $\mathcal{S} \subseteq \{(v_{j(i)}, i) : i \in [p]\}$, since only these temporal vertices can dominate more than two vertices in their respective snapshot. We obtain a solution $\mathcal{C}$ to the X3C instance by adding $F_i$ to $\mathcal{C}$ if and only if $\mathcal{S}$ contains a temporal vertex from the $i$th snapshot, that is, the temporal vertex $(v_{j(i)}, i)$. Observe that by definition of the first $p$ snapshots, the set $F_i \in \mathcal{C}$ covers exactly the elements from $X$ whose corresponding vertices are dominated by $(v_{j(i)}, i) \in \mathcal{S}$. This implies that every element of $X$ is contained in exactly one set from $\mathcal{C}$, since $\mathcal{S}$ dominates each vertex exactly once. ◀

We now show a similar hardness result for Temporal Vertex Cover with a slightly larger snapshot degree on instances with a clique as underlying graph. Recall that for instances with a star as underlying graph, TVC is known to be NP-hard [1, Thm. 2].

▶ **Theorem 3.4** (⋆). *TVC is NP-hard even when the underlying graph $G_\downarrow$ is a clique, there are at most three edges in each snapshot, and each edge appears at most twice.*

## 3.2 Polynomial-Time Solvable Cases for Unbounded Budget

We now discuss special cases of TDS that can be solved in polynomial time. If we restrict the underlying graph to paths or cycles, then TDS and TPDS are solvable in polynomial time. This follows directly from Theorem 4.6, which states that TPDS is in FPT with respect to tw $+\Delta$, and hence we do not describe specific algorithms for this case.

Considering the maximum snapshot degree, note that Theorem 3.3 shows that TDS is NP-hard when restricted to temporal graphs of maximum snapshot degree two. However, if we restrict ourselves further to temporal graphs of maximum snapshot degree one, then TDS and its partial variant can be solved efficiently by reducing the problem to a matching problem.

▶ **Theorem 3.5** (⋆). *TPDS can be solved in polynomial time if the maximum snapshot degree is one.*

For TPVC we provide a polynomial-time algorithm for a maximum snapshot degree of 2 instead of one as for TPDS.

▶ **Theorem 3.6.** *TPVC can be solved in polynomial time if the maximum snapshot degree is 2.*

**Proof.** Let $I = (\mathcal{G}, k, t)$ be an instance of TPVC. Without loss of generality we assume that $|E(G_\downarrow)| \geq t$ and that $2k \geq t$ since otherwise the instance is a trivial no-instance. Moreover, let $\mathcal{S}$ be a solution of $I$ and let $F \subseteq E(G_\downarrow)$ be the set of covered edges by $\mathcal{S}$. Since the maximum snapshot degree is two, each vertex in $\mathcal{S}$ can cover at most two edges of $G_\downarrow$. Consequently, each vertex in $\mathcal{S}$ covers either a single edge of $G_\downarrow$ or a pair of adjacent edges of $G_\downarrow$, that is, a $P_3$.

Now, we consider a mapping $\phi : F \to \mathcal{S}$ such that each edge in $F$ is mapped to one of its endpoints in $\mathcal{S}$. Let $\mathcal{S}_2 \subseteq \mathcal{S}$ be the set of temporal vertices such that exactly two edges of $F$ are mapped to each temporal vertex in $\mathcal{S}_2$. Next, we show that maximizing the number of covered edges is equivalent to maximizing the number of edge-disjoint $P_3$s. More precisely, we prove that $\mathcal{S}$ is a solution if and only if $|\mathcal{S}_2| \geq t - k$: Each temporal vertex in $\mathcal{S}^* := \mathcal{S} \setminus \mathcal{S}_2$ can cover at most one edge of $E(G_\downarrow)$ according to $\phi$. Moreover, let $F^* := \{f \in F : \phi(f) \notin \mathcal{S}_2\}$. Since each temporal vertex in $S^*$ covers at most one edge of $E(G_\downarrow)$ according to $\phi$, we have $|F^*| \leq |S^*|$. Hence, we obtain $|\mathcal{S}| = k = |\mathcal{S}_2| + |\mathcal{S}^*|$ and $t \leq |F| = 2|\mathcal{S}_2| + |F^*| \leq |\mathcal{S}_2| + |\mathcal{S}^*| = k + |\mathcal{S}_2|$ implying $|\mathcal{S}_2| \geq t - k$.

Consequently, we need to find a set $T$ of at least $t - k$ temporal vertices such that each temporal vertex of $T$ covers a $P_3$ of $G_\downarrow$ such that all these $P_3$s are edge-disjoint. Afterwards, we can greedily add $(2k - t)$ temporal vertices to $T$ which cover exactly one so-far uncovered edge of $G_\downarrow$ to obtain a solution for $(\mathcal{G}, k, t)$. This is always possible due to our assumptions that $|E(G_\downarrow)| \geq t$ and that $2k \geq t$.

It remains to maximize the number of these edge-disjoint $P_3$s. However, since some pairs of adjacent edges, that is, some non-induced $P_3$s, of $G_\downarrow$ might not appear together in one snapshot, we cannot simply find a maximum packing of edge-disjoint $P_3$s in $G_\downarrow$. Thus, essentially we are given the underlying graph $G_\downarrow$ and a list of possible $P_3$s, that is, pairs of adjacent edges appearing in a common snapshot, and the task is to find a maximum packing of such $P_3$s. This problem is known as Edge Disjoint List $P_3$-Packing and can be solved in polynomial time via a reduction to Maximum Matching [19]. Since [19] does not provide the construction, we describe it for sake of completeness. The auxillary graph $H$ contains a vertex $v_e$ for each edge $e \in E(G_\downarrow)$. Moreover, two vertices $v_{e_1}$ and $v_{e_2}$ of $H$ are adjacent if and only if $e_1 \cap e_2 \neq \emptyset$ and there exists a snapshot $G_i$ where both edges $e_1$ and $e_2$ are active. Now, maximizing the number $z$ of edge-disjoint $P_3$ covered by the solution is equivalent to finding a maximum matching in $H$ which can be done in polynomial time [28]. ◀

Theorem 3.6 implies that TPVC can be solved in polynomial time if $G_\downarrow$ is a path. Hence, our result generalizes a result of Hamm et al. [21, Thm. 5] who provided a polynomial-time algorithm for TVC if $G_\downarrow$ is a path.

## 3.3 The Influence of Budgets on the Complexity

We now study Budget-TDS and Budget-TVC. Recall that in those problems the input has an additional integer $b$ and we are allowed to select $k$ temporal vertices overall but only up to $b$ temporal vertices per snapshot.

We proceed by showing that Budget-TDS and Budget-TVC are already NP-hard even if the underlying graph is a path.

▶ **Theorem 3.7.** *Budget-TDS with a budget constraint of $b = 1$ is NP-hard, even if the underlying graph is a path and the maximum snapshot degree is one.*

**Proof.** We reduce from the NP-hard Rainbow Matching problem on properly edge-colored paths [27]. In this problem, the instance $(P, \phi, q)$ consists of a path $P$ with vertex set $\{v_1, \ldots, v_n\}$, an edge coloring $\phi\colon E(P) \to [c]$ for some $c \in \mathbb{N}$, and an integer $q$ and the question is whether there exists a matching of size at least $q$ such that all edges in the matching are assigned a different color.

Consider the temporal graph $\mathcal{G}$ with vertex set $V(\mathcal{G}) := V(P)$ and the following snapshots: For each color $i \in [c]$, we introduce one snapshot $G_i$, which contains exactly the edges colored with the corresponding color $i$. Furthermore, we add $n - 2q$ additional empty snapshots and set $k := n - q$ to obtain the Budget-TDS instance $(\mathcal{G}, k, b = 1)$. By construction, the underlying graph is a path and the maximum snapshot degree is one since $\phi$ is a proper edge coloring, that is, no two incident edges of $P$ have the same color.

*Correctness:* We show that $P$ has a rainbow matching of size at least $q$ if and only if $\mathcal{G}$ has a temporal dominating set $\mathcal{S}$ of size at most $k = n - q$ which does not exceed the budget of one in any of the snapshots.

(⇒) Suppose $M$ is a rainbow matching of size $q$. For each color $i \in [c]$ and each edge $v_j v_{j+1} \in M$ with $\phi(v_j v_{j+1}) = i$ we add $(v_j, i)$ to $\mathcal{S}$. Note that these temporal vertices already dominate $2q$ vertices since $M$ is a matching. For the remaining $n - 2q$ undominated vertices from $V(\mathcal{G})$ we add isolated temporal vertices from the empty snapshots to $\mathcal{S}$ such that there is no conflict with the budget constraint. This can be done because $\mathcal{G}$ contains $n - 2q$ empty snapshots. Finally, note that $\mathcal{S}$ dominates all vertices, has size $q + n - 2q = n - q = k$ and contains at most one temporal vertex from each snapshot since $M$ is a rainbow matching.

(⇐) Suppose that $\mathcal{S}$ is a solution to $(\mathcal{G}, k, b = 1)$. Since the maximum snapshot degree is one, we know that every temporal vertex dominates at most two vertices. A solution of size $k = n - q$ needs to dominate all $n$ vertices, which implies that there exist $q$ temporal vertices in $\mathcal{S}$ which dominate in total $2q$ vertices. Each of these $q$ temporal vertices is incident to exactly one edge and therefore comes from a snapshot which corresponds to some color of the edge coloring $\phi$. By adding all these edges to $M$, we obtain a rainbow matching of size at least $q$. As the $q$ temporal vertices dominate in total $2q$ vertices, it follows that $M$ is a matching. The rainbow property is a consequence of the budget restriction which implies that $\mathcal{S}$ contains no two vertices from the same snapshot and therefore no two edges of the same color are added to $M$. ◀

We get almost the same hardness for Budget-TVC.

▶ **Theorem 3.8** (⋆). *Budget-TVC with a budget constraint of $b = 1$ is NP-hard, even if the underlying graph is a path.*

Hence Budget-TDS and Budget-TVC are strictly harder than TDS and TVC with respect to the structure of the underlying graph and with respect to the maximum snapshot degree. Moreover, since $G_\downarrow$ is a path in Theorem 3.8, we directly obtain that Budget-TVC remains NP-hard even if the maximum snapshot degree is two. This is in sharp contrast to TVC which is solvable in polynomial time if the maximum snapshot degree is two, see Theorem 3.6. We now show that the snapshot degree of 2 in Theorem 3.8 is necessary for the NP-hardness on paths.

▶ **Theorem 3.9.** *Budget-TPVC can be solved in polynomial time if the maximum snapshot degree is one.*

**Proof.** Let $(\mathcal{G}, k, t, b)$ be a Budget-TPVC instance. Observe that since the maximum snapshot degree is one, each temporal vertex can cover at most one edge. Consequently, if $k < t$, we have a no-instance. Moreover, since an optimal solution never contains both endpoints of an edge, it is safe to assume that $k = t$.

To solve the instance $(\mathcal{G}, k, k, b)$, we use a max-flow problem, which can be solved in polynomial time [20]. Let $H$ be a flow-network with source $s$ and target $z$. Moreover, $H$ has a vertex $s_i$ for each snapshot $G_i$ and a vertex $e$ for each edge $e \in E(G_\downarrow)$. The source has an arc with capacity $b$ to each vertex $s_i$ corresponding to a snapshot. The vertex $s_i$ corresponding to snapshot $G_i$ has an arc with capacity one to edge $e \in E(G_\downarrow)$ if $e$ can be covered in $G_i$. Finally, each vertex $e$ for each $e \in E(G_\downarrow)$ has an arc to the target $z$ with capacity 1.

We now show that $(\mathcal{G}, k, k, b)$ is a yes-instance of Budget-TPVC if and only if $H$ has a maximum flow of $k$.

(⇒) Assume that $H$ has a maximum flow of value $k$. Since each vertex $e \in V(H)$ corresponding to an edge in $G_\downarrow$ has only one outgoing arc which has capacity one, there exists at most one vertex $s_i$ such that there is a flow of value 1 from $s_i$ to $e$. Now, since the maximum snapshot degree is 1, we let $(v, i)$ be a temporal vertex such that $v$ is one of the endpoints of the edge in $G_\downarrow$ corresponding to $e$. We now argue that the set $S$ of these temporal vertices is a solution for $(\mathcal{G}, k, k, b)$. By the above, each vertex in $S$ covers exactly one unique edge and by the construction of our flow-network $S$ contains at most $b$ temporal vertices from each snapshot. Furthermore, since the flow has value $k$, we obtain that $|S| = k$ and thus $S$ is a solution for $(\mathcal{G}, k, k, b)$.

($\Leftarrow$) Let $S$ be a solution for $(\mathcal{G}, k, k, b)$ where no edge is covered twice. Consequently, each vertex in $S$ covers exactly one edge and this edge is covered by no other vertex in $S$. More precisely, let $S_i \subseteq S$ be the temporal vertices contained in snapshot $i$. Note that $|S_i| \leq b$. For each snapshot $i$, we let a flow with value $|S_i|$ flow from the source $s$ to vertex $s_i$. Since each vertex $v$ in $S$ covers a unique edge $e \in E(G_\downarrow)$, we let a flow of value 1 flow from $S_i$ to each edge which is covered by a vertex in $S_i$. Clearly, these are $|S_i|$ many. Finally, we let a flow from each covered edge to the target $z$ with a value 1 flow. Consequently, $H$ has a flow with value $|S| = k$. ◀

## 4 Parameterized Complexity

In this section, we study all problems from the viewpoint of parameterized complexity. In particular, we first provide FPT-algorithms for the number $t$ of dominated vertices or covered edges, respectively, in the partial variants. Then, we extend these results to obtain FPT-algorithms for $k$ plus the $h$-index of the temporal vertex degrees $h(\mathcal{G})$. Finally, we provide FPT-algorithms for the purely structural parameter $\text{tw} + \Delta$ of the underlying graph. We state the algorithms in their most general form, that is, if possible for the partial and budget variant.

### 4.1 Parameters $t$ and $k + h(\mathcal{G})$

Recall that in the partial variants TPDS and TPVC the input contains an additional integer $t$ for the number of dominated vertices/covered edges. It is known that Partial Dominating Set and Partial Vertex Cover are in FPT with respect to $t$ (for example a consequence of [4]). We show that these result extend to the temporal and budget versions.

The idea is to use color coding [2] which is a very popular technique to obtain fixed-parameter algorithms [7]. Given some instance of a parameterized problem, the rough idea is to randomly color some elements of this instance and then solve a colorful version of the given instance. If the colorful version of the problem is fixed-parameter tractable, then we can often derandomize the algorithm such that it solves our original instance by deterministically considering a specific family of colorings. We consider the derandomized version of color coding that makes use of perfect hash families. More formally, a family $\mathcal{F}$ of functions from $[n]$ to $[k]$ is $(n, k)$-*perfect* if for each $S \subseteq [n]$ of size $k$ there exists a function $f \in \mathcal{F}$ such that the restriction of $\mathcal{F}$ to $S$ is injective. The following theorem shows that a perfect hash family can be constructed in FPT time.

▶ **Proposition 4.1** ([30]). *For any $n, k \geq 1$ one can construct an $(n, k)$-perfect hash family of size $e^k k^{\mathcal{O}(\log k)} \log n$ in time $e^k k^{\mathcal{O}(\log k)} n \log n$.*

Initially, we show our result for Budget-TPDS.

▶ **Theorem 4.2.** *Budget-TPDS can be solved in $2^{\mathcal{O}(t)} \cdot |I|^{\mathcal{O}(1)}$ time.*

**Proof.** Let $(\mathcal{G}, k, t, b)$ be a Budget-TPDS-instance. We can assume $b = 1$ by taking $b$ copies of each snapshot. Note that this increases the lifetime of the temporal graph by the factor $b$.

Consider the following randomized algorithm. Let $\phi\colon V(\mathcal{G}) \to [t]$ be a uniformly at random coloring of the vertices $V(\mathcal{G})$. We say that a temporal vertex $(v, i)$ *dominates the color* $q$ if there exists a neighbor of $v$ in $G_i$ with color $q$. With $C_{v,i}$ we denote the set of colors, which are dominated by the temporal vertex $(v, i)$. For each set of colors $X \subseteq [t]$ and each $i \in [0, T]$ let

$\mathrm{DP}[X, i] \mathrel{\hat{=}}$ Minimum size of any temporal vertex set $\mathcal{S} \subseteq V(\mathcal{G}) \times [1, i]$ such that at most one temporal vertex of each snapshot is used and the temporal vertices of $\mathcal{S}$ dominate the colors from $X$.

We initialize the table by setting $\mathrm{DP}[\emptyset, 0] = 0$ and $\mathrm{DP}[X, 0] = \infty$ for all $\emptyset \neq X \subseteq [t]$. The remaining entries are computed recursively by

$$\mathrm{DP}[X, i] = \min\big(\{\mathrm{DP}[X \setminus C_{v,i}, i-1] + 1 : v \in V, C_{v,i} \cap X \neq \emptyset\}, \{\mathrm{DP}[X, i-1]\}\big).$$

We prove the correctness by showing two inequalities.

($\geq$) Suppose that $\mathcal{S}$ is a set of temporal vertices for which the minimum in the definition of $\mathrm{DP}[X, i]$ is attained. Due to the minimizing, it is clear that each temporal vertex from $\mathcal{S}$ dominates at least one color from $X$. If $\mathcal{S}$ contains a temporal vertex $(v, i)$ from $G_i$, then $\mathcal{S} \setminus \{(v, i)\}$ is considered in the definition of the entry $\mathrm{DP}[X \setminus C_{v,i}, i-1]$. Otherwise $\mathcal{S}$ is considered in the definition of $\mathrm{DP}[X, i-1]$. Therefore the inequality holds.

($\leq$) Note that each set $\mathcal{S}$ which is considered in the definition of $\mathrm{DP}[X, i-1]$ is also considered in the definition of $\mathrm{DP}[X, i]$. Hence, we have $\mathrm{DP}[X, i] \leq \mathrm{DP}[X, i-1]$. Now suppose the minimum on the right hand side of the computation is attained for the entry $\mathrm{DP}[X \setminus C_{v,i}, i-1]$ and the set $\mathcal{S}$. Then clearly $\mathcal{S} \cup \{(v, i)\}$ is considered in the definition of $\mathrm{DP}[X, i]$ and we have $\mathrm{DP}[X, i] \leq \mathrm{DP}[X \setminus C_{v,i}, i-1] + 1$. This proves the inequality.

Finally, the algorithm returns yes if and only if $\mathrm{DP}[\{1, \dots, t\}, T] \leq k$. This means, it returns yes if and only if there is a temporal vertex set of size at most $k$ which dominates $t$ pairwise differently colored vertices.

If we run this algorithm once, then it is not guaranteed that the random coloring $\phi$ colors $t$ vertices, which are dominated by some solution of the Budget-TPDS-instance, with $t$ different colors. For derandomization we can iterate over a specific set of colorings, instead of randomly coloring the vertices. The set of colorings needs to contain at least one coloring which colors the "correct" $t$ vertices with $t$ distinct colors. Such a set can be constructed by using the definition of perfect hash families (see Proposition 4.1).

For a fixed coloring $\phi$, the size of the table is $\mathcal{O}(2^t \cdot T)$ and it takes $\mathcal{O}(n^2 T)$ time to compute one entry, which implies an overall running time of $\mathcal{O}(2^t \cdot n^2 T^2)$ if the budget equals one. If we take $b$ copies of each snapshot, such that we can assume $b = 1$, then the lifetime increases by the factor $b$ and consequently the running time is $\mathcal{O}(2^t \cdot n^2 T^2 b^2)$ for a fixed coloring $\phi$.

Using the construction mentioned in Proposition 4.1 and running the algorithm from above for each coloring of this construction, we obtain an overall running time of $(2e)^t t^{\mathcal{O}(\log t)} \cdot n^3 (\log n) b^2 T^2$. ◀

The algorithm of Theorem 4.2 also works analogously for Budget-TPVC: We only need to color the edges instead of the vertices. Hence, we obtain the following.

▶ **Corollary 4.3.** *Budget-TPVC can be solved in* $2^{\mathcal{O}(t)} \cdot |I|^{\mathcal{O}(1)}$ *time.*

Recall that $\Delta_{\max}(\mathcal{G}) \leq \Delta(G_\downarrow)$ is the maximum snapshot degree. Observe that if $t > k \cdot (\Delta_{\max}(\mathcal{G}) + 1)$ for the TDS variants and if $t > k \cdot \Delta_{\max}(\mathcal{G})$ for the TVC variants, then we have a no-instance. Thus, we obtain the following.

▶ **Proposition 4.4.** *Budget-TPDS and Budget-TPVC can be solved in* $2^{\mathcal{O}(k \cdot \Delta_{\max}(\mathcal{G}))} \cdot |I|^{\mathcal{O}(1)}$ *time.*

Furthermore, recall that TDS is W[2]-hard with respect to the solution size $k$ and that it is NP-hard even if $\Delta(\mathcal{G}) = 3$ [38]. Consequently, both parameters $k$ and $\Delta(\mathcal{G})$ are necessary to obtain an FPT-algorithm.

Of course, we would like to replace $\Delta_{\max}(\mathcal{G}) \leq \Delta(G_{\downarrow})$ by potentially smaller parameters. However, since TVC and TDS are W[2]-hard with respect to $k$ if $G_{\downarrow}$ is a star, we cannot make use of degree-based parameterizations of $G_{\downarrow}$ such as the $h$-index of $G_{\downarrow}$ or the $c$-closure [16] of $G_{\downarrow}$. To circumvent this problem we consider the $h$-index of the temporal graph $\mathcal{G}$, which we defined as the maximum number $h(\mathcal{G})$ such that $\mathcal{G}$ contains at least $h(\mathcal{G})$ temporal vertices of degree at least $h(\mathcal{G})$.

The idea for the algorithm is to first branch on the temporal vertices that have a high degree and then use the fact that after removing these vertices, the remaining instance has bounded maximum snapshot degree which allows us to use an FPT-algorithm for $t \leq k \cdot \Delta_{\max}$.

▶ **Theorem 4.5.** *Budget-TPDS and Budget-TPVC can be solved in* $2^{\mathcal{O}(k \cdot h(\mathcal{G}))} \cdot |I|^{\mathcal{O}(1)}$ *time.*

**Proof.** We only state the proof for Budget-TPDS, the proof for Budget-TPVC follows analogously. Let $I = (\mathcal{G}, k, t, b)$ be an instance of Budget-TPDS.

Let $Z$ be the set of temporal vertices of degree at least $h(\mathcal{G})+1$. By definition, $|Z| \leq h(\mathcal{G})$. The idea is to first branch on the high-degree temporal vertices $Z$ whether they are part of a fixed solution $\mathcal{S}$ or not and then to use an adaption of the color coding algorithm described in Theorem 4.2 for an auxiliary Budget-TPDS instance $I'$.

Now, let $\mathcal{S}$ be a fixed solution of $I$. Initially, we branch for each temporal vertex $(v, i) \in Z$ whether $(v, i)$ is part of $\mathcal{S}$ or not. Note that these are $2^{|Z|} \leq 2^{h(\mathcal{G})}$ possibilities. Next, consider a fixed possibility and let $Z_{\text{in}} := Z \cap \mathcal{S}$ and $Z_{\text{out}} := Z \setminus \mathcal{S}$. Now, let $t' := t - |V(N[Z_{\text{in}}])|$. Since each temporal vertex which is not in $Z$ can dominate at most $h(\mathcal{G}) + 1$ temporal vertices, if $t' > k \cdot (h(\mathcal{G}) + 1)$ then this choice of $Z$ cannot lead to a solution. If for no choice of $Z$ we obtain $t' \leq k \cdot (h(\mathcal{G}) + 1)$ then $I$ is a no-instance. Thus, in the following we assume that we consider a choice of $Z$ such that $t' \leq k \cdot (h(\mathcal{G}) + 1)$. Moreover, by $a_i$ we denote the number of temporal vertices of $Z_{\text{in}}$ of snapshot $i$. If $a_i > b$ for any $i \in [T]$, then we abort this choice of $Z_{\text{in}}$ and $Z_{\text{out}}$ since we can pick at most $b$ temporal vertices of each snapshot. Thus, in the following we can safely assume that $a_i \leq b$ for each $i \in [T]$. Now, let $b_i := b - a_i$. By our assumption on $a_i$, we have $b_i \geq 0$ for each $i \in [T]$.

Next, we solve an auxiliary Budget-TPDS instance $I' = (\mathcal{H}, k', t', b')$. Here, the temporal graph $\mathcal{H}$ has $b_i$ copies of snapshot $G_i$ for each $i \in [T]$. These are $T' = \sum_{i \in [T]} b_i$ snapshots in total. Note that $h(\mathcal{H})$ might be much higher than $h(\mathcal{G})$. Moreover, we set $b' := 1$ and $k' := k - |Z_{\text{in}}|$.

Next, we use color coding with $t' + 1$ colors on $V(\mathcal{H}) = V(\mathcal{G})$ to determine the set $\mathcal{S}' := \mathcal{S} \setminus Z_{\text{in}}$ as follows: The color $t' + 1$ is only used for the vertices in $V(N[Z_{\text{in}}])$ and consequently, in each uniformly at random coloring with the first $t'$ colors we only color the vertices $V(\mathcal{H}) \setminus V(N[Z_{\text{in}}])$. Again, we say that a temporal vertex $(v, i)$ *dominates the color* $q \in [t']$ if there exists a neighbor of $v$ in snapshot $H_i$ with color $q$. Now, the table $\text{DP}[X, i]$ for a set of colors $X \subseteq [t']$ and each $i \in [0, T]$ is defined similar to Theorem 4.2, that is:

$$\begin{aligned}\mathrm{DP}[X,i] \mathrel{\hat{=}} {} & \text{Minimum size of any temporal vertex set } \mathcal{S}' \text{ such that} \\ & \mathcal{S}' \subseteq (V(\mathcal{H}) \times [1,i]) \setminus Z_{\mathrm{out}} \text{ and such that} \\ & \text{at most one temporal vertex of each snapshot is used and} \\ & \text{the temporal vertices of } \mathcal{S}' \text{ dominate the colors from } X.\end{aligned}$$

We initialize the table analogously to the proof of Theorem 4.2, by setting $\mathrm{DP}[\emptyset, 0] = 0$ and $\mathrm{DP}[X, 0] = \infty$ for all $\emptyset \neq X \subseteq [t]$.

We again denote by $C_{v,i}$ the set of colors, which are dominated by the temporal vertex $(v, i)$. The remaining entries are then computed recursively by

$$\mathrm{DP}[X,i] = \min\left(\{\mathrm{DP}[X \setminus C_{v,i}, i-1]+1 : v \in V, C_{v,i} \cap X \neq \emptyset, (v,i) \notin Z_{\mathrm{out}}\}, \{\mathrm{DP}[X, i-1]\}\right).$$

Hence, the only change to the update in Theorem 4.2 is that we no do not consider temporal vertices of $Z_{\mathrm{out}}$. The correctness of the dynamic program follows since we assign all vertices in $V(N[Z_{\mathrm{in}}])$ color $t' + 1$ to indicate that they are already dominated. Since in the DP we only consider color subsets $X \subseteq [t']$, we cannot pick any vertex of $Z_{\mathrm{in}}$ again in the dynamic program. Moreover, during the updates of the dynamic program we do not pick any vertex of $Z_{\mathrm{out}}$. Consequently, $\mathcal{S} := \widehat{\mathcal{S}} \cup Z_{\mathrm{in}}$ is a solution for $I$, where $\widehat{\mathcal{S}} = \{(v, i) : (v, i') \in \mathcal{S}',$ snapshot $i'$ of $\mathcal{H}$ is a copy of snapshot $i$ in $\mathcal{G}\}$.

For the running time, observe that there are $2^{h(\mathcal{G})}$ possibilities for $Z_{\mathrm{in}}$ and $Z_{\mathrm{out}}$. Moreover, the running time of the dynamic program is $(2e)^{t'} t'^{\mathcal{O}(\log t')} \cdot n^3 (\log n) b^2 T^2$. Since we only use the dynamic program if $t' \leq k \cdot (h(\mathcal{G}) + 1)$, we obtain an overall running time of $2^{h(\mathcal{G})} (2e)^{k \cdot (h(\mathcal{G})+1)} k \cdot (h(\mathcal{G}) + 1)^{\mathcal{O}(\log(k \cdot (h(\mathcal{G})+1)))} \cdot n^3 (\log n) b^2 T^2$. ◀

## 4.2 Parameter tw +Δ

We finish this section by showing that TPDS and TPVC are in FPT for the purely structural parameter tw +Δ of the underlying graph. Also note that we cannot obtain FPT-algorithms for tw +Δ for Budget-TDS and Budget-TVC since they are already NP-hard on underlying graphs which are paths (see Theorems 3.7 and 3.8). Together with the hardness results from Section 3.1, this gives a comprehensive picture of the parameterized complexity for many structural parameters of the underlying graph.

▶ **Theorem 4.6** (⋆). *TPDS can be solved in* $f(\mathrm{tw} + \Delta) \cdot |I|^{\mathcal{O}(1)}$ *time.*

Again, we would like to obtain an FPT-algorithm for the combination of treewidth with a smaller parameter than Δ. However, this is unlikely due to the NP-hardness of stars where the treewidth, $h$-index [14], and $c$-closure [16] are constant.

Finally, we want to remark that a very similar approach also works to obtain fixed-parameter tractability for TVC and TPVC with respect to tw +Δ. This then generalizes a result from Hamm et al. [21] who showed that TVC can be solved in polynomial time for underlying paths and cycles.

▶ **Proposition 4.7** (⋆). *TPVC can be solved in* $f(\mathrm{tw} + \Delta) \cdot |I|^{\mathcal{O}(1)}$ *time.*

# 5 Future Work

Let us conclude with a list of open questions: First, it is open whether Temporal Vertex Cover and Temporal Dominating Set admit an FPT-algorithm for any structural parameterization for the underlying graph that is not already bounded in a function of tw $+\Delta$. Second, our FPT-algorithms for $t$ (Theorem 4.2) imply that all considered problems admit an FPT-algorithm for $n$, the number of vertices of the underlying graph. For Temporal Dominating Set the running time is $2^{\mathcal{O}(n)} \cdot |I|^{\mathcal{O}(1)}$ since $t \leq n$. For Temporal Vertex Cover, however, we only obtain a running time of $2^{\mathcal{O}(n^2)} \cdot |I|^{\mathcal{O}(1)}$ since $t \leq n^2$. Thus, it is open whether Temporal Vertex Cover can also be solved in $2^{\mathcal{O}(n)} \cdot |I|^{\mathcal{O}(1)}$ time. Finally, another direction for future work is the investigation of further temporal graph parameters: for example do our considered problems admit FPT-algorithms with respect to the temporal neighborhood diversity [13]?

## References


1. Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020. doi:10.1016/j.jcss.2019.08.002.
2. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
3. Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.
4. Markus Bläser. Computing small partial coverings. *Information Processing Letters*, 85(6):327–331, 2003. doi:10.1016/S0020-0190(02)00434-9.
5. Arnaud Casteigts. A journey through dynamic networks (with excursions). Habilitation thesis, Université de Bordeaux, 2018. URL: https://tel.archives-ouvertes.fr/tel-01883384.
6. Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010. doi:10.1016/j.tcs.2010.06.026.
7. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
8. Riccardo Dondi. Insights into the complexity of disentangling temporal graphs. In *Proceedings of the 23rd Italian Conference on Theoretical Computer Science (ICTCS '22)*, volume 3284 of *CEUR Workshop Proceedings*, pages 1–13. CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3284/2973.pdf.
9. Riccardo Dondi. Untangling temporal graphs of bounded degree. *Theoretical Computer Science*, 969:114040, 2023. doi:10.1016/j.tcs.2023.114040.
10. Riccardo Dondi and Manuel Lafond. An FPT algorithm for temporal graph untangling. In *Proceedings of the 18th International Symposium on Parameterized and Exact Computation (IPEC '23)*, volume 285 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.IPEC.2023.12.
11. Riccardo Dondi, Fabrizio Montecchiani, Giacomo Ortali, Tommaso Piselli, and Alessandra Tappini. Partial temporal vertex cover with bounded activity intervals. In *Proceedings of the 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND '24)*, volume 292 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SAND.2024.11.
12. Riccardo Dondi and Alexandru Popa. Timeline cover in temporal graphs: Exact and approximation algorithms. In *Proceedings of the 34th International Workshop on Combinatorial Algorithms (IWOCA '23)*, volume 13889 of *Lecture Notes in Computer Science*, pages 173–184. Springer, 2023. doi:10.1007/978-3-031-34347-6_15.
13. Jessica A. Enright, Samuel D. Hand, Laura Larios-Jones, and Kitty Meeks. Structural parameters for dense temporal graphs. In *Proceedings of the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS '24)*, volume 306 of *LIPIcs*, pages 52:1–52:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.MFCS.2024.52.

14 David Eppstein and Emma Spiro. The $h$-index of a graph and its application to dynamic subgraph statistics. *Journal of Graph Algorithms and Applications*, 16, May 2009.

15 Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. Multistage vertex cover. In *Proceedings of the 14th International Symposium on Parameterized and Exact Computation (IPEC '19)*, volume 148 of *LIPIcs*, pages 14:1–14:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.IPEC.2019.14.

16 Jacob Fox, Tim Roughgarden, C. Seshadhri, Fan Wei, and Nicole Wein. Finding cliques in social networks: A new distribution-free model. *SIAM Journal on Computing*, 49(2):448–464, 2020. doi:10.1137/18M1210459.

17 Vincent Froese, Pascal Kunz, and Philipp Zschoche. Disentangling the computational complexity of network untangling. *Theory of Computing Systems*, 68(1):103–121, 2024. doi:10.1007/s00224-023-10150-y.

18 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing (STOC '74)*, pages 47–63. ACM, 1974. doi:10.1145/800119.803884.

19 Tatsuya Gima, Tesshu Hanaka, Yasuaki Kobayashi, Yota Otachi, Tomohito Shirai, Akira Suzuki, Yuma Tamura, and Xiao Zhou. On the complexity of list $H$-packing for sparse graph classes. In *Proceedings of the 18th International Conference and Workshops on Algorithms and Computation (WALCOM '24)*, volume 14549 of *Lecture Notes in Computer Science*, pages 421–435. Springer, 2024. doi:10.1007/978-981-97-0566-5_30.

20 Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):783–797, 1998. doi:10.1145/290179.290181.

21 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI '22)*, pages 10193–10201. AAAI Press, 2022. doi:10.1609/aaai.v36i9.21259.

22 David G. Harris and N. S. Narayanaswamy. A faster algorithm for vertex cover parameterized by solution size. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS '24)*, volume 289 of *LIPIcs*, pages 40:1–40:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.STACS.2024.40.

23 Anton Herrmann. On the complexity of computing optimal dominating sets in temporal graphs. Master's thesis, Friedrich-Schiller-Universität Jena, 2024.

24 Petter Holme and Jari Saramäki. Temporal networks. *CoRR*, abs/1108.1780, 2011. arXiv:1108.1780.

25 Leon Kellerhals, Tomohiro Koana, and Pascal Kunz. Vertex cover and feedback vertex set above and below structural guarantees. In *Proceedings of the 17th International Symposium on Parameterized and Exact Computation (IPEC '22)*, volume 249 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.IPEC.2022.19.

26 David C. Kutner and Laura Larios-Jones. Temporal reachability dominating sets: Contagion in temporal graphs. In *Proceedings of the 19th International Symposium on Algorithmics of Wireless Networks (ALGOWIN '23)*, volume 14061 of *Lecture Notes in Computer Science*, pages 101–116. Springer, 2023. doi:10.1007/978-3-031-48882-5_8.

27 Van Bang Le and Florian Pfender. Complexity results for rainbow matchings. *Theoretical Computer Science*, 524:27–33, 2014. doi:10.1016/j.tcs.2013.12.013.

28 Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science (FOCS '80)*, pages 17–27. IEEE Computer Society, 1980. doi:10.1109/SFCS.1980.12.

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

30 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.

31 Lutz Oettershagen, Nils M. Kriege, and Petra Mutzel. A higher-order temporal h-index for evolving networks. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, pages 1770–1782. ACM, 2023. doi:10.1145/3580305.3599242.

32 Igor Razgon and Barry O'Sullivan. Almost 2-sat is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009. doi:10.1016/j.jcss.2009.04.002.

33 Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. The network-untangling problem: from interactions to activity timelines. *Data Mining and Knowledge Discovery*, 35(1):213–247, 2021. doi:10.1007/s10618-020-00717-5.

34 Carsten Schubert. Leveraging graph structure to untangle temporal networks efficiently. Master's thesis, TU Berlin, Institute of Software Engineering and Theoretical Computer Science, 2023.

35 M Verheije. Algorithms for domination problems on temporal graphs. Master's thesis, Utrecht University, Graduate School of Natural Sciences, 2021.

36 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. URL: http://www.cambridge.org/de/knowledge/isbn/item5759340/?site_locale=de_DE.

37 Dawei Zhao, Gaoxi Xiao, Zhen Wang, Lianhai Wang, and Lijuan Xu. Minimum dominating set of multiplex networks: Definition, application, and identification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(12):7823–7837, 2021. doi:10.1109/TSMC.2020.2987163.

38 Igor E. Zverovich and Vadim E. Zverovich. An induced subgraph characterization of domination perfect graphs. *Journal of Graph Theory*, 20(3):375–395, 1995. doi:10.1002/jgt.3190200313.