Multistage s-t Path: Confronting Similarity with Dissimilarity in Temporal Graphs*

Till Fluschnik
Technische Universität Berlin, Algorithmics and Computational Complexity, Germany
till.fluschnik@tu-berlin.de

Rolf Niedermeier
Technische Universität Berlin, Algorithmics and Computational Complexity, Germany
rolf.niedermeier@tu-berlin.de

Carsten Schubert
Technische Universität Berlin, Algorithmics and Computational Complexity, Germany
carsten.gm.schubert@campus.tu-berlin.de

Philipp Zschoche
Technische Universität Berlin, Algorithmics and Computational Complexity, Germany
zschoche@tu-berlin.de

Abstract

Addressing a quest by Gupta et al. [ICALP'14], we provide a first, comprehensive study of finding a short s-t path in the multistage graph model, referred to as the Multistage s-t Path problem. Herein, given a sequence of graphs over the same vertex set but changing edge sets, the task is to find short s-t paths in each graph ("snapshot") such that in the found path sequence the consecutive s-t paths are "similar". We measure similarity by the size of the symmetric difference of either the vertex set (vertex-similarity) or the edge set (edge-similarity) of any two consecutive paths. We prove that these two variants of Multistage s-t Path are already NP-hard for an input sequence of only two graphs and maximum vertex degree four. Motivated by this fact and natural applications of this scenario e.g. in traffic route planning, we perform a parameterized complexity analysis. Among other results, for both variants, vertex- and edge-similarity, we prove parameterized hardness (W[1]-hardness) regarding the parameter path length (solution size) for both variants, vertex- and edge-similarity. As a further conceptual study, we then modify the multistage model by asking for *dissimilar* consecutive paths. One of our main technical results (employing so-called representative sets known from non-temporal settings) is that dissimilarity allows for fixed-parameter tractability for the parameter solution size, contrasting the W[1]-hardness of the corresponding similarity case. We also provide partially positive results concerning efficient and effective data reduction (kernelization).

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

Keywords and phrases Temporal graphs, shortest paths, consecutive similarity, consecutive dissimilarity, parameterized complexity, kernelization, representative sets in temporal graphs

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.43

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.07569`.

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

* Some results are based on the third author's bachelor thesis [34].

31st International Symposium on Algorithms and Computation (ISAAC 2020).
Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 43; pp. 43:1–43:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Finding short paths is perhaps the most fundamental task in algorithmic graph theory and network analysis. There are numerous applications, including operations research, robotics, social network analysis, traffic and transportation, and VLSI design. More specifically, we are concerned with finding a short path connecting two designated vertices s and t. It is fair to say that for *static* graphs the algorithmics (also from a practical side) of finding short(est) paths is very well understood. This is much less so when considering path finding in *temporal* graphs, that is, graphs whose edge sets change over time[1], a framework that in recent years received more and more attention in the field of network science. For instance, models concerned with disease spreading or traffic routing typically are more realistic when taking into account that links between network nodes change over time. In this work, we study path finding in temporal graphs with the additional ("multistage") assumption that s-t-paths for consecutive snapshots of the temporal graph shall be sufficiently "similar". We confront this with the opposite view that s-t-paths for consecutive snapshots of the temporal graph shall be significantly "dissimilar". Herein, similarity can naturally be measured both by comparing the edge sets of the s-t paths or the vertex sets of the s-t paths. Altogether, we end up with four natural problem variants.

A few words on motivation. Both scenarios address different aspects of robustness in an environment changing over time. Let us first look at the dissimilarity scenario. Here one may think of a situation where because of necessary recovery or cleansing costs (in pandemic times one may think of disinfection measures) one wants to avoid that subsequent "agents" on the way from start to goal share too many parts of their routing paths. Moreover, one may also think of applications in the context of so-called VIP routing, which address security aspects [15, 17]. As to the similarity scenario, one may think of robustness in the sense of "path maintenance": every deviation from the path used before causes additional costs (set up, preparation, checking) and thus shall be kept at a minimum. This can be interpreted in the spirit of incremental changes (evolutionary rather than radical changes) [8, 23].

Formally, a temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ consists of a set V of vertices and lifetime τ many edge sets $E_1, E_2, \ldots, E_\tau$ over V. Finding an s-t path over time, also known as temporal s-t path, has already been studied [35, 25]. There, however, a path may use edges from $\bigcup_{i=1}^{\tau} E_i$, while in our setting we search for path sequences consisting of τ paths, one for each E_i. With focusing on similar *and* dissimilar paths here, however, we introduce a new view on finding paths in temporal graphs. More specifically, addressing a quest of Gupta et al. [22], one of the first studies on multistage problems, this paper initiates a study of finding short s-t paths in the *multistage* model, that is, finding a short s-t path in each *snapshot* (V, E_i) of the temporal graph $\mathcal{G}$ such that consecutive s-t paths do not differ too much; formally, we have the following (Π refers to a requested property of a solution path):

Π Multistage s-t Path (Π-MstP)

Input: A temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$, two distinct vertices $s, t \in V$, and two integers $k, \ell \in \mathbb{N}_0$.

Question: Is there a sequence $(P_1, P_2, \ldots, P_\tau)$ such that P_i is an s-t path in (V, E_i) with $|V(P_i)| \leq k$ for all $i \in \{1, \ldots, \tau\}$, and $\text{dist}_\Pi(P_i, P_{i+1}) \leq \ell$ for all $i \in \{1, \ldots, \tau - 1\}$?

[1] Holme and Saramäki [26, 27] and Michail [31] survey algorithmic aspects of temporal graphs.

The multistage model requests snapshot solutions such that (in time) consecutive ones are similar to each other. Herein, similarity is measured by the symmetric difference of the sets describing the consecutive snapshot solutions. For paths, there are two natural choices for comparing: the sets of vertices and the sets of edges. Thus, we obtain two distance measures defined as follows.

$$\mathrm{dist}_{\mathrm{V}\triangle\mathrm{V}}(P_i, P_{i+1}) \coloneqq |V(P_i) \triangle V(P_{i+1})| \qquad (\mathrm{V}\triangle\mathrm{V}\text{-}\textsc{MstP}),$$
$$\mathrm{dist}_{\mathrm{E}\triangle\mathrm{E}}(P_i, P_{i+1}) \coloneqq |E(P_i) \triangle E(P_{i+1})| \qquad (\mathrm{E}\triangle\mathrm{E}\text{-}\textsc{MstP}).$$

Confronting the similarity request of the multistage framework with a dissimilarity request instead leads to the following.

$$\mathrm{dist}_{\mathrm{V}\cap\mathrm{V}}(P_i, P_{i+1}) \coloneqq |(V(P_i) \cap V(P_{i+1})) \setminus \{s,t\}| \qquad (\mathrm{V}\cap\mathrm{V}\text{-}\textsc{MstP}),$$
$$\mathrm{dist}_{\mathrm{E}\cap\mathrm{E}}(P_i, P_{i+1}) \coloneqq |E(P_i) \cap E(P_{i+1})| \qquad (\mathrm{E}\cap\mathrm{E}\text{-}\textsc{MstP}).$$

Note that we can easily compute each of the four distance measures in linear time.

In the following, we study the classical and parameterized complexity of all four variants E△E-MstP, V△V-MstP, V∩V-MstP, and E∩E-MstP. With performing a parameterized complexity analysis, we do not only aim for a better understanding of the influence of several natural problem parameters like path length $k-1$ or the upper bound ℓ on the distance values between consecutive snapshots, but we also want to find out where (and why) the problem variants are potentially different from each other; in particular, this means confronting the similarity (a.k.a. as classical multistage) view with the dissimilarity view.

Our contributions. We introduce four natural variants of the Multistage s-t Path problem by employing four different ways to measure the distance between consecutive solutions. Doing so, seemingly for the first time for multistage models in general, we provide a seemingly first systematic study on the impact on the algorithmic complexity when switching between edge and vertex distances on the one hand, and similarity versus dissimilarity distance measurements on the other hand.

We prove all four problems to be NP-complete, even in the restricted case of only two snapshots, each snapshot being series-parallel and the underlying graph being of maximum degree four. We provide an extensive study on the parameterized complexity landscape of the problems regarding the parameters k (path length), ℓ (maximum path distance between consecutive snapshots), τ (lifetime), n (number of graph vertices), $\nu_\downarrow$ (vertex cover number of the "underlying graph"), and $\Delta_\downarrow$ (maximum vertex degree in the underlying graph); see Figure 1 for an overview. The results of our parameterized complexity analysis reveal a clear distinction between similarity and dissimilarity. When parameterized by the maximum number k of vertices in each s-t path, while E△E-MstP and V△V-MstP are W[1]-hard, E∩E-MstP and V∩V-MstP are fixed-parameter tractable. To this end, we develop one of the first uses of the technique of representative sets [32, 19] in the context of temporal graphs. In addition, we show that, under standard complexity-theoretic assumptions, the similarity problem V△V-MstP parameterized by the number of vertices has no polynomial kernel, while the dissimilarity problem V∩V-MstP has one.

Related work. Our studies are within algorithmic temporal graph theory and, more specifically, contribute and extend a series of studies on the multistage model. Notably, all previous studies (on various basic computational problems) within the multistage framework adhere to the "similarity view"; we extend this by introducing also a "dissimilarity view".

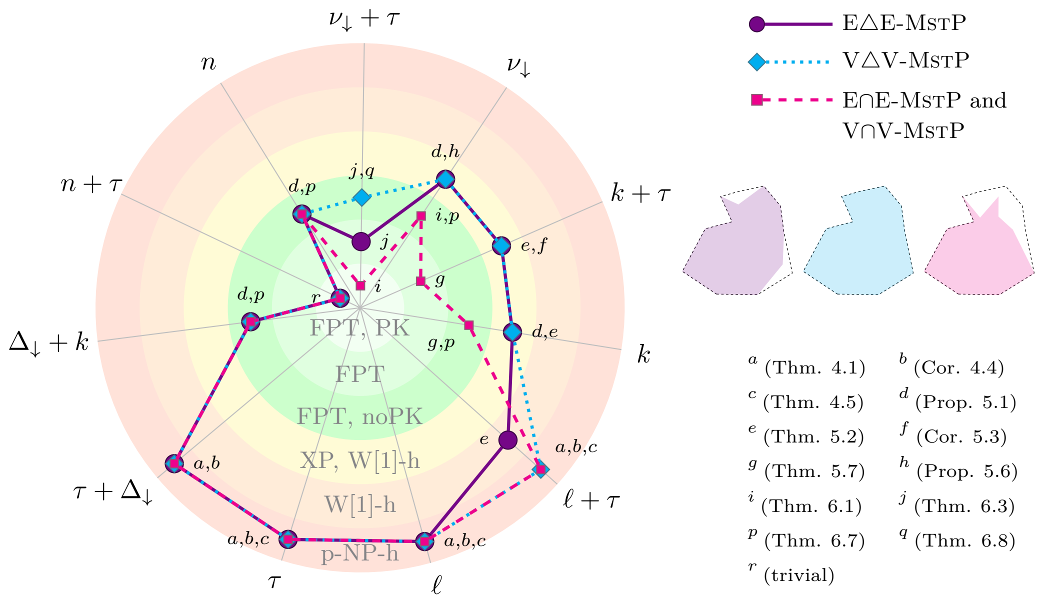

Figure 1 Overview of our results. “p-NP-h.”, “W[1]-h.”, “FPT”, “PK”, and “noPK” respectively abbreviate para-NP-hard, W[1]-hard, fixed-parameter tractable, polynomial kernel, and “no polynomial kernel unless NP $\subseteq$ coNP / poly”. Note that $\ell \le 2k$ and $k \le 2\nu_\downarrow + 1$.

To the best of our knowledge, the *multistage* model (which is a temporal model not necessarily only applying to graph problems) first appeared in 2014 in works of Eisenstat et al. [10] and Gupta et al. [22]. In a nutshell, the model considers a sequence $(I_1, \dots, I_\tau)$ of instances of some problem P as input, and it asks for a “robust” sequence of solutions to the instances in the sense that any two consecutive solutions are similar. Several classical problems have been studied in the multistage model, both from an approximate [1, 4, 2, 3] and from a parameterized [18, 24, 6] algorithmics point of view. While E△E-MstP and V△V-MstP adhere to the original multistage model, our two problems E∩E-MstP and V∩V-MstP can be seen as a novel and natural variation of the multistage model by replacing the goal of consecutive similarity with consecutive dissimilarity.

Several basic temporal graph problems are closely related to the task of finding a (short) temporal s-t path (finding an s-t path over time, that is, an s-t path where the edges have non-decreasing time stamps along the path) [25, 7, 14, 13, 12, 11, 28, 35, 36, 16]. While these problems typically deal with temporal s-t paths that may span over several snapshots of the temporal graph, in our multistage-inspired framework we aim for finding an s-t path in *each* snapshot.

We mention in passing that there is also somewhat related work on short paths in multiplex networks (also known as multilayer or multimodal networks) [20]. The main difference to our scenario is that the temporal aspect imposes an ordering of the layers whereas the multiplex view does not; in addition, Ghariblou et al. [20] perform a multiobjective optimization, being particularly interested in Pareto efficiency.

Due to the lack of space, several details and proofs (marked by ★) are deferred to a full version of this paper.

2 Preliminaries

We denote by $\mathbb{N}$ and $\mathbb{N}_0$ the natural numbers excluding and including 0, respectively. By $\log(\cdot)$ we denote the logarithm to base two. We use basic notation from graph theory and parameterized algorithmics.

Graph theory. An undirected graph $G = (V, E)$ is a tuple consisting of a set V of vertices and a set $E \subseteq \{\{v, w\} \mid v, w \in V, v \neq w\}$ of edges. For a graph G, we also denote by $V(G)$ and $E(G)$ the vertex and edge set of G, respectively. For a vertex set $W \subseteq V$, the induced subgraph $G[W]$ is defined as the graph $(W, \{\{v, w\} \in E \mid v, w \in W\})$. A path $P = (V, E)$ is a graph with a set $V = \{v_1, \ldots, v_k\}$ of distinct vertices and edge set $E = \{\{v_i, v_{i+1}\} \mid 1 \leq i < k\}$ (we often represent path P by the tuple $(v_1, v_2, \ldots, v_k)$); we say that P is a v_1-v_k path. The length of a path is its number of edges. For two vertices $s, t \in V(G)$, an s-t separator $S \subseteq V(G) \setminus \{s, t\}$ is a set of vertices such that there is no s-t path in $G - S$, where $G - S = G[V \setminus S]$. We denote by $N_G(v) = \{w \in V \mid \{w, v\} \in E\}$ the neighborhood of a vertex v in G, and by $\deg(v) = |N_G(v)|$ the degree of v in G. Moreover, we denote by Δ (or $\Delta(G)$) the maximum vertex-degree of G, that is, $\Delta(G) = \max_{v \in V} \deg(v)$. A *vertex cover* of G is a set W of vertices such that $G - W$ contains no edge; we denote by ν (or $\nu(G)$) the smallest size of a vertex cover in G. A graph with distinct terminal vertices s, t is series-parallel if it can be turned into a single edge by a sequence of contractions of degree-two vertices except s and t while removing any parallel edge that appears [9].

Temporal graph theory. A temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ consists of a set V of vertices and lifetime τ many edge sets $E_1, E_2, \ldots, E_\tau$ over V. We also denote by $\tau(\mathcal{G})$ the lifetime of $\mathcal{G}$. The size of $\mathcal{G}$ is $|\mathcal{G}| := |V| + \sum_{i=1}^{\tau} |E_i|$. The static graph (V, E_i) is called the i-th snapshot. The *underlying graph* $\mathcal{G}_\downarrow$ of $\mathcal{G}$ is the static graph $(V, E_1 \cup \cdots \cup E_\tau)$. The underlying vertex cover number $\nu_\downarrow$ is $\nu(\mathcal{G}_\downarrow)$. The underlying maximum degree $\Delta_\downarrow$ is $\Delta(\mathcal{G}_\downarrow)$.

Parameterized complexity. Let Σ denote a finite alphabet. A parameterized problem $L \subseteq \{(x, k) \in \Sigma^* \times \mathbb{N}_0\}$ is a subset of all instances (x, k) from $\Sigma^* \times \mathbb{N}_0$, where k denotes the parameter. A parameterized problem L is (i) fixed-parameter tractable if there is an algorithm that decides every instance (x, k) for L in $f(k) \cdot |x|^{O(1)}$ time, (ii) contained in the class XP if there is an algorithm that decides every instance (x, k) for L in $|x|^{f(k)}$ time, and (iii) para-NP-hard if the problem for some constant value of the parameter is NP-hard, where f is some computable function only depending on the parameter. For two parameterized problems L, L', an instance $(x, k) \in \Sigma^* \times \mathbb{N}_0$ of L is equivalent to an instance $(x', k') \in \Sigma^* \times \mathbb{N}_0$ for L' if $(x, k) \in L \iff (x', k') \in L'$. A problem L is hard for the class W[1] (W[1]-hard) if for every problem $L' \in$ W[1] there is an algorithm that maps any instance (x, k) in $f(k) \cdot |x|^{O(1)}$ time to an equivalent instance (x', k') with $k' = g(k)$ for some computable functions f, g. It holds true that FPT $\subseteq$ W[1] $\subseteq$ XP, where FPT denotes the class of all fixed-parameter tractable parameterized problems. It is believed that FPT $\neq$ W[1], and that hence no W[1]-hard problem is fixed-parameter tractable. A problem kernelization for a parameterized problem L is a polynomial-time algorithm that maps any instance (x, k) of L to an equivalent instance (x', k') of L (the kernel) such that $|x'| + k \leq f(k)$ for some computable function f; If f is a polynomial, we say that the problem kernelization (and kernel) is polynomial. It is well-known that a decidable parameterized problem is fixed-parameter tractable if and only if it admits a problem kernelization.

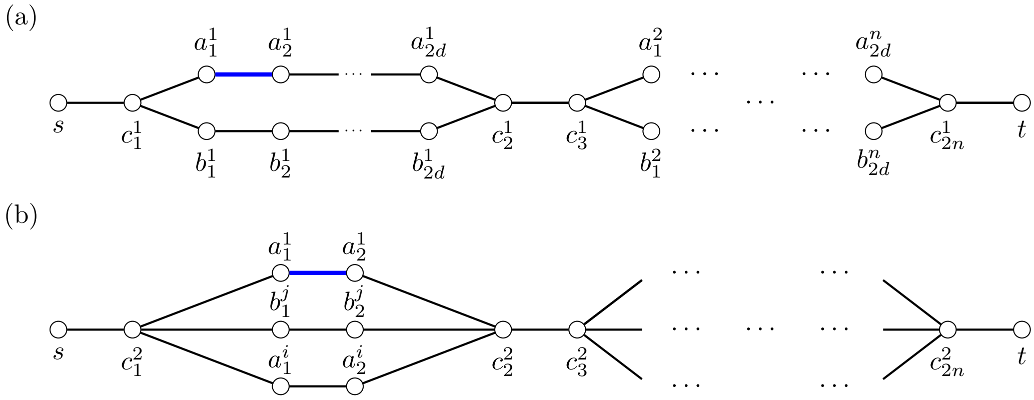

Figure 2 Illustration to Construction 1 with (a) illustrating the first snapshot and (b) illustrating the second snapshot, exemplified for clause $C_1 = (x_1 \vee \overline{x_j} \vee x_i)$. The edge $\{a_1^1, a_2^1\}$ is highlighted in both (a) and (b).

3 Relation between distance measures: from edges to vertices

We show that there are polynomial-time algorithms that, given an instance of E△E-MStP or of E∩E-MStP, construct an equivalent instance of the respective vertex-counterpart.

▶ **Proposition 3.1** (★). *There is an algorithm that, on every input* $(\mathcal{G}, s, t, k, \ell)$ *to* E△E-MStP, *computes in* $\mathcal{O}(|\mathcal{G}| \cdot \ell)$ *time an equivalent instance* $(\mathcal{G}', s, t, k', \ell')$ *of* V△V-MStP *such that* $k' \in O(k \cdot \ell)$, $\ell' \in O(\ell^2)$, $\Delta(\mathcal{G}_\downarrow) = \Delta(\mathcal{G}'_\downarrow)$, *and* $\tau(\mathcal{G}) = \tau(\mathcal{G}')$.

▶ **Proposition 3.2** (★). *There is an algorithm that, on every input* $(\mathcal{G}, s, t, k, \ell)$ *to* E∩E-MStP, *computes in* $\mathcal{O}(|\mathcal{G}|)$ *time an equivalent instance* $(\mathcal{G}', s, t, k', \ell')$ *of* V∩V-MStP *such that* $k' = 2k - 1$, $\ell' = \ell$, $\Delta(\mathcal{G}_\downarrow) = \max\{\Delta(\mathcal{G}'_\downarrow), 4\}$, *and* $\tau(\mathcal{G}) = \tau(\mathcal{G}')$.

Due to Propositions 3.1 and 3.2, often we just may prove lower bounds for E△E-MStP and E∩E-MStP, and upper bounds for V△V-MStP and V∩V-MStP, and transfer the results to their respective counterparts.

4 NP-hardness even for two snapshots of maximum degree four

In this section, we prove that all four problems are NP-hard even for only two snapshots and the maximum underlying vertex-degree being four.

▶ **Theorem 4.1.** E△E-MStP *and* E∩E-MStP, *the latter with* $\ell = 0$, *are* NP*-hard even if* $\mathcal{G}$ *consists of two snapshots both being series-parallel graphs and* $\Delta(\mathcal{G}_\downarrow) = 4$.

We give two polynomial-time many-one reductions from the NP-complete 3-SAT, each employing the following.

▶ **Construction 1.** Let $(X = \{x_1, \ldots, x_n\}, \mathcal{C} = (C_1, \ldots, C_n))$ be an instance of 3-SAT where w.l.o.g. the number n of variables equals the number of clauses, and let $d \geq 2$ denote the most frequent appearance (along the clause sequence) of any literal of some variable in X. We construct a temporal graph $\mathcal{G} = (V, E_1, E_2)$ as follows (see Figure 2 for an illustration).

Let $V := \{s, t\} \cup \{c_1^i, \ldots, c_{2n}^i \mid i \in \{1, 2\}\} \cup \{a_1^i, \ldots, a_{2d}^i \mid x_i \in X\} \cup \{b_1^i, \ldots, b_{2d}^i \mid x_i \in X\}$. Let $E_{i,a} := \bigcup_{1 \leq j < 2d} \{\{a_j^i, a_{j+1}^i\}\}$ and $E_{i,b} := \bigcup_{1 \leq j < 2d} \{\{b_j^i, b_{j+1}^i\}\}$. Then E_1 contains the edge $\{s, c_1^1\}$, the edge set $\bigcup_{1 \leq i \leq n} \{\{c_{2i-1}^1, a_1^i\}, \{c_{2i-1}^1, b_1^i\}\}$, the edge set $\bigcup_{1 \leq i \leq n} \{\{c_{2i}^1, a_{2d}^i\},$

$\{c^1_{2i}, b^i_{2d}\}\}$, the edge $\{t, c^1_{2n}\}$, the edge set $\bigcup_{1\leq i<n}\{\{c^1_{2i}, c^1_{2i+1}\}\}$, and the edge sets $\bigcup_{1\leq i\leq n} E_{i,a}$ and $\bigcup_{1\leq i\leq n} E_{i,b}$. For E_2, for each clause $C_q \in \mathcal{C}$ we define the vertex set V_{C_q} and edge set E_{C_q} as follows. If C_q contains the j-th appearance of the positive literal x_i, then add a^i_{2j-1}, a^i_{2j} to V_{C_q} and the edges $\{a^i_{2j-1}, a^i_{2j}\}, \{c^2_{2q-1}, a^i_{2j-1}\}, \{c^2_{2q}, a^i_{2j}\}$ to E_{C_q}. If C_q contains the j-th appearance of the negative literal $\overline{x_i}$, then add b^i_{2j-1}, b^i_{2j} to V_{C_q} and the edges $\{b^i_{2j-1}, b^i_{2j}\}, \{c^2_{2q-1}, b^i_{2j-1}\}, \{c^2_{2q}, b^i_{2j}\}$ to E_{C_q}. Then, E_2 contains the edges $\{s, c^2_1\}$, $\{t, c^2_{2n}\}$, the edge set $\bigcup_{1\leq i<n}\{\{c^2_{2i}, c^2_{2i+1}\}\}$, and E_{C_q} for each $q \in \{1, \ldots, n\}$. This finishes the construction of $\mathcal{G}$. It is not difficult to see that (V, E_1) and (V, E_2) are series-parallel graphs. Moreover, $\Delta(\mathcal{G}_\downarrow) = 4$. Set $k = 2 + 2n + 2d \cdot n$. ⋄

Intuitively, if an instance constructed using Construction 1 is a yes-instance for E△E-MstP, then the s-t path in the first snapshot selects setting variables to true or false such that the s-t path in the second snapshot can pass a literal for each clause. It follows that Construction 1 is a polynomial-time many-one reduction.

The next two propositions, Propositions 4.2 and 4.3, together prove Theorem 4.1.

▶ **Proposition 4.2 (★).** E△E-MstP *is* NP*-hard even if* $\mathcal{G}$ *consists of two snapshots both being series-parallel graphs and* $\Delta(\mathcal{G}_\downarrow) = 4$.

Interestingly, Construction 1 also gives a polynomial-time many-one reduction for E∩E-MstP. Here the intuition is opposite: the first snapshot path selects setting the variables to the *complement* of a satisfying assignment such that the second snapshot path can pass the "clause gadgets" without passing any edge contained in the first snapshot path.

▶ **Proposition 4.3 (★).** E∩E-MstP *is* NP*-hard even if* $\mathcal{G}$ *consists of two snapshots both being series-parallel graphs,* $\Delta(\mathcal{G}_\downarrow) = 4$*, and* $\ell = 0$.

The theorem follows directly from Propositions 4.2 and 4.3. Due to Propositions 3.1 and 3.2, we get the following from Theorem 4.1.

▶ **Corollary 4.4.** V△V-MstP *and* V∩V-MstP *with* $\ell = 0$ *are* NP*-hard even if* $\tau = 2$ *and* $\Delta(\mathcal{G}_\downarrow) = 4$.

We proved E∩E-MstP and V∩V-MstP to remain NP-hard even if $\ell = 0$ and $\tau = 2$. This leads us to ask whether for a constant value of $\ell + \tau$, E△E-MstP or V△V-MstP remain NP-hard. In fact, we prove this to be true for the vertex-variant.

▶ **Theorem 4.5 (★).** V△V-MstP *is* NP*-hard and admits no* $2^{o(k)} \cdot (|\mathcal{G}|)^{O(1)}$*-time algorithm unless the Exponential Time Hypothesis fails, even if* $\ell = 0$ *and* $\tau = 2$.

It remains open whether E△E-MstP is contained in XP regarding $\ell + \tau$.

5 The role of the parameter path length

In this section, we focus on the parameter k, the maximum number of vertices in any s-t path. It is not hard to see that all variants allow for an XP-algorithm when parameterized by the maximum number k of vertices in each path.

▶ **Proposition 5.1 (★).** V△V-MstP *and* V∩V-MstP*, and hence* E△E-MstP *and* E∩E-MstP*, are solvable in* $\Delta_{\max}^{O(k)} \cdot |\mathcal{G}|^{O(1)}$ *time, where* $\Delta_{\max} = \max_{i\in\{1,\ldots,\tau\}} \Delta((V, E_i))$.

We will prove that the parameterization with k distinguishes similarity from dissimilarity: While E△E-MstP and V△V-MstP are W[1]-hard regarding k (even regarding $k + \tau$), each of E∩E-MstP and V∩V-MstP turn out to be fixed-parameter tractable.

5.1 W[1]-hardness for the similarity variant regarding $k + \tau$ and $\nu_\downarrow$

We prove that E△E-MstP is W[1]-hard regarding $k + \tau$ even if the upper bound ℓ on the sizes of consecutive symmetric differences is constant. Due to Proposition 3.1, we then obtain the same result for V△V-MstP. The proof is by a parameterized reduction from the W[1]-complete Multicolored Clique problem.

▶ **Theorem 5.2 (★).** *Even if $\ell = 4$ and each snapshot is bipartite,* E△E-MstP *is* NP*-hard and* W[1]*-hard when parameterized by $k + \tau$.*

Due to Proposition 3.1, we get the following.

▶ **Corollary 5.3.** V△V-MstP *is* W[1]*-hard when parameterized by $k+\tau$, even if ℓ is constant.*

By Proposition 5.1 and since $k \leq n$, we know that E△E-MstP and V△V-MstP are fixed-parameter tractable regarding the number n of graph vertices. Regarding the parameter number k of path vertices (and even for $k + \tau$), by Theorem 5.2 and Corollary 5.3 we know that both problems are in XP yet W[1]-hard. Since we can assume $k \leq 2\nu_\downarrow + 1$ (recall that $\nu_\downarrow$ is the vertex cover number of the underlying graph) in every instance and thus naturally $\nu_\downarrow \leq n$, we can settle the parameterized complexity regarding $\nu_\downarrow$:

▶ **Theorem 5.4.** *When parameterized by $\nu_\downarrow$,* V△V-MstP *with $\ell = 1$ and* E△E-MstP *are* W[1]*-hard.*

We prove each statement of Theorem 5.4 separately, both proofs rely on parameterized reductions from Multicolored Clique.

▶ **Proposition 5.5 (★).** E△E-MstP *when parameterized by $\nu_\downarrow$ is* W[1]*-hard.*

For V△V-MstP, we have an even stronger result: the problem is W[1]-hard regarding $\nu_\downarrow$ even if the size of any symmetric difference of the vertex sets of consecutive paths is at most one. The proof is, however, similar to the proof of Proposition 5.5.

▶ **Proposition 5.6 (★).** V△V-MstP *when parameterized by $\nu_\downarrow$ is* W[1]*-hard, even if $\ell = 1$.*

We will see in the next section that a similar result for E∩E-MstP or V∩V-MstP is unlikely.

5.2 Fixed-parameter tractability for dissimilarity variant regarding k

In stark contrast to Theorem 5.2 and Corollary 5.3, we show in this section that V∩V-MstP and E∩E-MstP can be solved in linear time for constant path lengths; put differently, they are fixed-parameter tractable when parameterized by path length $k - 1$.

▶ **Theorem 5.7.** V∩V-MstP *and* E∩E-MstP *can be solved in $2^{O(k)} \cdot |\mathcal{G}|$ time.*

We defer the proof of Theorem 5.7 towards the end of this section and, moreover, only describe the algorithm for V∩V-MstP. In a nutshell, the algorithm behind Theorem 5.7 computes for each snapshot *sufficiently many* s-t paths such that no matter which vertices are used in the snapshots beforehand and afterwards, one of these s-t paths has a small intersection with these vertices. To this end, we introduce q-robust sets[2] of s-t paths.

[2] In a nutshell, q-robust sets are q-representative families [32], just explicitly coined to s-t paths of length at most k. This notion shall avoid confusion with the later defined q-representatives of independent sets.

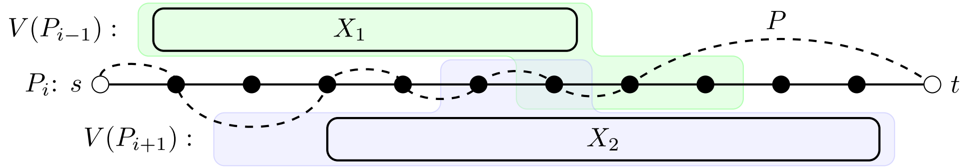

Figure 3 Illustration of Case 1 in the proof of Lemma 5.9, where $|V(P_{i+1}) \setminus V(P_i)| > k - \ell$.

▶ **Definition 5.8.** *Let $G = (V, E)$ be a graph, $s, t \in V$ two distinct vertices, $\mathcal{F}$ be a set of s-t paths of length at most $k - 1$, and $q \in \mathbb{N}_0$. We call $\mathcal{F}$ q-*robust *if for each set $X \subseteq (V(G) \setminus \{s, t\})$ of size at most q the following holds: if there is an s-t path in $G - X$ of length at most $k - 1$, then there is an s-t path $P \in \mathcal{F}$ which is an s-t path in $G - X$.*

To find a solution, it is sufficient to have a $2(k - \ell)$-robust set of s-t paths of length at most $k - 1$ for each snapshot of the temporal graph:

▶ **Lemma 5.9.** *Let $I = (\mathcal{G} = (V, (E_i)_{i=1}^{\tau}), s, t, k, \ell)$ be an instance of* V∩V-MstP *and $\mathcal{F}_i$ be a $2(k-\ell)$-robust set of s-t paths of length at most $k-1$ in $G_i = (V, E_i)$, for all $i \in \{1, \ldots, \tau\}$. Then, I is a* `yes`*-instance if and only if there is a solution $(P_1, \ldots, P_\tau)$ such that $P_i \in \mathcal{F}_i$, for all $i \in \{1, \ldots, \tau\}$.*

Proof. Since the converse is trivially true, we only show that if I is a `yes`-instance, then there is a solution $(P_1, \ldots, P_\tau)$ for I such that for all $i \in \{1, \ldots, \tau\}$ we have $P_i \in \mathcal{F}_i$.

For all $p \in \{1, \ldots, \tau + 1\}$, let $\mathcal{S}_p$ be the set of solutions for I such that for all $j < p$ we have $P_j \in \mathcal{F}_j$. Let $i := \max\{p \in \{1, \ldots, \tau + 1\} \mid \mathcal{S}_p \neq \emptyset\}$. If $i = \tau + 1$, then we are done. Hence, assume towards a contradiction that $i \leq \tau$.

(Case 1): Suppose $1 < i < \tau$. Let $X_1 = V(P_{i-1}) \setminus V(P_i)$ and $X_2 = V(P_{i+1}) \setminus V(P_i)$. If $X \in \{X_1, X_2\}$ is larger than $k - \ell$, then remove arbitrary vertices from X such that $|X| = k - \ell$. Note that $|V(P_{i-1}) \setminus X_1| \leq \ell$ and $|V(P_{i+1}) \setminus X_2| \leq \ell$. Observe that P_i is an s-t path of length at most $k - 1$ in $G_i - (X_1 \cup X_2)$. Since $\mathcal{F}_i$ is $2(k - \ell)$-robust, there is an s-t path $P \in \mathcal{F}_i$ of length at most $k - 1$ in $G_i - (X_1 \cup X_2)$, see Figure 3 for an illustration. Hence, $|V(P) \cap V(P_{i-1})| \leq |V(P) \cap (V(P_{i-1}) \setminus X_1)| \leq \ell$ and $|V(P) \cap V(P_{i+1})| \leq |V(P) \cap (V(P_{i+1}) \setminus X_2)| \leq \ell$. Thus, $S = (P_1, \ldots, P_{i-1}, P, P_{i+1}, \ldots, P_\tau)$ is a solution for I. This contradicts i being maximal.

(Case 2): If $i = 1$ ($i = \tau$), then we set $X_1 = \emptyset$ ($X_2 = \emptyset$) and conclude analogously to Case 1 that i is not maximized. ◀

The main tool of our algorithm is a fast ("linear-time FPT") computation of small sets of s-t paths of length at most $k-1$ which are q-robust. We believe that such a use of representative families may become a general algorithmic tool being potentially helpful for other multistage problems. Formally, we show the following.

▶ **Lemma 5.10** (★)**.** *Let $G = (V, E)$ be a graph with two distinct vertices $s, t \in V$, and $k, q \in \mathbb{N}_0$. We can compute, in $2^{O(k+q)} \cdot |E|$ time, a q-robust set $\mathcal{F}$ of s-t paths of length at most $k - 1$ such that $|\mathcal{F}| \leq 2^{q+k}$.*

In order to prove Lemma 5.10, we extend the "representative-family-based" algorithm for k-Path of Fomin et al. [19] such that we can find s-t paths avoiding a size-at-most-q set of vertices.Having Lemmata 5.9 and 5.10, we are set to prove Theorem 5.7.

Proof of Theorem 5.7. We only show the proof for V∩V-MstP. The fixed-parameter tractability of E∩E-MstP follows from Proposition 3.2.

Given an instance $I = (\mathcal{G} = (V, (E_i)_{i=1}^{\tau}), s, t, k, \ell)$ of V∩V-MstP, we first check whether there is an empty E_i. If this is the case, then I is a no-instance. Afterwards, we can assume that $\tau \leq |\mathcal{G}|$. For each $i \in \{1, \ldots, \tau\}$, we compute in $2^{O(k+2(k-\ell))}|E_i| = 2^{O(k)}|E_i|$ time a $2(k-\ell)$-robust set $\mathcal{F}_i$ of s-t paths of length at most $k-1$ in $G_i = (V, E_i)$ such that $|\mathcal{F}_i| \leq 2^{O(k)}$, see Lemma 5.10.

Next, we construct a directed graph $G' = (V', E')$, where beside s, t each path in $\mathcal{F}_i$ has a corresponding vertex, for all $i \in \{1, \ldots, \tau\}$. Formally, that is, $V' := \{s, t\} \cup \bigcup_{i=1}^{\tau} \mathcal{F}_i$, and $E' := \{(P, P') \mid P \in \mathcal{F}_i, P' \in \mathcal{F}_{i+1}, |V(P) \cap V(P')| \leq \ell, \text{ for some } i \in \{1, \ldots, \tau-1\}\} \cup \{(s, P) \mid P \in \mathcal{F}_1\} \cup \{(P, t) \mid P \in \mathcal{F}_\tau\}$. Observe that $|V'| + |E'| \leq 2^{O(k)} \cdot \tau$. We note that I is a yes-instance if and only if there is an s-t path in G'. Since $\sum_{i=1}^{\tau} |E_i| \leq |\mathcal{G}|$, this yields an overall running time of $2^{O(k)} \cdot \max\{\tau, |\mathcal{G}|\} = 2^{O(k)} \cdot |\mathcal{G}|$.

It remains to show that I is a yes-instance if and only if there is an s-t path in G'. We only show that if I is a yes-instance, then there is an s-t path in G' since the converse is easy to verify from the definition of G'. Let I be a yes-instance. Then, by Lemma 5.9, there is a solution $(P_1, \ldots, P_\tau)$ such that $P_i \in \mathcal{F}_i$, for all $i \in \{1, \ldots, \tau\}$. For each $i \in \{1, \ldots, \tau-1\}$, we have that $|V(P_i) \cap V(P_{i+1})| \leq \ell$. It follows that G' has an edge from the vertex corresponding to P_i to the vertex corresponding to P_{i+1}. Hence, there is an s-t path in G' because s is adjacent to all vertices corresponding to a path in $\mathcal{F}_1$ and each vertex corresponding to a path in $\mathcal{F}_\tau$ is adjacent to t. ◀

6 Looking through the lens of efficient data reduction

In this section, we study whether (polynomial) problem kernels for our four multistage s-t path problems exist. We start from the simple observation that every problem trivially admits a problem kernel of size polynomial in $n + \tau$. When strengthening n to $\nu_\downarrow$, that is, when parameterizing by $\nu_\downarrow + \tau$, where $\nu_\downarrow$ denotes the vertex cover number of the underlying graph, for E∩E-MstP and V∩V-MstP we prove a polynomial-size problem kernel (Section 6.1) and for E△E-MstP and V△V-MstP we prove a single-exponential-size problem kernel (Section 6.2). We prove that, unless NP ⊆ coNP / poly, the latter cannot be improved to polynomial size for V△V-MstP and that when parameterized by n (i.e., dropping τ from $n + \tau$) none of the four problems admits a polynomial kernel (Section 6.3).

6.1 Polynomial kernel for the dissimilarity variant regarding $\nu_\downarrow + \tau$

In this section, we prove V∩V-MstP and E∩E-MstP to admit problem kernels of polynomial size in $\nu_\downarrow + \tau$.

▶ **Theorem 6.1.** *Each of* V∩V-MstP *and* E∩E-MstP *admits a problem kernel with at most* $\tau \cdot (2\nu_\downarrow + 2 + \binom{2\nu_\downarrow}{2}(3k-3)) \in O(\tau \nu_\downarrow^3)$ *vertices and* τ *snapshots.*

The kernelization behind Theorem 6.1 basically relies on the following data reduction rule.

▶ **Reduction Rule 1.** Let $I = (\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau), s, t, k, \ell)$ be an instance of V∩V-MstP or E∩E-MstP with underlying graph $\mathcal{G}_\downarrow$.

1. Compute a vertex cover V' of $\mathcal{G}_\downarrow$ of size at most $2\nu_\downarrow$.
2. For each pair of distinct vertices $v, w \in V'$ and each $i \in \{1, \ldots, \tau\}$, in $N_{vw}^i := (N_{(V,E_i)}(v) \cap N_{(V,E_i)}(w)) \setminus V'$ mark $\min\{3k-3, |N_{vw}^i|\}$ vertices.

3. Construct a set V'' containing $\{s,t\} \cup V'$ and all marked vertices, and then construct the temporal graph $\mathcal{G}' = (V'', E'_1, \dots, E'_\tau)$, where $E'_i = \{\{v,w\} \in E_i \mid v,w \in V''\}$, for all $i \in \{1,\dots,\tau\}$.
4. Output the instance $O = (\mathcal{G}', s, t, k, \ell)$.

First, we prove that we can efficiently execute Reduction Rule 1.

▶ **Lemma 6.2 (★).** *Reduction Rule 1 is correct and can be executed in $O(n \cdot \nu_\downarrow^2)$ time.*

Proof of Theorem 6.1. Given an instance $I = (\mathcal{G} = (V, E_1, E_2, \dots, E_\tau), s, t, k, \ell)$, we apply Reduction Rule 1 in polynomial time to obtain the instance $O = (\mathcal{G}', s, t, k, \ell)$ being equivalent to I (Lemma 6.2), containing τ snapshots and at most $\tau \cdot (2\nu_\downarrow + 2 + \binom{2\nu_\downarrow}{2})(3k-3))$ vertices. ◀

6.2 Single-exponential kernel for the similarity variant regarding $\nu_\downarrow + \tau$

We prove that E△E-MstP and V△V-MstP admit problem kernels of single-exponential size in $\nu_\downarrow + \tau$, proving containment in FPT. As we will see later, unless NP ⊆ coNP / poly this result for V△V-MstP cannot be improved to size polynomial in $\nu_\downarrow + \tau$.

▶ **Theorem 6.3.** *Each of* E△E-MstP *and* V△V-MstP *admits a problem kernel with at most $2\nu_\downarrow + 4^{\nu_\downarrow \tau}(2\nu_\downarrow + 1)$ vertices and τ snapshots.*

To prove Theorem 6.3, we lift the well-known graph-theoretic notion of (false) twins to temporal graphs as follows.

▶ **Definition 6.4.** *Two vertices v, w in a temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_\tau)$ are called* (false) temporal twins *if $N_{(V,E_i)}(v) = N_{(V,E_i)}(w)$ for every $i \in \{1,\dots,\tau\}$.*

Note that Definition 6.4 implies an equivalence relation $\sim$ on the vertex set V, where $v \sim w$ if and only if they are temporal twins, and, hence, a partition of the vertex set into classes of temporal twins. Moreover, every pair of vertices in the same temporal twin class is non-adjacent. We show that such a partition is efficiently computable.

▶ **Lemma 6.5.** *For a temporal graph $\mathcal{G} = (V, E_1, E_2, \dots, E_\tau)$, a partition $V = (V_1, \dots, V_p)$ of V into temporal twin classes is computable in $O(\tau \cdot |V|^2)$ time.*

Proof. Firstly, we compute all (false) twin classes in the first snapshot (V, E_1) in time linear in $|V| + |E_1|$. Next, for each vertex $v \in V$, check for each w with $v \sim w$ whether w is a false twin in each snapshot $(V, E_2), \dots, (V, E_\tau)$, and adjust $\sim$ accordingly. ◀

In a nutshell, given a vertex cover X of our underlying graph, we aim for having few (i.e., upper-bounded by some single-exponential function in $\nu_\downarrow + \tau$) temporal twin classes in the independent set $Y = V \setminus X$, where each temporal twin class in turn contains only few vertices. By definition we have only few temporal twin classes.

▶ **Observation 1.** *Let $\mathcal{G} = (V, E_1, E_2, \dots, E_\tau)$ be a temporal graph with partition $V = (X, Y)$ of V such that Y is an independent set in each snapshot. Then the size of every partition of Y into temporal twin classes is at most $2^{|X|\tau}$.*

Proof. There are at most $2^{|X|}$ different neighborhoods for any vertex in Y per snapshot. As there are τ snapshots, there are at most $(2^{|X|})^\tau$ many temporal twin classes. ◀

We next aim for shrinking temporal twin classes. Recall that each temporal twin class forms an independent set, and hence every s-t path must "alternate" between the class and its neighboring vertices. Thus, for every temporal twin class disjoint from $\{s,t\}$ it holds true that any s-t path contains less vertices from the temporal twin class than the number of vertices neighboring it. In fact, temporal twin classes that are large compared to their neighborhood size can be shrunk.

▶ **Reduction Rule 2.** Let S be a temporal twin class with $|S \setminus \{s,t\}| \geq \max_{1 \leq i \leq \tau} |N_{(V,E_i)}(S)|$. Then delete a vertex $v \in S \setminus \{s,t\}$.

▶ **Lemma 6.6 (★).** *Reduction Rule 2 is correct and exhaustively applicable in $O(\tau \cdot |V|^3)$ time.*

Proof of Theorem 6.3. First, in $\mathcal{G}_\downarrow$ compute (via a maximal matching) a vertex cover X of size at most $2\nu_\downarrow$ in linear time. Let $V = (X, Y)$, where $Y = V \setminus X$ is an independent set. Next, compute all temporal twin classes of Y in polynomial time (Lemma 6.5). Apply Reduction Rule 2 exhaustively on every temporal twin class. Due to Lemma 6.6, this returns an equivalent instance in polynomial time where every temporal twin class contains at most $|X|+1$ vertices. Due to Observation 1, there are at most $2^{|X|\tau}$ many temporal twin classes. In total, the obtained temporal graph contains at most $|X| + 2^{|X|\tau}(|X|+1)$ vertices and τ snapshots. ◀

6.3 Lower bounds on kernelization regarding n and $\nu_\downarrow + \tau$

We know that relaxing n to $\nu_\downarrow$ in $n + \tau$ allows for polynomial and single-exponential kernelization for dissimilarity and similarity, respectively. We know that dropping n is not possible (Proposition 5.6). In this section, we prove that, unless NP $\subseteq$ coNP / poly, dropping τ is not possible.

▶ **Theorem 6.7 (★).** *Unless* NP $\subseteq$ coNP / poly, *none of* E△E-MstP, V△V-MstP, E∩E-MstP, *and* V∩V-MstP *admits a problem kernel of size polynomial in n.*

We prove that, unless NP $\subseteq$ coNP / poly, improving the single-exponential kernel for V△V-MstP regarding $\nu_\downarrow + \tau$ to polynomial size is not possible.

▶ **Theorem 6.8 (★).** *Unless* NP $\subseteq$ coNP / poly, V△V-MstP *admits no problem kernel of size polynomial in $\nu_\downarrow + \tau$.*

To prove Theorem 6.8, we OR-cross-compose [5] from the following NP-complete [33] problem.

Positive 1-in-3 SAT
Input: A set X of variables and a set $\mathcal{C}$ of clauses each containing three positive literals over X.
Question: Is there $X' \subseteq X$ such that setting exactly the variables in X' to true results in each clause having exactly one variable set to true?

We call two instances $(X, \mathcal{C}), (X', \mathcal{C}')$ of Positive 1-in-3 SAT $\mathcal{R}$-equivalent if $|X| = |X'|$ and $|\mathcal{C}| = |\mathcal{C}'|$. Note that $\mathcal{R}$ defines a polynomial equivalence relation [5]. In particular, we show the following.

▶ **Proposition 6.9 (★).** *There is an algorithm that given p many $\mathcal{R}$-equivalent instances $I_1 = (X_1, \mathcal{C}_1), \ldots, I_p = (X_p, \mathcal{C}_p)$ of* Positive 1-in-3 SAT*, where p is a power of two, computes in polynomial time an instance I of* V△V-MstP *such that $k+\tau+\nu_\downarrow \in (\max_{i \in \{1,\ldots,p\}} |X_i| + |\mathcal{C}_i| + \log(p))^{O(1)}$ and I is a* `yes`*-instance if and only if at least one of $I_1, \ldots, I_p$ is a* `yes`*-instance.*

We use the following Construction 2 to show Proposition 6.9, see Figure 4 for an illustration. The basic idea of the construction is that the temporal graph has, among other vertices, a vertex set $D = \bigcup_{q=1}^{p} D^q$, where D^q has one vertex for each variable in the q-th input instance.

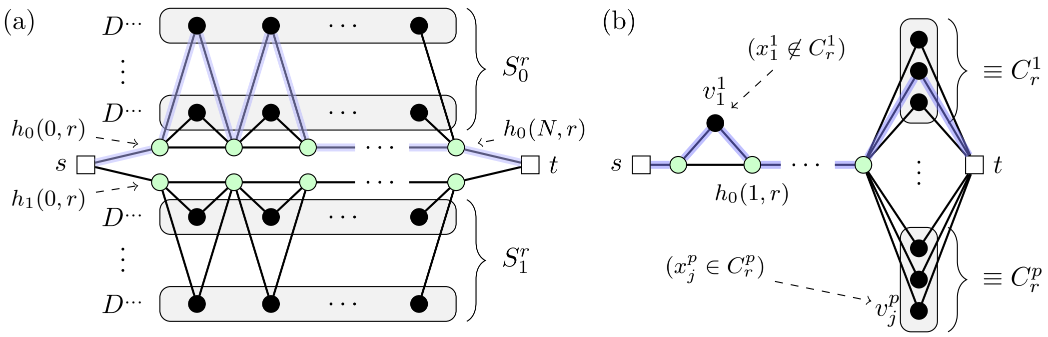

Figure 4 Illustration of Construction 2 with p input instances. (a) shows a snapshot (V, E_r) with $r \le \log(p)$. (b) shows a snapshot $(V, E_{\log(p)+r})$ for the r-th clause of each input instance. Observe that the green (bright) vertices (including s, t) form a vertex cover of the underlying graph.

If we use a vertex from D^q in the s-t path, then we set the corresponding variable to true. In the first $\log(p)$ snapshots, we ensure that each s-t path can only use vertices from D which come from the same input instance. The remainder of the snapshots ensures that the clauses are satisfied. Here, the $(\log(p)+r)$-th snapshot ensures that the r-th clause of some input instance is satisfied with exactly one variable (vertex). Since we only use variables from one instance, Proposition 6.9 follows.

▶ **Construction 2.** Let $I_1 = (X_1, \mathcal{C}_1), \dots, I_p = (X_p, \mathcal{C}_p)$ be p, where p is a power of two, $\mathcal{R}$-equivalent instances of Positive 1-in-3 SAT where $N = |X_i|$ and $M = |\mathcal{C}_i|$ for all $i \in \{1, \dots, p\}$. Let $D^q = \{v_i^q \mid i \in \{1, \dots, N\}\}$ for all $q \in \{1, \dots, p\}$, and $D = \bigcup_{q \in \{1,\dots,p\}} D^q$. Let $A = \{a_0^i, a_1^i \mid i \in \{0, \dots, N\}\}$ and $B = \{b_0^i, b_1^i \mid i \in \{0, \dots, N\}\}$. Set $V = \{s, t\} \cup D \cup A \cup B$. Define for each $d \in \{0, 1\}$ the auxiliary function

$$h_d(i, r) := \begin{cases} a_d^i, & r \text{ odd} \\ b_d^i, & r \text{ even.} \end{cases}$$

We next describe the edge sets $E_1, \dots, E_{\log(p)}$ and $E_{\log(p)+1}, \dots, E_{\log(p)+M}$. For edge set E_r with $r \le \log(p)$, let E_r contain the edges $\{s, h_d(0, r)\}, \{t, h_d(N, r)\}$ and the edge set $\bigcup_{1 \le i \le N} \{\{h_d(i-1, r), h_d(i, r)\}\}$ for each $d \in \{0, 1\}$. These sets form two s-t paths in (V, E_r). Finally, let S_0^r be the union of D^q with the r-th bit of the binary encoding of $q-1$ being 0, and S_1^r be the union of D^q with the r-th bit of the binary encoding of $q-1$ being 1. For $v_i^q \in S_0^r$, add the edges $\{h_0(i-1, r), v_i^q\}$ and $\{h_0(i, r), v_i^q\}$. Similarly, for $v_i^q \in S_1^r$, add the edges $\{h_1(i-1, r), v_i^q\}$ and $\{h_1(i, r), v_i^q\}$. For edge set $E_{\log(p)+r}$ with $r \le M$, let $E_{\log(p)+r}$ contain the edge $\{s, h_0(0, r)\}$ and the edge set $\bigcup_{1 \le i \le N} \{\{h_0(i-1, r), h_0(i, r)\}\}$. Consider the clauses $C_r^1, \dots, C_r^p$. For each C_r^q, if $x_i^q \in C_r^q$, then add the edges $\{h_0(N, r), v_i^q\}, \{v_i^q, t\}$, and if $x_i^q \notin C_r^q$, then add the edges $\{h_0(i-1, r), v_i^q\}, \{h_0(i, r), v_i^q\}$. Set $k = 2N+3$ and $\ell = 2(N+1)$. This finishes the construction. ◇

Proposition 6.9 describes an OR-cross-composition from an NP-hard problem to V△V-MstP parameterized by $\nu_\downarrow + \tau$, and hence Theorem 6.8 follows [5]. We leave open whether E△E-MstP allows for a problem kernel of size polynomial in $\nu_\downarrow + \tau$.

7 Conclusion

On the one extreme, our hardness results exploit that a temporal graph can change dramatically from one time step to another. On the other extreme, the NP-hard (and typically parameterized hard) Length-Bounded Disjoint Path problem [21] easily reduces to all four MstP variants with each snapshot having the same edge set. This leads to the natural question for further islands of computational tractability between these two extremes. Moreover, for the similarity case, we leave open whether working with edge distances decisively differs from working with vertex distances.

The models we introduced (and future, more refined models based upon these) may find several applications as they naturally capture time-dependent route-querying tasks. Besides resolving questions we explicitly stated as open throughout the text, future work could address generalizing the "consecutiveness" property by requiring that also short sequences (as in the time-window model of temporal graphs [29, 30]) of consecutive paths are (pairwise) similar or dissimilar. Furthermore, with introducing the "dissimilarity view" we entered new territory in the context of multistage problems; it seems natural to also study it for other problems beyond s-t Path. Finally, to analyze s-t Path in the *global multistage*[3] setting is well-motivated as well [24].

References

1 Evripidis Bampis, Bruno Escoffier, and Alexander Kononov. LP-based algorithms for multistage minimization problems. *CoRR*, abs/1909.10354, 2019. arXiv:1909.10354.

2 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.

3 Evripidis Bampis, Bruno Escoffier, Kevin Schewior, and Alexandre Teiller. Online multistage subset maximization problems. In *Proc. of 27th ESA*, volume 144 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

4 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.

5 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.

6 Robert Bredereck, Till Fluschnik, and Andrzej Kaczmarczyk. Multistage committee election. *CoRR*, abs/2005.02300, 2020. arXiv:2005.02300.

7 Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. In *Proc. of 31st ISAAC*, LIPIcs, pages 30:1–30:18. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2020. doi:10.4230/LIPIcs.ISAAC.2020.30.

8 Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417–1440, 2004.

9 Richard J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.

10 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *Proc. of 41st ICALP*, volume 8572 of *LNCS*, pages 459–470. Springer, 2014.

11 Jessica Enright and Kitty Meeks. Deleting edges to restrict the size of an epidemic: A new application for treewidth. *Algorithmica*, 80(6):1857–1889, 2018.

[3] That is, the total sum over all differences between consecutive paths in the solution is upper-bounded.

12 Jessica Enright, Kitty Meeks, George B. Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. In *Proc. of 44th MFCS*, volume 138 of *LIPIcs*, pages 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

13 Thomas Erlebach, Frank Kammer, Kelin Luo, Andrej Sajenko, and Jakob T. Spooner. Two moves per time step make a difference. In *Proc. of 46th ICALP*, volume 132 of *LIPIcs*, pages 141:1–141:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

14 Thomas Erlebach and Jakob T. Spooner. Faster exploration of degree-bounded temporal graphs. In *Proc. of 43rd MFCS*, volume 117 of *LIPIcs*, pages 36:1–36:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

15 Till Fluschnik, Stefan Kratsch, Rolf Niedermeier, and Manuel Sorge. The parameterized complexity of the minimum shared edges problem. *Journal of Computer and System Sciences*, 106:23–48, 2019.

16 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theor. Comput. Sci.*, 806:197–218, 2020.

17 Till Fluschnik, Marco Morik, and Manuel Sorge. The complexity of routing with collision avoidance. *Journal of Computer and System Sciences*, 102:69–86, 2019.

18 Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. Multistage vertex cover. In *Proc. of 14th IPEC*, volume 148 of *LIPIcs*, pages 14:1–14:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

19 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM*, 63(4):29:1–29:60, 2016.

20 Saeed Ghariblou, Mostafa Salehi, Matteo Magnani, and Mahdi Jalili. Shortest paths in multiplex networks. *Nature Scientific Reports*, 7:2142, 2017.

21 Petr A. Golovach and Dimitrios M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. *Discrete Optimization*, 8(1):72–86, 2011.

22 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.

23 Sepp Hartung and Rolf Niedermeier. Incremental list coloring of graphs, parameterized by conservation. *Theoretical Computer Science*, 494:86–98, 2013.

24 Klaus Heeger, Anne-Sophie Himmel, Frank Kammer, Rolf Niedermeier, Malte Renken, and Andrej Sajenko. Multistage problems on a global budget. *CoRR*, abs/1912.04392, 2019. arXiv:1912.04392.

25 Anne-Sophie Himmel, Matthias Bentert, André Nichterlein, and Rolf Niedermeier. Efficient computation of optimal temporal walks under waiting-time constraints. In *Proc. of 8th COMPLEX NETWORKS*, volume 882 of *Studies in Computational Intelligence*, pages 494–506. Springer, 2019.

26 Petter Holme and Jari Saramäki (eds.). *Temporal Networks.* Springer, 2013.

27 Petter Holme and Jari Saramäki (eds.). *Temporal Network Theory.* Springer, 2019.

28 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

29 Matthieu Latapy, Marco Fiore, and Artur Ziviani. Link streams: Methods and applications. *Computer Networks*, 150:263–265, 2019.

30 Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61:1–61:29, 2018.

31 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.

32 Burkhard Monien. How to find long paths efficiently. In *North-Holland Mathematics Studies*, volume 109, pages 239–254. Elsevier, 1985.

33 Thomas J Schaefer. The complexity of satisfiability problems. In *Proc. 10th STOC*, pages 216–226, 1978.

34 Carsten Schubert. Preserving paths in temporal graphs. Master's thesis, TU Berlin, September 2019. Bachelor thesis. URL: `http://fpt.akt.tu-berlin.de/zschoche/bachelor-carsten-schubert.pdf`.

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

36 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020.