Multistage Vertex Cover

Till Fluschnik
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany
till.fluschnik@tu-berlin.de

Rolf Niedermeier
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany
rolf.niedermeier@tu-berlin.de

Valentin Rohm
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany
valentinl.rohm@campus.tu-berlin.de

Philipp Zschoche
Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Germany
zschoche@tu-berlin.de

Abstract

Covering all edges of a graph by a small number of vertices, this is the NP-hard Vertex Cover problem, is among the most fundamental algorithmic tasks. Following a recent trend in studying dynamic and temporal graphs, we initiate the study of Multistage Vertex Cover. Herein, having a series of graphs with same vertex set but over time changing edge sets (known as temporal graph consisting of time layers), the goal is to find for each layer of the temporal graph a small vertex cover *and* to guarantee that the two vertex cover sets between two subsequent layers differ not too much (specified by a given parameter). We show that, different from classic Vertex Cover and some other dynamic or temporal variants of it, Multistage Vertex Cover is computationally hard even in fairly restricted settings. On the positive side, however, we also spot several fixed-parameter tractability results based on some of the most natural parameterizations.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases NP-hardness, dynamic graph problems, temporal graphs, time-evolving networks, W[1]-hardness, fixed-parameter tractability, kernelization

Digital Object Identifier 10.4230/LIPIcs.IPEC.2019.14

Related Version A full version of the paper: `https://arxiv.org/abs/1906.00659`.

Funding *Till Fluschnik*: Supported by the DFG, project TORE (NI 369/18).

1 Introduction

Vertex Cover (VC) asks, given an undirected graph G and an integer $k \geq 0$, whether at most k vertices can be deleted from G such that the remaining graph contains no edge. VC is NP-hard and it is a formative problem of algorithmics and combinatorial optimization. We study a *time-dependent*, "*multistage*" version, namely a variant of VC on temporal graphs. A *temporal graph* $\mathcal{G}$ is a tuple $(V, \mathcal{E}, \tau)$ consisting of a set V of vertices, a discrete time-horizon τ, and a set of temporal edges $\mathcal{E} \subseteq \binom{V}{2} \times \{1, \ldots, \tau\}$. Equivalently, a temporal graph $\mathcal{G}$ can be seen as a vector $(G_1, \ldots, G_\tau)$ of static graphs (*layers*), where each graph is defined over the same vertex set V. Then, our specific goal is to find a small vertex cover S_i for each layer G_i such that the sizes of the symmetric differences $S_i \triangle S_{i+1}$ between the vertex covers S_i and S_{i+1} of every two consecutive layers G_i and G_{i+1} are small. Formally, we thus introduce and study the following problem.

MULTISTAGE VERTEX COVER (MSVC)

Input: A temporal graph $\mathcal{G} = (V, \mathcal{E}, \tau)$ and two integers $k \in \mathbb{N}, \ell \in \mathbb{N}_0$.

Question: Is there a sequence $\mathcal{S} = (S_1, \ldots, S_\tau)$ such that

(i) for all $i \in \{1, \ldots, \tau\}$, it holds that $S_i \subseteq V$ is a size-at-most-k vertex cover for G_i, and

(ii) for all $i \in \{1, \ldots, \tau - 1\}$, it holds that $|S_i \triangle S_{i+1}| \leq \ell$?

Throughout this paper we assume that $0 < k < |V|$ because otherwise we have a trivial instance. In our model, we follow the recently proposed *multistage* [4, 16, 5, 11] view on classical optimization problems on temporal graphs.

In general, the motivation behind a multistage variant of a classical problem such as VERTEX COVER is that the environment changes over time (here reflected by the changing edge sets in the temporal graph) and a corresponding adaptation of the current solution comes with a cost. In this spirit, the parameter ℓ in the definition of MSVC allows to model that only moderate changes concerning the solution vertex set may be wanted when moving from one layer to the subsequent one. Indeed, in this sense ℓ can be interpreted as a parameter measuring the degree of (non-)conservation [17, 1].

It is immediate that MSVC is NP-hard as it generalizes VERTEX COVER ($\tau = 1$). We will study its parameterized complexity regarding the problem-specific parameters k, τ, ℓ, and some of their combinations, as well as restrictions to temporal graph classes [13].

Related Work. The literature on vertex covering is extremely rich, even when focusing on parameterized complexity studies. Indeed, VERTEX COVER (VC) can be seen as "drosophila" of parameterized algorithmics. Thus, we only consider VC studies closely related to our setting. First, we mention in passing that VC is studied in dynamic graphs [19, 3] and graph stream models [6]. More importantly for us, Akrida et al. [2] studied a variant of VC on temporal graphs. Their model significantly differs from ours: They want an edge to be covered at least once over every time window of some given size Δ. That is, they define a temporal vertex cover as a set $S \subseteq V \times \{1, \ldots, \tau\}$ such that, for every time window of size Δ and for each edge $e = \{v, w\}$ appearing in a layer contained in the time window, it holds that $(v, t) \in S$ or $(w, t) \in S$ for some t in the time window with $(e, t) \in \mathcal{E}$. For their model, they ask whether such an S of small cardinality exists. Note that if $\Delta > 1$, then for some $t \in \{1, \ldots, \tau\}$ the set $S_t := \{v \mid (v, t) \in S\}$ is not necessarily a vertex cover of layer G_t. For $\Delta = 1$, each S_t must be a vertex cover of G_t. However, in Akrida et al.'s model the size of each S_t as well as the size of the symmetric difference between each S_t and S_{t+1} may strongly vary. They provide several hardness results and algorithms (mostly referring to approximation or exact algorithms, but not to parameterized complexity studies).

A second related line of research, not directly referring to temporal graphs though, studies reconfiguration problems which arise when we wish to find a step-by-step transformation between two feasible solutions of a problem such that all intermediate results are feasible solutions as well [18, 15]. Mouawad et al. [22, 21] studied, among other reconfiguration problems, VERTEX COVER RECONFIGURATION which takes as input a graph G, two vertex covers S and T of size at most k each, and an integer τ. The goal is to determine whether there is a sequence $(S = S_1, \ldots, S_\tau = T)$ such that each S_t is a vertex cover of size at most k. The essential difference to our model is that from one "sequence element" to the next only one vertex may be changed and that the input graph does not change over time. Indeed, there is an easy reduction of this model to ours while the opposite direction is unlikely to hold. This is substantiated by the fact that Mouawad et al. [22] showed that VERTEX COVER RECONFIGURATION is fixed-parameter tractable when parameterized by vertex cover size k while we show W[1]-hardness for the corresponding case of MSVC.

Table 1 Overview on our results. The column headings describe the restrictions on the input and each row corresponds to a parameter. p-NP-hard, PK, and NoPK abbreviate para-NP-hard, polynomial problem kernel, and no problem kernel of polynomial size unless coNP ⊆ NP/ poly. † (Obs. 2.5)

	general layers $0 \le \ell < 2k$	general layers $\ell \ge 2k$	tree layers $0 \le \ell < 2k$	one-edge layers $0 \le \ell < 2$
	NP-hard		NP-hard (Thm. 3.1(i))	NP-hard (Thm. 3.1(ii))
τ	p-NP-hard (Thm. 3.1)		p-NP-hard (Thm. 3.1)	FPT, PK (Obs. 5.8)
k	XP, W[1]-h., (Thm. 4.1)	FPT†, NoPK (Thm. 5.1)	XP, W[1]-h. (Thm. 4.1)	*open*, NoPK (Thm. 5.1)
$k+\tau$	FPT, PK (Thm. 5.5)		FPT, PK (Thm. 5.5)	FPT, PK (Thm. 5.5)

Finally, there is also a close relation to the research on dynamic parameterized problems [1, 20]. Krithika et al. [20] studied Dynamic Vertex Cover where one is given two graphs on the same vertex set and a vertex cover for one of them together with the guarantee that the cardinality of the symmetric difference between the two edge sets is upper-bounded by a parameter d. The task then is to find a vertex cover for the second graph that is "close enough" (measured by a second parameter) to the vertex cover of the first graph. They show fixed-parameter tractability and a linear kernel with respect to d.

Our Contributions. Our results, focusing on the three perhaps most natural parameters, are summarized in Table 1.[1] We highlight a few specific results. Multistage Vertex Cover remains NP-hard even if every layer consists of only one edge; clearly, the corresponding hardness reduction then exploits an unbounded number τ of time layers. If one only has two layers, however, one of them being a tree and the other being a path, then again Multistage Vertex Cover already becomes NP-hard. MSVC parameterized by solution size k is fixed-parameter tractable if $\ell \ge 2k$, but becomes W[1]-hard if $\ell < 2k$. Considering the tractability results for Dynamic Vertex Cover [20] and Vertex Cover Reconfiguration [22], this hardness is surprising and is our most technical result. Furthermore, in the former case (parameterization by k with $\ell \ge 2k$) MSVC does not admit a problem kernel of polynomial size unless coNP ⊆ NP/ poly. If one considers the combined parameter $k+\tau$, however, then besides fixed-parameter tractability in *all* cases we also obtain polynomial-sized kernels.

2 Preliminaries

We denote by $\mathbb{N}$ and $\mathbb{N}_0$ the natural numbers excluding and including zero, respectively. For two sets A and B, we denote by $A \triangle B := (A \setminus B) \cup (B \setminus A)$ the symmetric difference of A and B, and by $A \uplus B$ the disjoint union of A and B. We use basic notation from graph theory [8] and parameterized algorithmics [7].

Temporal Graphs. A temporal graph $\mathcal{G}$ is a tuple $(V, \mathcal{E}, \tau)$ consisting of the set of vertices V, the set of temporal edges $\mathcal{E}$, and a discrete time-horizon τ. A temporal edge e is an element in $\binom{V}{2} \times \{1, \ldots, \tau\}$. Equivalently, a temporal graph $\mathcal{G}$ is a vector of static graphs $(G_1, \ldots, G_\tau)$,

[1] Several details and proofs (marked with ⋆) are deferred to the full version of the paper: `https://arxiv.org/abs/1906.00659`.

where each graph is defined over the same vertex set V. We also denote by $V(\mathcal{G})$, $\mathcal{E}(\mathcal{G})$, and $\tau(\mathcal{G})$ the set of vertices, the set of temporal edges, and the discrete time-horizon of $\mathcal{G}$, respectively. The *underlying graph* $G_\downarrow = G_\downarrow(\mathcal{G})$ of a temporal graph $\mathcal{G}$ is the static graph with vertex set $V(\mathcal{G})$ and edge set $\{e \mid \exists t \in \{1,\dots,\tau(\mathcal{G})\} : (e,t) \in E\}$.

General Observations on MSVC. We state some simple but useful observations on MSVC and its relation to Vertex Cover.

▶ **Observation 2.1** (⋆). *Every instance $(\mathcal{G},k,\ell)$ of* MSVC *with $k \geq \sum_{i=1}^{\tau(\mathcal{G})} |E(G_i)|$ is a* yes*-instance.*

▶ **Observation 2.2** (⋆). *Let $(\mathcal{G},k,\ell)$ be an instance of* MSVC*. If $(\mathcal{G},k,\ell)$ is a* yes*-instance, then there is a solution $\mathcal{S} = (S_1,\dots,S_\tau)$ such that $|S_1| = k$ and $k-1 \leq |S_i| \leq k$ for all $i \in \{1,\dots,\tau\}$.*

▶ **Observation 2.3** (⋆). *There is an algorithm that maps any instance (G,k) of* Vertex Cover *in $\tau \cdot |V(G)|^{O(1)}$ time to an equivalent instance $(\mathcal{G},k,\ell)$ of* MSVC *with $\ell = 0$, where $\mathcal{G}$ is a sequence of any τ subgraphs of G such that the underlying graph is G.*

▶ **Observation 2.4** (⋆). *There is a polynomial-time algorithm that maps any instance $(\mathcal{G},k,\ell)$ of* MSVC *with $\ell = 0$ to an equivalent instance $(G_\downarrow(\mathcal{G}),k)$ of* Vertex Cover*.*

▶ **Observation 2.5** (⋆). *An instance $(\mathcal{G},k,\ell)$ of* MSVC *with $\ell \geq 2k$ and $\mathcal{G} = (G_1,\dots,G_\tau)$ can be decided by deciding each instance of the set $\{(G_i,k) \mid 1 \leq i \leq \tau\}$ of* Vertex Cover*-instances.*

3 Hardness On Restricted Inputs

MSVC is NP-hard as it generalizes Vertex Cover ($\tau = 1$). In this section we prove that MSVC remains NP-hard on very restricted inputs.

▶ **Theorem 3.1.** Multistage Vertex Cover *is* NP*-hard even if*

(i) *$\tau = 2$, $\ell = 0$, and the first layer is a path and the second layer is a tree, or*

(ii) *every layer contains only one edge and $\ell = 1$.*

▶ Remark 3.2. Theorem 3.1(i) is tight regarding τ since Vertex Cover (i.e., MSVC with $\tau = 1$) on trees is solvable in polynomial time. Theorem 3.1(ii) is tight regarding ℓ, because in the case of $\ell \neq 1$ either Observation 2.3 or Observation 2.5 is applicable.

Vertex Cover remains NP-complete on cubic Hamiltonian graphs when a Hamiltonian cycle is additionally given in the input [12]—we refer to this problem as Hamiltonian Cubic Vertex Cover (HCVC). To prove Theorem 3.1(i), we give a polynomial-time many-one reduction from HCVC to MSVC with two layers, one being a path, the other being a tree.

▶ **Proposition 3.3** (⋆). *There is a polynomial-time algorithm that maps any instance $(G = (V,E),k,C)$ of* HCVC *to an equivalent instance $(\mathcal{G},k',\ell')$ of* MSVC *with $\tau = 2$ and the first layer G_1 being a path and second layer G_2 being a tree.*

In order to prove Theorem 3.1(ii), we give a polynomial-time many-one reduction from Vertex Cover to MSVC.

▶ **Proposition 3.4** (⋆). *There is a polynomial-time algorithm that maps any instance $(G = (V,E),k)$ of* Vertex Cover *to an equivalent instance $(\mathcal{G},k',\ell')$ of* MSVC *where $\ell' = 1$ and every layer G_i contains only one edge.*

4 Parameter Vertex Cover Size

In this section, we study the parameter size k of the vertex cover of each layer for MSVC. Vertex Cover and Vertex Cover Reconfiguration [22] when parameterized by the vertex cover size are fixed-parameter tractable. We prove that this is no longer true for MSVC (unless FPT = W[1]).

▶ **Theorem 4.1.** Multistage Vertex Cover *parameterized by k is in* XP *and* W[1]*-hard.*

We first show the XP-algorithm (Section 4.1) and then prove W[1]-hardness (Section 4.2).

4.1 An XP-Algorithm

In this section, we prove the following.

▶ **Proposition 4.2.** *Every instance $(\mathcal{G}, k, \ell)$ of* Multistage Vertex Cover *can be decided in $O(\tau(\mathcal{G}) \cdot |V(\mathcal{G})|^{2k+1})$ time.*

In a nutshell, we first consider for each layer all subsets of vertices of size at most k that form a vertex cover. Second, we find a sequence of vertex covers for all layers such that the sizes of the symmetric differences for every two consecutive solutions is at most ℓ. We show that the second step can be solved via computing a directed source-sink path in a helper graph that we call *configuration graph*.

▶ **Definition 4.3.** *Given a temporal graph $\mathcal{G}$, the (k, ℓ)-*configuration graph *of $\mathcal{G}$ is the graph $D = (V = V_1 \uplus \cdots \uplus V_\tau \uplus \{s, t\}, A, \gamma)$ equipped with a function $\gamma : V \to \{V' \subseteq V(\mathcal{G}) \mid |V'| \le k\}$ such that*

(i) *for every $i \in \{1, \ldots, \tau(\mathcal{G})\}$, it holds true that S is a vertex cover of G_i of size at most k if and only if there is a vertex $v \in V_i$ with $\gamma(v) = S$,*

(ii) *there is an arc from v to w, $v, w \in V$, if and only if $v \in V_i$, $w \in V_{i+1}$, and $|\gamma(v) \triangle \gamma(w)| \le \ell$, and*

(iii) *there is an arc (s, v) for all $v \in V_1$ and an arc (v, t) for all $v \in V_\tau$.*

Note that Mouawad et al. [22] used a similar configuration graph to show fixed-parameter tractability of Vertex Cover Reconfiguration parameterized by the vertex cover size k. In the multistage setting the configuration graph is too large for fixed-parameter tractability regarding k. However, we show an XP-algorithm regarding k to construct the configuration graph.

▶ **Lemma 4.4** (⋆). *The (k, ℓ)-configuration graph of a temporal graph $\mathcal{G}$ with n vertices and time horizon τ*

(i) *can be constructed in $O(\tau \cdot n^{2k+1})$ time, and*

(ii) *contains at most $\tau \cdot 2n^k + 2$ vertices and $(\tau - 1)n^{2k} + 2n^k$ arcs.*

▶ **Lemma 4.5** (⋆). MSVC*-instance $(\mathcal{G}, k, \ell)$ is a* `yes`*-instance if and only if there is an s-t path in the (k, ℓ)-configuration graph D of $\mathcal{G}$.*

We are ready to prove Proposition 4.2.

Proof of Proposition 4.2. First, compute the configuration graph D of the instance $(\mathcal{G} = (V, \mathcal{E}, \tau), k, \ell)$ of Multistage Vertex Cover in $O(\tau \cdot |V|^{2k+1})$ time (Lemma 4.4(i)). Then, find an s-t path in D with a breadth-first search in $O(\tau \cdot |V|^{2k})$ time (Lemma 4.4(ii)). If an s-t path is found, then return `yes`, otherwise return `no` (Lemma 4.5). ◀

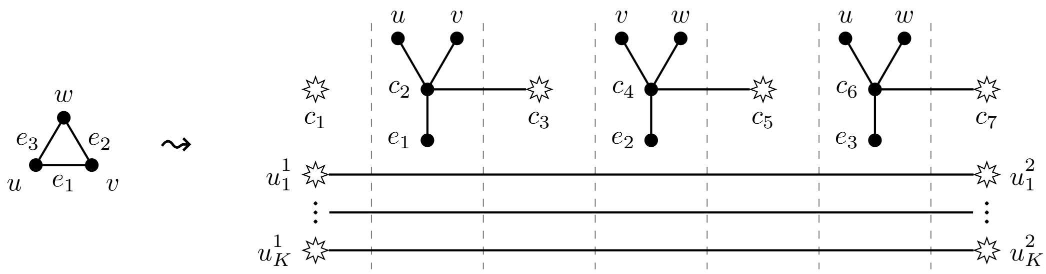

Figure 1 Illustration of Construction 1 on an example graph (left-hand side) and the first seven layers of the obtained graph (right-hand side). Star-shapes illustrate star graphs with $k'+1$ leaves. Dashed vertical lines separate layers.

▶ Remark 4.6. The reason why the algorithm behind Proposition 4.2 is only an XP-algorithm and not an FPT-algorithm regarding k for Multistage Vertex Cover is because we do not have a better upper bound on the number of vertices in the (k,ℓ)-configuration graph for $\mathcal{G}$ than $O(\tau(\mathcal{G})\cdot|V(\mathcal{G})|^k)$. This is due to the fact that we check for each subset of $V(\mathcal{G})$ of size k or $k-1$ whether it is a vertex cover in some layer.

This changes if we consider Minimal Multistage Vertex Cover where we additionally demand the i-th set in the solution to be a *minimal* vertex cover for the layer G_i. Here, we can enumerate for each layer G_i all minimal vertex covers of size at most k (and hence all candidates for the i-th set of the solution) with the folklore search-tree algorithm for vertex cover. This leads to $O(2^k\tau(\mathcal{G}))$ many vertices in the (k,ℓ)-configuration graph (for Minimal Multistage Vertex Cover) and thus to fixed-parameter tractability of Minimal Multistage Vertex Cover parameterized by the vertex cover size k.

However, as we show next it is not likely (unless FPT=W[1]) that one can substantially improve the algorithm behind Proposition 4.2.

4.2 Parameterized Intractability

In this section we show that MSVC is W[1]-hard when parameterized by k. This hardness result is established by the following parameterized reduction from the W[1]-complete [9] Clique problem, where, given an undirected graph G and a positive integer k, the question is whether G contains a clique of size k (that is, k vertices that are pairwise adjacent).

▶ **Proposition 4.7.** *There is an algorithm that maps any instance (G,k) of* Clique *in polynomial time to an equivalent instance $(\mathcal{G},k',\ell)$ of* MSVC *with $k'=2\binom{k}{2}+k+1$, $\ell=2$, and each layer of $\mathcal{G}$ being a tree.*

The proof of Proposition 4.7 is deferred to the end of this section. It is a reduction from Clique where we construct an instance of MSVC from an instance of Clique as follows (see Figure 1 for an illustrative example).

▶ **Construction 1.** Let $(G=(V,E),k)$ be an instance of Clique with $m=|E|$ and $E=\{e_1,\ldots,e_m\}$. Let

$$K=\binom{k}{2},\qquad k'=2K+k+1,\qquad \text{and}\qquad \kappa=K+k+3.$$

We construct a temporal graph $\mathcal{G}=(V',\mathcal{E},\tau)$ as follows. Let V' be initially $V\cup E$ (note that E simultaneously describes the edge set of G and a vertex subset of $\mathcal{G}$). We add the following vertex sets

$$U^t = \{u_j^t \mid j \in \{1, \dots, K\}\},\ t \in \{1, \dots, \kappa + 1\} \text{ and } C = \{c_1, \dots, c_{2m\kappa+1}\}.$$

Let $\mathcal{E}$ be initially empty. We extend the set V' and define $\mathcal{E}$ through the $\tau = 2m\kappa + 1$ layers we construct in the following.

(1) In each layer G_i with i being odd, make c_i the center of a star with $k' + 1$ leaves.

(2) In each layer G_{2mj+1}, $j \in \{0, \dots, \kappa\}$, make each vertex in U^{j+1} the center of a star with $k' + 1$ leaves.

(3) For each $j \in \{0, \dots, \kappa - 1\}$, in each layer G_{2mj+i} with $i \in \{1, \dots, 2m + 1\}$, make u_x^{j+1} adjacent to u_x^{j+2} for each $x \in \{1, \dots, K\}$.

(4) For each i being even, add the edge $\{c_i, c_{i+1}\}$ to G_i and to G_{i+1}.

(5) For each $j \in \{0, \dots, \kappa - 1\}$, for each $i \in \{1, \dots, m\}$, in G_{2mj+2i}, make c_{j2m+2i} adjacent with $e_i = \{v, w\}$, v, and w.

This finishes the construction of $\mathcal{G}$. ⌟

The construction essentially repeats the same gadget (which we call *phase*) κ times where the layer $2m \cdot i + 1$ is simultaneously last layer of phase i and the first layer of phase $i + 1$. In the beginning of phase i, a solution must contain the vertices of U^i. The idea now is that during phase i one has to exchange the vertices of U^i with the vertices of U^{i+1}.

It is not difficult to see that the instance in Construction 1 can be computed in polynomial time. Hence, it remains to prove the equivalence stated in Proposition 4.7. We prepare the proofs of the forward and the backward direction in Sections 4.2.1 and 4.2.2, respectively.

▶ Remark 4.8. We can turn the instance $(\mathcal{G}, k', \ell)$ computed by Construction 1 into an equivalent instance $(\mathcal{G}', k'', \ell)$ where each layer is a tree as follows. Set $k'' = k' + 1$. Add a vertex x to $\mathcal{G}$. In each layer of $\mathcal{G}$, make x a star with $k'' + 1$ vertices and connect x with exactly one vertex of each connected component. Note that in every solution x is contained in a vertex cover for each layer in $\mathcal{G}'$.

4.2.1 Forward direction

The forward direction of Proposition 4.7 is—in a nutshell—as follows: If $V' \cup E'$ with $V' \subseteq V$ and $E' \subseteq E$ correspond to the vertex set and edge set of a clique of size k, then there are K layers in each phase covered by $V' \cup E'$. Hence, having K layers where no vertices from C have to be exchanged, in each phase t we can exchange all vertices from U^t to U^{t+1}. Starting with set $S_1 = U^1 \cup V' \cup E' \cup \{c_1\}$ then yields a solution.

▶ **Lemma 4.9 (⋆).** *Let (G, k) be an instance of* Clique *and $(\mathcal{G}, k', \ell)$ be the instance of* Multistage Vertex Cover *resulting from Construction 1. If (G, k) is a* `yes`*-instance, then $(\mathcal{G}, k', \ell)$ is a* `yes`*-instance.*

4.2.2 Backward direction

In this section we prepare the proof of the backward direction for the proof of Proposition 4.7. We first show that if an instance of Multistage Vertex Cover computed by Construction 1 is a `yes`-instance, then it is safe to assume that neither two vertices are deleted from nor added to a vertex cover in a consecutive step (we refer to these solutions as *smooth*, see Definition 4.11). Moreover, a vertex from C is only exchanged with another vertex from C and, at any time, there is exactly one vertex from C contained in the solution (similarly to the constructed solution in Lemma 4.9). We call these solutions *one-centered* (Definition 4.13).

We then prove that there must be a phase t for any one-centered solution that is deleting at least $\binom{k}{2}$ times a vertex from “past” sets $U_{t'}$, $t' \le t$. This at hand, we prove that such a phase witnesses a clique of size k.

That a solution needs to contain at least one vertex from C at any time follows immediately from the fact that there is either an edge between two vertices in C or there is a vertex in C which is the center of a star with $k' + 1$ leaves.

▶ **Observation 4.10.** *Let $(\mathcal{G}, k', \ell)$ from Construction 1 be a* `yes`*-instance. Then for each solution $(S_1, \dots, S_\tau)$ it holds true that $|S_i \cap C| \ge 1$ for all $i \in \{1, \dots, \tau(\mathcal{G})\}$.*

In the remainder of this section we denote the vertices which are removed from the set S_{i-1} and added to the next set S_i in a solution $S = (\dots, S_{i-1}, S_i, \dots)$ by

$$S_{i-1} \diamondminus S_i := (S_{i-1} \setminus S_i, S_i \setminus S_{i-1}).$$

If $S_{i-1} \setminus S_i$ or $S_i \setminus S_{i-1}$ have size one, then we will omit the brackets of the singleton.

▶ **Definition 4.11.** *A solution $\mathcal{S} = (S_1, \dots, S_\tau)$ for $(\mathcal{G}, k', \ell)$ from Construction 1 is* smooth *if for all $i \in \{2, \dots, \tau\}$ we have $|S_{i-1} \setminus S_i| \le 1$ and $|S_{i-1} \setminus S_i| \le 1$.*

▶ **Observation 4.12.** *Let $(\mathcal{G}, k', \ell)$ from Construction 1 be a* `yes`*-instance. Then there is a smooth solution $(S_1, \dots, S_\tau)$.*

Proof. By Observation 2.1, we know that there is a solution $\mathcal{S} = (S_1, \dots, S_\tau)$ such that $|S_1| = k'$ and $k' - 1 \le |S_i| \le k'$ for all $i \in \{1, \dots, \tau\}$. Hence, for all $i \in \{2, \dots, \tau\}$ it holds true that $\big||S_i| - |S_{i-1}|\big| \le 1$. It follows that $|S_{i-1} \setminus S_i| \le 1$ and $|S_{i-1} \setminus S_i| \le 1$, and thus, $\mathcal{S}$ is a smooth solution. ◀

Our goal is to prove the existence of the following type of solutions.

▶ **Definition 4.13.** *A smooth solution $\mathcal{S} = (S_1, \dots, S_\tau)$ for $(\mathcal{G}, k', \ell)$ from Construction 1 is* one-centered *if*

(i) *for all $i \in \{1, \dots, \tau\}$ we have $|S_i \cap C| = 1$, and*
(ii) *for all $i \in \{2, \dots, \tau\}$ and $S_{i-1} \diamondminus S_i = (a, b)$ we have that $a \in C \iff b \in C$.*

We now show that in the output instance of Construction 1, there are solutions (if there is any) where $c_1 \in C$ is the only vertex from C in the first set of the solution.

▶ **Lemma 4.14 (⋆).** *Let $(\mathcal{G}, k', \ell)$ from Construction 1 be a* `yes`*-instance. Then there is a smooth solution $(S_1, \dots, S_\tau)$ such that $S_1 \cap C = \{c_1\}$.*

Next, we show that there are solutions such that whenever we remove a vertex in C from the vertex cover, then we simultaneously add another vertex from C to the vertex cover. Formally, we prove the following.

▶ **Lemma 4.15 (⋆).** *Let $(\mathcal{G}, k', \ell)$ from Construction 1 be a* `yes`*-instance. Then there is a smooth solution $(S_1, \dots, S_\tau)$ such that $S_1 \cap C = \{c_1\}$ and for all i with $S_{i-1} \diamondminus S_i = (a, c)$ and $c \in C$ we also have $a \in C$.*

Combining Observation 4.10 and Lemma 4.15, we can assume that given a `yes`-instance, there is a solution which is one-centered.

▶ **Corollary 4.16.** *Let $(\mathcal{G}, k', \ell)$ from Construction 1 be a* `yes`*-instance. Then, there is a solution $\mathcal{S}$ which is one-centered.*

Table 2 Note that $\varepsilon_i - \varepsilon_{i-1} = |F_i^t| - |F_{i-1}^t| - (|Y_i^t| - |Y_{i-1}^t|) - (f_i^t - f_{i-1}^t)$.

$S_{i-1}^t \diamondminus S_i^t$		$\lvert F_i^t\rvert - \lvert F_{i-1}^t\rvert$	$-(\lvert Y_i^t\rvert - \lvert Y_{i-1}^t\rvert)$	$-(f_i^t - f_{i-1}^t)$	$\varepsilon_i - \varepsilon_{i-1}$
(u,b)	$b \in E$	$\in \{-1,0\}$	$\in \{0,1\}$	1	$\in \{0,1,2\}$
	$b \in \widehat{U}_{\kappa+1}$	$\in \{-1,0\}$	1	0	$\in \{0,1\}$
	$b \in V$, $b = \varnothing$	$\in \{-1,0\}$	1	1	$\in \{1,2\}$
(a,u)	$a \in E$	0	$\in \{1,2\}$	-1	$\in \{0,1\}$
	$a \in V$, $a = \varnothing$	0	1	-1	0
(a,v)	$a \in E$	0	$\in \{1,2\}$	0	$\in \{1,2\}$
	$a \in V$, $a = \varnothing$	0	1	0	1
(a,e)	$a \in V$	0	1	0	1
	$a \in E$, $a = \varnothing$	0	$\in \{0,1\}$	0	$\in \{0,1\}$

In the remainder of this section, for each $t \in \{1,\dots,\kappa+1\}$ let the union of U^i for all $i \le t$ be denoted by

$$\widehat{U}_t = \bigcup_{i=1}^t U^i.$$

We introduce further notation regarding a one-centered solution $\mathcal{S} := (S_1^1,\dots,S_{2m+1}^1 = S_1^2,\dots,\dots,S_1^\kappa,\dots,S_{2m+1}^\kappa)$ for $(\mathcal{G},k',\ell)$. Here, S_i^t is the i-th set of phase t and thus the $(2m(t-1)+i)$-th set of $\mathcal{S}$. We define the sets

$$Y_i^t := \{e_j \in S_i^t \cap E \mid 2j \ge i\} \quad \text{and} \quad F_i^t := \{j > i \mid S_{j-1}^t \diamondminus S_j^t = (u,b) \text{ with } u \in \widehat{U}_t\}.$$

Set Y_i^t is the set of vertices e_j from E in S_i^t such that the corresponding layer for e_j in phase t is not before the layer with index i in phase t. Set F_i^t is the set of indices greater than i of layers from $\mathcal{G}$ in phase t where a vertex from $\widehat{U}_t$ is not carried over to the next layer's vertex cover. We now show that there is a phase t where $|F_1^t| \ge K$.

▶ **Lemma 4.17** (★). *Let $\mathcal{S} = (S_1^1,\dots,S_{2m+1}^1 = S_1^2,\dots,\dots,S_1^\kappa,\dots,S_{2m+1}^\kappa)$ be a one-centered solution to $(\mathcal{G},k',\ell)$ from Construction 1 being a* yes*-instance. Then, there is a $t \in \{1,\dots,\kappa\}$ such that $|F_1^t| \ge K$.*

In the remainder of this section the value

$$f_i^t := |S_i^t \cap \widehat{U}_{\kappa+1}| - K$$

describes the number of vertices in $\widehat{U}_{\kappa+1}$ which we could remove from S_i^t such that S_i^t is still a vertex cover for $G_{2m(t-1)+i}$ (the i-th layer of phase t). Observe that $f_i^t \ge 0$ for all $i \in \{1,\dots,2m+1\}$ and $t \in \{1,\dots,\kappa\}$, because we need in each layer exactly K vertices from $\widehat{U}_{\kappa+1}$ in the vertex cover.

We now derive an invariant which must be true in each phase.

▶ **Lemma 4.18** (★). *Let $\mathcal{S} = (S_1^1,\dots,S_{2m+1}^1 = S_1^2,\dots,\dots,S_1^\kappa,\dots,S_{2m+1}^\kappa)$ be a one-centered solution to $(\mathcal{G},k',\ell)$ from Construction 1 being a* yes*-instance. Then, for all $t \in \{1,\dots,\kappa\}$ and $i \in \{1,\dots,2m+1\}$, it holds true that $|F_i^t| - |Y_i^t| \le f_i^t$.*

Proof. Define $\varepsilon_i = |F_i^t| - |Y_i^t| - f_i^t$ for all $i \in \{1,\dots,2m+1\}$. We show that $\varepsilon_i - \varepsilon_{i-1} \ge 0$ for all $i \in \{1,\dots,2m+1\}$. Since $\mathcal{S}$ is one-centered, in Table 2 all relevant tuples for $S_{i-1}^t \diamondminus S_i^t$ are shown.

Now assume towards a contradiction that there is a $j \in \{1, \dots, 2m+1\}$ such that $\varepsilon_j > 0$. Since $\varepsilon_i - \varepsilon_{i-1} \geq 0$ for all $i \in \{1, \dots, 2m+1\}$, we have that $\varepsilon_{2m+1} > 0 \iff |F^t_{2m+1}| - |Y^t_{2m+1}| > f^t_{2m+1}$. By definition, we have that $|F^t_{2m+1}| = 0$ and $|Y^t_{2m+1}| = 0$ Moreover, since $\mathcal{S}$ is a solution and each vertex cover needs at least K vertices from $\widehat{U}_\tau$, we have that $f^t_{2m+1} \geq 0$. It follows that $0 = |F^t_{2m+1}| - |Y^t_{2m+1}| > f^t_{2m+1} \geq 0$, yielding a contradiction. ◀

Next, we prove that in a phase t with $|F^t_1| \geq K$, there are at most k vertices from V contained in the union of the vertex covers of phase t.

▶ **Lemma 4.19** (⋆). *Let $\mathcal{S} = (S^1_1, \dots, S^1_{2m+1} = S^2_1, \dots, \dots, S^\kappa_1, \dots, S^\kappa_{2m+1})$ be a one-centered solution to $(\mathcal{G}, k', \ell)$ from Construction 1 being a* `yes`*-instance, and let $t \in \{1, \dots, \kappa\}$ be such that $|F^t_1| \geq K$. Then, $|\bigcup_{i=1}^{2m+1} S^t_i \cap V| \leq k$.*

Proof. From Lemma 4.18, we know that $|Y^t_1| \geq K - f^t_1$. Let $|Y^t_1| = K - f^t_1 + \lambda$ for some $\lambda \in \mathbb{N}_0$, and $\varepsilon_i = |F^t_i| - |Y^t_i| - f^t_i$, for all $i \in \{1, \dots, 2m+1\}$.

We now show that there are at most λ layers where we exchange a vertex currently in the vertex cover with a vertex in V. Let $i \in \{2, \dots, 2m+1\}$ such that $S^t_{i-1} \diamond S^t_i = (a, v)$ with $v \in V$. From Table 2 (recall that one-centered solutions are smooth), we know that $\varepsilon_i \geq \varepsilon_{i-1} + 1$.

Assume towards a contradiction that there are $\lambda + 1$ many of these exchanges. Then, there is a $j \in \{1, \dots, 2m+1\}$ such that

$$\begin{aligned} &\varepsilon_j \geq \varepsilon_1 + \lambda + 1 = |F^t_1| - |Y^t_1| - f^t_1 + \lambda + 1 \geq K - (K - f^t_1 + \lambda) - f^t_1 + \lambda + 1 \geq 1 \\ \iff\ & |F^t_j| - |Y^t_j| > f^t_j. \end{aligned}$$

This contradicts the invariant of Lemma 4.18.

In the beginning of phase t, we have at most $k - \lambda$ vertices from V in the vertex cover, because $|S^t_1 \cap V| \leq K + k - |Y^t_1| - f^t_1 = K + k - (K - f^t_1 + \lambda) - f^t_1 = k - \lambda$. Since there are at most λ many exchanges $S^t_{i-1} \diamond S^t_i = (a, v)$ where $v \in V$ and $i \in \{2, \dots, 2m+1\}$, we know that the vertex set $\bigcup_{i=1}^{2m+1} S^t_i \cap V$ is of size at most k. ◀

4.2.3 Proof of Proposition 4.7

Proof of Proposition 4.7. Let (G, k) be an instance of Clique and $(\mathcal{G}, k', \ell)$ be the instance of MSVC resulting from Construction 1. Observe that Construction 1 runs in polynomial time. We prove that (G, k) is a `yes`-instance of Clique if and only if $(\mathcal{G}, k', \ell)$ is a `yes`-instance of MSVC.

($\Rightarrow$) It follows from Lemma 4.9 that $(\mathcal{G}, k', \ell)$ is a `yes`-instance if (G, k) is a `yes`-instance.

($\Leftarrow$) Let $(\mathcal{G}, k', \ell)$ be a `yes`-instance. From Corollary 4.16 it follows that there is a one-centered solution $\mathcal{S} = (S^1_1, \dots, S^1_{2m+1} = S^2_1, \dots, \dots, S^\kappa_1, \dots, S^\kappa_{2m+1})$ for $(\mathcal{G}, k', \ell)$. By Lemma 4.17, there is a $t \in \{1, \dots, \kappa\}$ such that $|F^t_1| \geq K = \binom{k}{2}$. By Lemma 4.19, we know that $|\bigcup_{i=1}^{2m+1} S^t_i \cap V| \leq k$. Now we identify the clique of size k in G. Since $|F^t_1| \geq K$, we know that, by Construction 1, at least K layers are covered by vertices in $V \cup E \cup \widehat{U}_{\kappa+1} \cup \{c^t_{2j+1} \mid j \in \{1, \dots, m\}\}$ in phase t. Note that each of these layers corresponds to an edge $e = \{v, w\}$ in G and that we need in particular the vertices v and w in the vertex cover. Since we have at most k vertices in $\bigcup_{i=1}^{2m+1} S^t_i \cap V$, these vertices induce a clique of size k in G.

Finally, following Remark 4.8, we can turn each layer into a tree preserving equivalence. The W[1]-hardness of Clique [9] regarding k and that $k' \in O(k^2)$ then finish the proof. ◀

5 On Efficient Data Reduction

In this section, we study the possibility of effective data reduction for MSVC when parameterized by k, τ, and $k + \tau$, that is, the possible existence of problem kernels of polynomial size. We prove that unless coNP $\subseteq$ NP/ poly, MSVC admits no problem kernel of size polynomial in k (Section 5.1). Yet, when combining k and τ, we prove a problem kernel of size $O(k^2\tau)$ (Section 5.2). Moreover, we prove a problem kernel of size 5τ when each layer consists of only one edge (Section 5.3). Recall that MSVC is para-NP-hard regarding τ even if each layer is a tree.

5.1 No problem kernel of size polynomial in k

We prove that if

(i) each layer consists only of one edge and $\ell = 1$, or

(ii) if each layer is planar and $\ell \geq 2k$,

then MSVC admits no kernel of size polynomial in k unless coNP $\subseteq$ NP/ poly. Recall that MSVC parameterized by k is fixed-parameter tractable in case of (i) (see Observation 2.5), while we left open whether it also holds true in case (ii).

▶ **Theorem 5.1.** *Unless* coNP $\subseteq$ NP/ poly*,* MSVC *admits no polynomial kernel when parameterized by k, even if*

(i) *each layer consists of one edge and $\ell = 1$, or if*

(ii) *each layer is planar and $\ell \geq 2k$.*

We prove Theorem 5.1 using AND-compositions.

▶ **Definition 5.2.** *An* AND-composition *for a parameterized problem L is an algorithm that given p instances $(x_1, k), \dots, (x_p, k)$ of L, computes in time polynomial in $\sum_{i=1}^{p} |x_i|$ an instance (y, k') of L such that*

(i) *$(y, k') \in L$ if and only if $(x_i, k) \in L$ for all $i \in \{1, \dots, p\}$, and*

(ii) *k' is polynomially upper-bounded in k.*

Drucker [10] showed that if a parameterized problem whose unparameterized version is NP-hard admits an AND-composition, then coNP $\subseteq$ NP/ poly. Note that coNP $\subseteq$ NP/ poly implies a collapse of the polynomial-time hierarchy to its third level [23]. In the proof of Theorem 5.1(i), we use the following.

▶ **Construction 2.** Let $(\mathcal{G}_1 = (V, \mathcal{E}_1, \tau_1), k, \ell), \dots, (\mathcal{G}_p = (V, \mathcal{E}_p, \tau_p), k, \ell)$ be p instances of MSVC where each layer consists of one edge and $\ell = 1$. We construct an instance $(\mathcal{G} = (V, \mathcal{E}, \tau), k, \ell)$ of MSVC as follows. Denote by $(G_1^i, \dots, G_{\tau_i}^i)$ the sequence of layers of $\mathcal{G}_i$. Initially, let $\mathcal{G}$ be the temporal graph with layer sequence $((G_j^i)_{1 \leq j \leq \tau_i})_{1 \leq i \leq p}$. Next, for each $i \in \{1, \dots, p-1\}$, insert between $G_{\tau_i}^i$ and G_1^{i+1} the sequence $(H_1^i, H_2^i, \dots, H_{2k}^i) = (G_{\tau_i}^i, G_1^{i+1}, \dots, G_1^{i+1})$. This finishes the construction. Note that $\tau = 2k(p-1) + \sum_{i=1}^{p} \tau_i$. ⌟

Construction 2 gives an AND-composition used in the proof of Theorem 5.1(i).

▶ **Proposition 5.3 (⋆).** MSVC *where each layer consists of one edge and $\ell = 1$ AND-composes into itself parameterized by k.*

Turning a set of input instances of Vertex Cover on planar graphs (this is equivalent to MSVC with one layer which is a planar graph) into a sequence gives an AND-composition used in the proof of Theorem 5.1(ii).

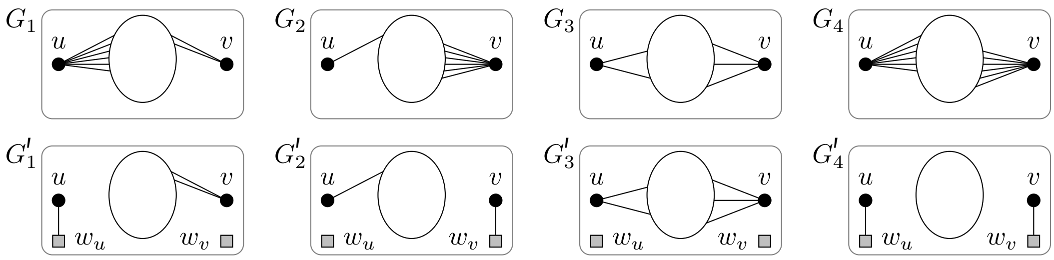

Figure 2 Illustration of Reduction Rule 2, exemplified for two vertices u, v and $k = 5$. The vertices w_v, w_u (gray squares) are introduced by the application of Reduction Rule 2.

▶ **Proposition 5.4** (⋆). MSVC *with one layer being a planar graph AND-composes into* MSVC *parameterized by k with $\ell \geq 2k$ and each layer being planar.*

Proof of Theorem 5.1. Using Drucker's result for AND-copositions [10], Propositions 5.3 and 5.4 prove Theorem 5.1(i) and (ii), respectively. Recall that MSVC where each layer consists of one edge (Theorem 3.1) and Vertex Cover on planar graphs [14] are NP-hard. ◀

5.2 A problem kernel of size $O(k^2\tau)$

MSVC remains NP-hard for $\tau = 2$, even if each layer is a tree (Theorem 3.1). Moreover, MSVC does not admit a problem kernel of size polynomial in k, even if each layer consists of one edge (Theorem 5.1). Yet, when combining both parameters we obtain a problem kernel of cubic size.

▶ **Theorem 5.5.** *There is an algorithm that maps any instance $(\mathcal{G} = (V, \mathcal{E}, \tau), k, \ell)$ of* MSVC *in time $O(|V|^2\tau)$ to an instance $(\mathcal{G}', k, \ell)$ of* MSVC *with at most $2k^2\tau$ vertices and $k^2\tau$ temporal edges.*

To prove Theorem 5.5, we apply three polynomial-time data reduction rules. These reduction rules can be understood as temporal variants of the folklore reduction rules for Vertex Cover. Our first reduction rule is immediate.

▶ **Reduction Rule 1** (Isolated vertices). *If there is some vertex $v \in V$ such that $e \cap v = \emptyset$ for all $e \in E(G_\downarrow)$, then delete v.*

For Vertex Cover when asking for a vertex cover of size q, there is the well-known reduction rule dealing with high-degree vertices: If there is a vertex v of degree larger than q, then delete v and its incident edges and decrease q by one. For MSVC a high-degree vertex can only appear in some layers, and hence deleting this vertex is in general not correct. However, there is a temporal variant of the high-degree rule as follows.

▶ **Reduction Rule 2** (High degree). *If there exists a vertex v such that there is an inclusion-maximal subset $J \subseteq \{1, \dots, \tau\}$ such that $\deg_{G_i}(v) > k$ for all $i \in J$, then add a vertex w_v to V and for each $i \in J$, remove all edges incident to v in G_i, and add the edge $\{v, w_v\}$.*

See Figure 2 for an illustration. We now show how Reduction Rule 2 can be applied and that it does not turn a `yes`-instance into a `no`-instance or vice versa.

▶ **Lemma 5.6** (⋆). *Reduction Rule 2 is correct and exhaustively applicable in $O(|V|^2\tau)$ time.*

Similarly as in the reduction rules for Vertex Cover, we now count the number of edges in each layer: If more than k^2 edges are contained in one layer, then no set of k vertices each of degree at most k can cover more than k^2 edges.

▶ **Reduction Rule 3** (no-instance). *If none of Reduction Rules 1 and 2 is applicable and there is a layer with more than k^2 edges, then output a trivial* no*-instance.*

We are ready to prove that when none of the Reduction Rules 1 to 3 can be applied, then the instance contains "few" vertices and temporal edges.

▶ **Lemma 5.7** (★). *Let $(\mathcal{G}, k, \ell)$ be an instance of* MSVC *such that none of Reduction Rules 1 to 3 is applicable. Then $\mathcal{G}$ consists of at most $2k^2\tau(\mathcal{G})$ vertices and $k^2\tau(\mathcal{G})$ temporal edges.*

We are ready to prove the main result of this section.

Proof of Theorem 5.5. Apply Reduction Rules 1 to 3 exhaustively in $O(|V|^2\tau)$ time to obtain an equivalent instance $(\mathcal{G}', k, \ell)$. Due to Lemma 5.7, $\mathcal{G}'$ consists of at most $2k^2\tau$ vertices and at most $k^2\tau$ temporal edges. ◀

5.3 A problem kernel of size 5τ

MSVC when each layer is a tree does not admit a problem kernel of any size in τ unless P = NP. Yet, when each layer consists of only one edge, then each instance of MSVC contains at most τ edges and, hence, at most 2τ non-isolated vertices. Thus, MSVC admits a straight-forward problem kernel of size linear in τ.

▶ **Observation 5.8.** *Let $(\mathcal{G} = (V, \mathcal{E}, \tau), k, \ell)$ be an instance of* MSVC *where each layer consists of one edge. Then we can compute in $O(|V| \cdot \tau)$ time an instance $(\mathcal{G}', k, \ell)$ of size at most 5τ.*

Proof. Observe that we can immediately output a trivial yes-instance if $k \geq \tau$ (Observation 2.1) or $\ell \geq 2$ (Observation 2.5). Hence, assume that $k \leq \tau - 1$ and $\ell \leq 1$. Apply Reduction Rule 1 exhaustively on $(\mathcal{G}, k, \ell)$ to obtain $(\mathcal{G}', k, \ell)$. Since there are τ edges in $\mathcal{G}$, there are at most 2τ vertices in $\mathcal{G}'$. It follows that the encoding length of $(\mathcal{G}', k, \ell)$ is at most 5τ. ◀

6 Conclusion

We introduced Multistage Vertex Cover, proved it to be NP-hard even on restricted inputs, and studied its parameterized complexity regarding the natural parameters k, ℓ, and τ (each given as input). We leave open whether MSVC parameterized by k is fixed-parameter tractable when each layer consists of only one edge (see Table 1). Moreover, it is open whether MSVC remains NP-hard on two layers each being a path (that is, strengthening Theorem 3.1(i)).

References

1 Faisal N. Abu-Khzam, Judith Egan, Michael R. Fellows, Frances A. Rosamond, and Peter Shaw. On the parameterized complexity of dynamic problems. *Theor. Comput. Sci.*, 607:426–434, 2015.

2 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal Vertex Cover with a Sliding Time Window. In *Proc. of 45th ICALP*, volume 107 of *LIPIcs*, pages 148:1–148:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

3 Josh Alman, Matthias Mnich, and Virginia Vassilevska Williams. Dynamic Parameterized Problems and Algorithms. In *Proc. of 44th ICALP*, volume 80 of *LIPIcs*, pages 41:1–41:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

4 Evripidis Bampis, Bruno Escoffier, Michael Lampis, and Vangelis Th. Paschos. Multistage Matchings. In *Proc. of 16th SWAT*, volume 101 of *LIPIcs*, pages 7:1–7:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

5 Evripidis Bampis, Bruno Escoffier, and Alexandre Teiller. Multistage Knapsack. In *Proc. of 44th MFCS*, volume 138 of *LIPIcs*, pages 22:1–22:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

6 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via Sampling with Applications to Finding Matchings and Related Problems in Dynamic Graph Streams. In *Proc. of 27th SODA*, pages 1326–1344. SIAM, 2016.

7 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

8 Reinhard Diestel. *Graph Theory*, volume 173 of *GTM*. Springer, 5th edition, 2016.

9 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

10 Andrew Drucker. New Limits to Classical and Quantum Instance Compression. *SIAM J. Comput.*, 44(5):1443–1479, 2015.

11 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility Location in Evolving Metrics. In *Proc. of 41st ICALP*, volume 8573 of *LNCS*, pages 459–470. Springer, 2014.

12 Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. Maximum independent sets in 3- and 4-regular Hamiltonian graphs. *Discrete Math.*, 310(20):2742–2749, 2010.

13 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal Graph Classes: A View Through Temporal Separators. *Theor. Comput. Sci.*, 2019. In press.

14 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976.

15 Parikshit Gopalan, Phokion G Kolaitis, Elitza Maneva, and Christos H Papadimitriou. The connectivity of Boolean satisfiability: computational and structural dichotomies. *SIAM J. Comput.*, 38(6):2330–2355, 2009.

16 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing Bases: Multistage Optimization for Matroids and Matchings. In *Proc. of 41st ICALP*, volume 8572 of *LNCS*, pages 563–575. Springer, 2014.

17 Sepp Hartung and Rolf Niedermeier. Incremental list coloring of graphs, parameterized by conservation. *Theor. Comput. Sci.*, 494:86–98, 2013.

18 Takehiro Ito, Erik D Demaine, Nicholas JA Harvey, Christos H Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011.

19 Yoichi Iwata and Keigo Oka. Fast Dynamic Graph Algorithms for Parameterized Problems. In *Proc. of 12th SWAT*, volume 8503 of *LNCS*, pages 241–252. Springer, 2014.

20 R. Krithika, Abhishek Sahu, and Prafullkumar Tale. Dynamic Parameterized Problems. *Algorithmica*, 80(9):2637–2655, 2018.

21 Amer Mouawad, Naomi Nishimura, Venkatesh Raman, and Sebastian Siebertz. Vertex cover reconfiguration and beyond. *Algorithms*, 11(2):20, 2018.

22 Amer E Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017.

23 Chee-Keng Yap. Some Consequences of Non-Uniform Conditions on Uniform Classes. *Theor. Comput. Sci.*, 26:287–300, 1983.