Bipartite Temporal Graphs and the Parameterized Complexity of Multistage 2-Coloring

Till Fluschnik
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

Pascal Kunz
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

Abstract

We consider the algorithmic complexity of recognizing bipartite temporal graphs. Rather than defining these graphs solely by their underlying graph or individual layers, we define a bipartite temporal graph as one in which every layer can be 2-colored in a way that results in few changes between any two consecutive layers. This approach follows the framework of multistage problems that has received a growing amount of attention in recent years. We investigate the complexity of recognizing these graphs. We show that this problem is NP-hard even if there are only two layers or if only one change is allowed between consecutive layers. We consider the parameterized complexity of the problem with respect to several structural graph parameters, which we transfer from the static to the temporal setting in three different ways. Finally, we consider a version of the problem in which we only restrict the total number of changes throughout the lifetime of the graph. We show that this variant is fixed-parameter tractable with respect to the number of changes.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Dynamic graph algorithms

Keywords and phrases structural parameters, NP-hardness, parameterized algorithms, multistage problems

Digital Object Identifier 10.4230/LIPIcs.SAND.2022.16

Related Version *Full Version*: https://arxiv.org/abs/2111.09049

Funding *Till Fluschnik*: Supported by the DFG, project MATE (NI 369/17).
Pascal Kunz: Supported by the DFG Research Training Group 2434 "Facets of Complexity".

1 Introduction

Bipartite graphs form a well-studied class of static graphs. A graph $G = (V, E)$ is bipartite if it admits a proper 2-coloring. A function $f : V \to \{1, 2\}$ is a *proper 2-coloring* of G if for all edges $\{v, w\} \in E$ it holds that $f(v) \neq f(w)$. In this work, we study the question of what a bipartite *temporal* graph is and how fast we can determine whether a temporal graph is bipartite. We approach this question through the prism of the novel program of multistage problems. Thus, we consider the following decision problem:

▶ **Problem 1.** Multistage 2-Coloring (MS2C)

Input: A temporal graph $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ and an integer $d \in \mathbb{N}_0$.
Question: Are there $f_1, \ldots, f_\tau : V \to \{1, 2\}$ such that f_t is a proper 2-coloring for (V, E_t) for every $t \in \{1, \ldots, \tau\}$ and $|\{v \in V \mid f_t(v) \neq f_{t+1}(v)\}| \leq d$ for every $t \in \{1, \ldots, \tau - 1\}$?

In other words, $(\mathcal{G}, d)$ is a `yes`-instance if $\mathcal{G}$ admits a proper 2-coloring of each layer where only d vertices change colors between any two consecutive layers.

1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022).
Editors: James Aspnes and Othon Michail; Article No. 16; pp. 16:1–16:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There have been various approaches to transferring graph classes from static to temporal graphs. If $\mathcal{C}$ is a class of static graphs, then the two most obvious ways of defining a temporal analog to $\mathcal{C}$ are (i) including all temporal graphs whose underlying graph is in $\mathcal{C}$ or (ii) including all temporal graphs that have all of their layers in $\mathcal{C}$ (see, for instance, [15]). Most applied research that has employed a notion of bipartiteness in temporal graphs [1, 24, 34] has defined it using the underlying graph, seeking to model relationships between two different types of entities. This is certainly appropriate as long as the type of an entity is not itself time-varying. Situations where entities can change their types require more sophisticated notions of bipartiteness. With MS2C, we model situations where we expect few entities to change their type between any two consecutive time steps. Later, in Section 5, we will consider a model for settings where we expect few changes overall.

The issue with both of the aforementioned classical approaches to defining temporal graph classes is that they do not take the time component into account when deciding membership in a class. For example, if the order of the layers is permuted arbitrarily, then this has no effect on membership in $\mathcal{C}$ in either approach. Defining bipartiteness in the manner we propose does take the temporal order of the layers into consideration. It also leads to a hierarchy of temporal graph classes that are inclusion-wise between the two classes defined in the two aforementioned more traditional approaches: It is easy to see that $(\mathcal{G}, 0)$ is a `yes`-instance for MS2C if and only if the underlying graph of $\mathcal{G}$ is bipartite. Conversely, if any layer of $\mathcal{G}$ is not bipartite, then $(\mathcal{G}, d)$ is a `no`-instance no matter the value of d. The two main drawbacks to defining temporal bipartiteness in this way are that (i) there is not one class of bipartite temporal graphs, but an infinite hierarchy depending on the value of d and (ii) as we will show, testing for bipartiteness in this sense is computationally much harder, but we will attempt to partially remedy this by analyzing the problem's parameterized complexity for a variety of parameters.

Related work. The multistage framework is still young, but several problems have been investigated in it, mostly in the last couple of years, including Matching [3, 7, 18], Knapsack [4], s-t Path [17], Vertex Cover [16], Committee Election [5], and others [2]. The framework has also been extended to goals other than minimizing the number of changes in the solution between layers [19, 22]. Since these types of problems are NP-hard even in fairly restricted settings, most research has focused on their parameterized complexity and approximability. MS2C is most closely related to Multistage 2-SAT [13] (see Section 2).

Our contributions. We prove that MS2C remains NP-hard even if $d = 1$ or if $\tau = 2$. We then analyze three ways of transferring structural graph parameters to the multistage setting: the maximum over the layers, the sum over all layers' values, and its value on the underlying graph plus τ. We provide several (fixed-parameter) intractability and tractability results regarding these three notions of structural parameterizations (see Figure 1). Finally, we show that a slightly modified version of the problem in which there is no restriction on the number of changes between any two consecutive layers, but on the total number of changes throughout the lifetime of the graph, is fixed-parameter tractable with respect to the number of allowed changes. The proofs of statements marked with ★ are deferred to the full version.

Discussion and outlook. While we proved that MS2C is NP-hard even if $\tau = 2$ or if $d = 1$, we leave open whether it is fixed-parameter tractable for the combined parameter $\tau + d$. We introduce a framework for analyzing the parameterized complexity of multistage problems regarding structural graph parameters. We resolve the parameterized complexity of MS2C

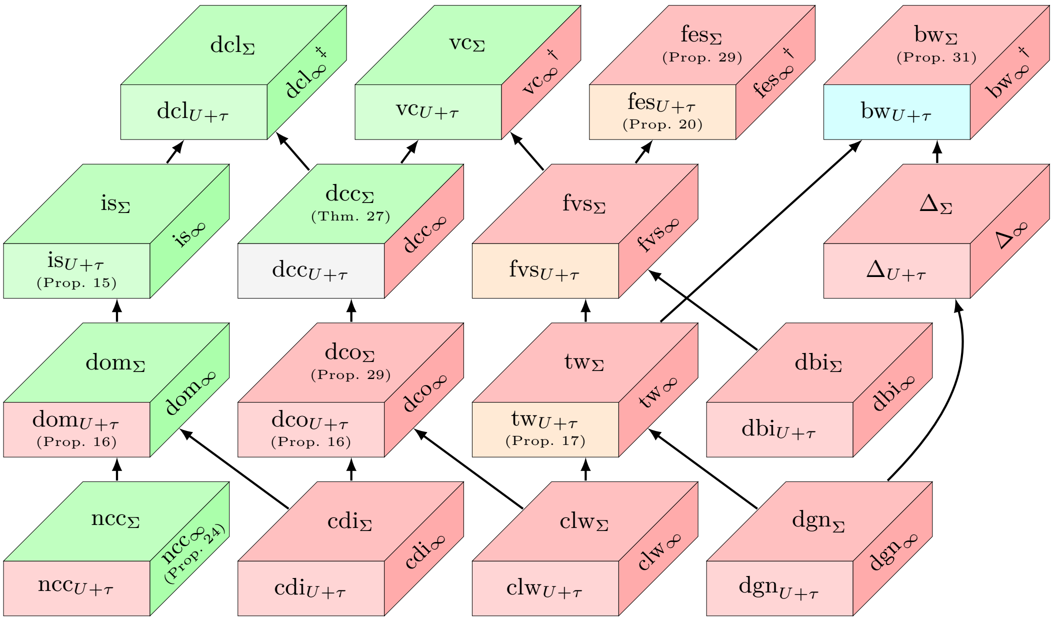

Figure 1 Overview of selected structural parameters and our results (green: in FPT; orange: XP and W[1]-hard red: para-NP-hard; blue: XP and open whether FPT or W[1]-hard; gray: open). [Δ: maximum degree; bw: bandwidth; cdi: diameter of connected component; clw: clique-width; dbi: distance to bipartite; dcc: distance to co-cluster; dcl: distance to clique; dco: distance to cograph; dgn: degeneracy; dom: domination number; fes: feedback edge number; fvs: feedback vertex number; is: independence number; ncc: number of connected components; tw: treewidth; vc: vertex cover number; for definitions of these parameters, see Section 2 or [32].] † (Proposition 25) ‡ (no polynomial kernel unless NP $\subseteq$ coNP / poly).

with respect to most of the parameters, but two cases are left open (cf. Figure 1). For instance, we proved that MS2C is in XP when parameterized by bw$_{U+\tau}$, but we do not know whether it is in FPT or W[1]-hard. Another interesting example is MS2C parameterized by dcc$_{U+\tau}$, for which we do not know whether it is contained in XP or para-NP-hard. Note that we proved fixed-parameter tractability regarding dcc$_\Sigma$. Finally, we suspect that it may also be worthwhile to investigate other multistage graph problem in our framework.

2 Preliminaries

We denote by $\mathbb{N}$ ($\mathbb{N}_0$) the natural number excluding (including) zero. We use standard terminology from graph theory [9] and parameterized algorithmics [8].

Static and temporal graphs. We will frequently refer to graphs as static graphs in order to avoid confusion with temporal graphs. A static graph $G = (V, E)$ is 2-*colorable* if it admits a proper 2-coloring. It is well-known that a static graph is 2-colorable if and only if contains no odd cycle. This can be checked in $\mathcal{O}(|V| + |E|)$ time by a simple search algorithm.

Let $G = (V, E)$ be a static graph. The *independence number* (is(G)) is the size of a largest set $X \subseteq V$ such that $G[X]$ is edgeless. The *domination number* (dom(G)) is the size of a smallest set $X \subseteq V$ such that every vertex in $V \setminus X$ has a neighbor in X. The maximum degree ($\Delta(G)$) is the maximum number of edges incident to a single vertex. A set of $X \subseteq V$ is a *connected component* if there is a path between any two vertices in X and no edge between X and $V \setminus X$. We denote the *number of connected components* in G by $\mathrm{ncc}(G)$. The *feedback edge number* ($\mathrm{fes}(G)$) is $|E| - |V| + \mathrm{ncc}(G)$ or equivalently the size of a minimum $X \subseteq E$ such that $G - X$ is acyclic. If S_n denotes the set of all permutations of $\{1, \ldots, n\}$ and $V = \{v_1, \ldots, v_n\}$, then the *bandwidth* ($\mathrm{bw}(G)$) is $\min_{\pi \in S_n} \max_{\{v_i, v_j\} \in E} |\pi(i) - \pi(j)|$. A *tree decomposition* of G is a pair $(\mathcal{T}, \{\mathcal{X}_\alpha \mid \alpha \in V(\mathcal{T})\})$ where $\mathcal{T}$ is a tree with node set $V(\mathcal{T})$ and $\mathcal{X}_\alpha \subseteq V$ for every $\alpha \in V(\mathcal{T})$ such that (i) $\bigcup_{\alpha \in V(\mathcal{T})} \mathcal{X}_\alpha = V$, (ii) for every $\{u, v\} \in E$ there is an $\alpha \in V(\mathcal{T})$ such that $u, v \in \mathcal{X}_\alpha$, and (iii) for every $v \in V$ the node set $\{\alpha \in V(\mathcal{T}) \mid v \in \mathcal{X}_\alpha\}$ induces a subtree of $\mathcal{T}$. The *width* of $(\mathcal{T}, \mathcal{X})$ is $\max_{\alpha \in V(\mathcal{T})} |\mathcal{X}_\alpha| - 1$. The *treewidth* ($\mathrm{tw}(G)$) is the minimum width of a tree decomposition of G. If $\mathcal{C}$ is a class of static graphs, then $X \subseteq V$ is a $\mathcal{C}$*-modulator* in G if $G - X \in \mathcal{C}$. The (i) *distance to cograph*, (ii) *vertex cover number*, (iii) *distance to bipartite*, and (iv) *distance to co-cluster* are the size of a minimum $\mathcal{C}$-modulator where $\mathcal{C}$ is the set of all (i) cographs, (ii) edgeless graphs, (iii) bipartite graphs, and (iv) co-clusters, respectively.

A *temporal graph* $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ consists of a finite vertex set V and τ edge sets $E_1, \ldots, E_\tau \subseteq \binom{V}{2}$. The *underlying graph* of $\mathcal{G}$ is the static graph $\mathcal{G}_U := (V, \bigcup_{t=1}^{\tau} E_t)$. For $t \in \{1, \ldots, \tau\}$, the t-th layer of $\mathcal{G}$ is also a static graph, namely $\mathcal{G}_t := (V, E_t)$. The lifetime of $\mathcal{G}$ is τ, the number of layers.

If $f_1, f_2 \colon X \to Y$ are two functions that share a domain and a codomain, then $\delta(f_1, f_2) := |\{x \in X \mid f_1(x) \neq f_2(x)\}|$ is the number of elements of X whose value under f_1 differs from the value under f_2.

Preliminary results. There is a connection between MS2C and the Multistage 2-SAT problem [13], which allows to transfer positive algorithmic results from the latter to the former.

▶ **Observation 1.** *There is a polynomial time algorithm that, taking an instance of* Multistage 2-Coloring*, constructs an equivalent instance of* Multistage 2-SAT *with n variables, $2m$ clauses, and $d' = d$.*

Proof. For each vertex v, construct a variable x_v. For each edge $\{v, w\}$ in a layer, construct the clauses $(x_v \vee x_w), (\overline{x_v} \vee \overline{x_w})$. ◀

Results on Multistage 2-Sat [13] hence imply the following.

▶ **Corollary 2.** Multistage 2-Coloring *is (i) polynomial-time solvable if $d \in \{0, n\}$, (ii) in* XP *regarding $n - d$ and $\tau + d$, (iii)* FPT *regarding $m + n - d$ and n, and (iv) admits a polynomial kernel regarding $m + \tau$ and $n + \tau$.*

We briefly note the following:

▶ **Observation 3.** *Given two 2-colorable graphs $G = (V, E)$ and $G' = (V, E')$, and two 2-colorings f of G and f' of G', we can determine $\delta(f, f')$ in linear time.*

We can strengthen the first statement in Corollary 2 with the following proposition:

▶ **Proposition 4 (★).** Multistage 2-Coloring *is polynomial-time solvable if $d \geq \frac{1}{2}n$.*

Testing all sequences of functions $f_1, \ldots, f_\tau \colon V \to \{1, 2\}$ gives us the following:

▶ **Observation 5.** Multistage 2-Coloring *can be decided in time $\mathcal{O}(2^{\tau n} \cdot m)$ where τ is the lifetime, n the number of vertices, and m the number of time edges in a temporal graph.*

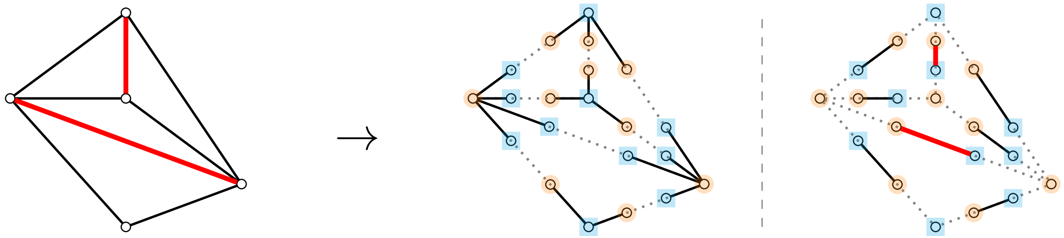

Figure 2 Illustration of Construction 1: The input graph G on the left hand-side (thick/red edges indicate a solution) and the output temporal graph $\mathcal{G}$ on the right-hand side (thick/red edges in the second layer indicate where a recoloring was made; gray/dotted lines help to match with original edges from G).

3 NP-hard cases

We start by proving some complexity lower bounds for MULTISTAGE 2-COLORING. We will show that the problem is NP-hard in two fairly restricted cases.

3.1 Few changes allowed

▶ **Theorem 6** (★). *MULTISTAGE 2-COLORING is NP-hard even for $d = 1$ and restricted to temporal graphs where each layer contains just three edges and has maximum degree one.*

The proof is deferred to the full version.

3.2 Few stages

▶ **Theorem 7** (★). *MULTISTAGE 2-COLORING is* NP*-hard on temporal graphs with at least two layers each of which is a forest.*

To prove Theorem 7, we give a polynomial-time many-one reduction from the NP-complete [35] EDGE BIPARTIZATION problem defined by:

▶ **Problem 2.** EDGE BIPARTIZATION

Input: An undirected graph $G = (V, E)$ and $k \in \mathbb{N}_0$.
Question: Is there a set of edges $E' \subseteq E$ with $|E'| \leq k$ such that $G - E'$ is bipartite?

▶ **Construction 1.** Let $G = (V, E)$ be a graph and let $k \in \mathbb{N}_0$. We assume that $V = \{v_1, \ldots, v_n\}$. We construct an instance $(\mathcal{G}, d)$ of MS2C with $\mathcal{G} := (V', E_1, E_2)$ and $d := k$ as follows (see Figure 2 for an illustrative example).

The underlying graph of $\mathcal{G}$ is obtained by subdividing each edge in G twice. Let u_i^e and u_j^e be the two vertices obtained by subdividing $e = \{v_i, v_j\}$ where u_i^e is adjacent to v_i and u_j^e to v_j. Then, $V' := V \cup \{u_i^e, u_j^e \mid e = \{v_i, v_j\} \in E\}$. The first layer of $\mathcal{G}$ has edge set $E_1 := \{\{v_i, u_i^e\} \mid i \in \{1, \ldots, n\}, e \in E, v_i \in e\}$. The second layer has edge set $E_2 := \{\{u_i^e, u_j^e\} \mid e = \{v_i, v_j\} \in E\}$. ⌟

▶ **Lemma 8** (★). *Instance (G, k) is a* yes*-instance for* EDGE BIPARTIZATION *if and only if instance $(\mathcal{G}, d)$ output by Construction 1 is a* yes*-instance for* MULTISTAGE 2-COLORING.

The reduction also implies the following:

▶ **Proposition 9.** *Unless the ETH fails,* Multistage 2-Coloring *admits no* $\mathcal{O}(2^{o(n+m)})$*-time algorithm, where* n *is the number of vertices and* m *is the number of time edges in a temporal graph, even for* $\tau = 2$.

Proof. Unless the ETH fails, Edge Bipartization cannot be solved in time $\mathcal{O}(2^{o(n)})$, where n is the number of vertices. This follows from the corresponding lower bound for Maximum Cut [27]. The instance output by Construction 1 contains $n+2m$ vertices. The claim follows by Lemma 8. ◀

4 Parameterized complexity

In the previous section we showed that Multistage 2-Coloring is NP-hard, even for constant values of τ and d. In this section, we study the parameterized complexity of Multistage 2-Coloring. To begin with, we will now show that Multistage 2-Coloring is fixed-parameter tractable with respect to $n - d$. This is in contrast to Multistage 2-SAT, which is W[1]-hard with respect to this parameter [13, Theorem 3.6].

▶ **Proposition 10.** Multistage 2-Coloring *is fixed-parameter tractable regarding* $n - d$.

Proof. If $d \geq \frac{n}{2}$, the problem can be solved in polynomial time (see Proposition 4). If $d < \frac{n}{2}$, then it follows that $n < 2(n - d)$. Hence, the fixed-parameter tractability of MS2C with respect to n (see Corollary 2) implies fixed-parameter tractability with respect to $n - d$. ◀

Additionally, we note the following kernelization lower bound.

▶ **Proposition 11 (★).** *Unless* NP $\subseteq$ coNP / poly*,* Multistage 2-Coloring *admits no problem kernel of size polynomial in the number* n *of vertices.*

In the following, we will consider the parameterized complexity of Multistage 2-Coloring with respect to structural graph parameters. Research on the parameterized complexity of multistage problems has thus far mostly focused on the parameters that are given as part of the input such as d or τ. Although Fluschnik et al. [17] considered the vertex cover number and maximum degree of the underlying graph, there has been no systematic study of multistage problems concerned with structural parameters of the input temporal graph. We seek to initiate this line of research in the following. It follows the call by Fellows et al. [10, 12] to investigate problems' "parameter ecology" in order to fully understand what makes them computationally hard. We will begin with a short discussion of how graph parameters can be applied to multistage problems. This question is closely related to issues that arise when applying such parameters to temporal graph problems (see [14] and [26, Sect. 2.4]).

A *(temporal) graph parameter* p is a function that maps any (temporal) graph G to a nonnegative integer $p(G)$. We will consider three ways of transferring graph parameters to temporal graphs. If p is a graph parameter, $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ is a temporal graph, $\mathcal{G}_t \coloneqq (V, E_t)$ its t-th layer, and $\mathcal{G}_U \coloneqq (V, \bigcup_{t=1}^{\tau} E_t)$ its underlying graph, then we define:

$$p_\infty(\mathcal{G}) \coloneqq \max_{t \in \{1,\dots,\tau\}} p(\mathcal{G}_t), \qquad (\textit{maximum} \text{ parameterization})$$

$$p_\Sigma(\mathcal{G}) \coloneqq \sum_{t=1}^{\tau} \max\{1, p(\mathcal{G}_t)\}, \text{ and} \qquad (\textit{sum} \text{ parameterization})$$

$$p_{U+\tau}(\mathcal{G}) \coloneqq p(\mathcal{G}_U) + \tau. \qquad (\textit{underlying graph} \text{ parameterization})$$

We will briefly explain our choice to define these parameters in this manner and describe the relationship between the parameters. For any two (temporal) graph parameters p_1 and p_2, the first parameter p_1 *is larger than* p_2, written $p_1 \succeq p_2$ or $p_2 \preceq p_1$, if there is a function $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ such that $f(p_1(G)) \geq p_2(G)$ for all (temporal) graphs G. Such relationships between parameters are useful because, if $p_1 \succeq p_2$, then any problem that is fixed-parameter tractable with respect to p_2 is also fixed-parameter tractable with respect to p_1. The $\succeq$-relation between static graph parameters is well-understood [21, 30, 31, 32, 33]. We will use these relationships implicitly and explicitly throughout this article. Many of the results claimed in Figure 1 will not be explicitly proved, because they are immediate consequences of other results and the $\succeq$-relation. The relationships under $\succeq$ between selected graph parameters are pictured in that figure.

When it comes to transferring graph parameters from the static to the multistage setting, the parameters p_∞ and $p_{U+\tau}$ simply apply the graph parameter to the individual layers and to the underlying graph, respectively, and were used in a similar manner by Fluschnik et al. [14] and Molter [26]. The reasoning behind the definition of the sum parameterization may not be quite as obvious. It seems natural to consider the sum of the parameters over all layers. The issue with this is that it may not preserve the $\succeq$-relation. For example, it is well-known that feedback vertex number is a larger parameter (in the sense of $\succeq$) than treewidth. However, consider a temporal graph where each layer is a forest. Then, the sum of the feedback vertex numbers of the layers is 0, but the sum of the layers' treewidths is τ. Hence, treewidth is no longer bounded from above by the feedback vertex number. Our definition gets around this problem. In fact, all three aforementioned ways of transferring parameters from the static to the multistage setting preserve the $\succeq$-relation:

▶ **Proposition 12.** *Let p and q be graph parameters with $p \succeq q$. Then, $p_\alpha \succeq q_\alpha$ for any $\alpha \in \{\infty, \Sigma, U+\tau\}$.*

Proof. Let $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ be a function such that $f(p(G)) \geq q(G)$ for all static graphs G. Without loss of generality, we may assume that (i) f is monotonically increasing, that is, $f(a) \geq f(b)$ if $a \geq b$, and (ii) $f(a) \geq a$ for every $a \in \mathbb{N}_0$ (consider $f'(a) := a + \max_{b\in\{1,\dots,a\}} f(b)$, $a \in \mathbb{N}_0$, for instance).

Let $\mathcal{G}$ be an arbitrary temporal graph. Then:

$$f(p_\infty(\mathcal{G})) = f\left(\max_{t\in\{1,\dots,\tau\}} p(\mathcal{G}_t)\right) \overset{\text{(i)}}{=} \max_{t\in\{1,\dots,\tau\}} f(p(\mathcal{G}_t)) \geq \max_{t\in\{1,\dots,\tau\}} q(\mathcal{G}_t) = q_\infty(\mathcal{G})$$

For $n \in \mathbb{N}$, let $\mathrm{Part}(n)$ denote the set of all partitions of n, that is all possible ways of writing n as $n = n_1 + n_2 + \ldots + n_r$ for $r \geq 1$ and $n_i \in \mathbb{N}$. Let $g\colon \mathbb{N}_0 \to \mathbb{N}_0$ with:

$$g(0) := 0, \quad g(n) := \max\left\{\sum_{i=1}^{r} f(n_i) \;\middle|\; (n_1,\dots,n_r) \in \mathrm{Part}(n)\right\} \text{ if } n > 0.$$

The maximum is well-defined, because $\mathrm{Part}(n)$ is finite. Then, any temporal graph $\mathcal{G}$ satisfies:

$$\begin{aligned} g(p_\Sigma(\mathcal{G})) &= g\left(\sum_{t=1}^{\tau} \max\{1, p(\mathcal{G}_t)\}\right) \geq \sum_{t=1}^{\tau} f(\max\{1, p(\mathcal{G}_t)\}) \overset{\text{(i)}}{=} \sum_{t=1}^{\tau} \max\{f(1), f(p(\mathcal{G}_t))\} \\ &\overset{\text{(ii)}}{\geq} \sum_{t=1}^{\tau} \max\{1, q(\mathcal{G}_t)\} = q_\Sigma(\mathcal{G}). \end{aligned}$$

(Note that the first inequality relies on the fact that every term in the sum is at least 1, since a partition can only be composed of positive summands. Therefore, this argument would not apply, if we defined the sum parameterization as simply the sum over the parameters of the individual layers.)

Lastly, for any temporal graph $\mathcal{G}$, we have:

$$g(p_{U+\tau}(\mathcal{G})) = g(p(\mathcal{G}_U) + \tau) \geq f(p(\mathcal{G}_u)) + f(\tau) \overset{\text{(ii)}}{\geq} q(\mathcal{G}_U) + \tau = q_{U+\tau}(\mathcal{G}). \qquad \blacktriangleleft$$

Finally, we will briefly consider the relationship between p_∞, p_Σ, and $p_{U+\tau}$. We will say that a graph parameter p is *monotonically increasing* if for any two static graphs $G = (V, E)$ and $G' = (V, E')$ with the same vertex set, it is the case that $E \subseteq E'$ implies $p(G) \leq p(G')$. Conversely, it is *monotonically decreasing* if $E \subseteq E'$ implies $p(G) \geq p(G')$.

▶ **Proposition 13.** *Let p be a graph parameter. Then:*

(i) $p_\infty \preceq p_\Sigma$,
(ii) $p_\Sigma \preceq p_{U+\tau}$, *if p is monotonically increasing, and*
(iii) $p_\Sigma \succeq p_{U+\tau}$, *if p is monotonically decreasing.*

Proof.

(i) Obvious.
(ii) Let $\mathcal{G}$ be a temporal graph. Note that since $\mathcal{G}_t \subseteq \mathcal{G}_U$, it follows that $p(\mathcal{G}_t) \leq p(\mathcal{G}_U)$ for all $t \in \{1, \ldots, \tau\}$. Hence:

$$\begin{aligned} p_\Sigma(\mathcal{G}) &= \sum_{t=1}^{\tau} \max\{1, p(\mathcal{G}_t)\} \leq \tau + \sum_{t=1}^{\tau} p(\mathcal{G}_t) \leq \tau + \tau \cdot p(\mathcal{G}_U) \\ &\leq (\tau + p(\mathcal{G}_U))^2 = p_{U+\tau}(\mathcal{G})^2. \end{aligned}$$

(iii) Let $\mathcal{G}$ be a temporal graph. Note that since $\mathcal{G}_t \subseteq \mathcal{G}_U$, it follows that $p(\mathcal{G}_t) \geq p(\mathcal{G}_U)$ for all $t \in \{1, \ldots, \tau\}$. If $\tau = 1$ or $p(\mathcal{G}_U) \leq 1$, the claim is obvious. Otherwise, we have that:

$$\begin{aligned} p_\Sigma(\mathcal{G}) &= \sum_{t=1}^{\tau} \max\{1, p(\mathcal{G}_t)\} \geq \sum_{t=1}^{\tau} \max\{1, p(\mathcal{G}_U)\} \geq \sum_{t=1}^{\tau} p(\mathcal{G}_U) \\ &= \tau \cdot p(\mathcal{G}_U) \geq p_{U+\tau}(\mathcal{G}). \end{aligned} \qquad \blacktriangleleft$$

We will now investigate the problem's parameterized complexity with respect to the three types of parameterizations. Figure 1 gives an overview of our results and of the abbreviations we use for the parameters. Our choice of parameters is partly motivated by Sorge and Weller's compendium [32] on graph parameters, but we limit our attention to those that are most interesting in the context of MS2C. For full definitions of the parameters, we refer the reader to Sorge and Weller's manuscript [32] or Section 2.

4.1 Underlying graph parameterization

▶ **Lemma 14.** *If $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ is a temporal graph and every layer $\mathcal{G}_t = (V, E_t)$ of $\mathcal{G}$ is bipartite for $t \in \{1, \ldots, \tau\}$, then $\mathrm{is}_{U+\tau}(\mathcal{G}) \geq 2^{-\tau}|V|$.*

Proof. (By induction on τ.) If $\tau = 1$, then $\mathcal{G}_U$ is bipartite and the larger color class in any 2-coloring of $\mathcal{G}_U$ forms an independent set containing at least $\frac{1}{2}|V|$ vertices. Suppose the claim holds for $\tau - 1$. Then, the underlying graph of $\mathcal{G}' = (V, (E_t)_{t=1}^{\tau-1})$ contains an independent set $X \subseteq V$ of size at least $2^{-(\tau-1)}|V|$. The graph $(X, \binom{X}{2} \cap E_\tau)$ is bipartite since it is a subgraph of (V, E_τ). Hence, it contains an independent set Y of size at least $\frac{1}{2}|X| \geq 2^{-\tau}|V|$. Then, Y is also an independent set in $\mathcal{G}_U$. ◀

▶ **Proposition 15.** Multistage 2-Coloring *is fixed-parameter tractable regarding* $\mathrm{is}_{U+\tau}$.

Proof. If any layer of $\mathcal{G}$ is not bipartite, then the input can be immediately rejected. Otherwise, let $\mathcal{G}_U$ be the underlying graph of $\mathcal{G}$. By Observation 5, MS2C can be solved in time $\mathcal{O}^*(2^{\tau\cdot|V|}) \leq \mathcal{O}^*(2^{\tau\cdot \mathrm{is}_{U+\tau}(\mathcal{G})\cdot 2^\tau})$. ◀

▶ **Proposition 16 (★).** Multistage 2-Coloring *is NP-hard even if* $\tau = 4$, $\mathrm{dom}(\mathcal{G}_U) \leq 2$, *and* $\mathrm{dco}(\mathcal{G}_U) = 0$. *Hence, the problem is para-NP-hard with respect to* $\mathrm{dom}_{U+\tau}$ *and* $\mathrm{dco}_{U+\tau}$.

▶ **Proposition 17 (★).** Multistage 2-Coloring *can be solved in* $\mathcal{O}^*(2^{\tau\cdot \mathrm{tw}_{U+\tau}(\mathcal{G})}\cdot(d+1)^{2\tau})$ *time. Hence, the problem is in* XP *when parameterized by* $\mathrm{tw}_{U+\tau}$.

The proof of this proposition utilizes a standard dynamic programming approach for problems parameterized by treewidth, extending it to the multistage context. Note that the running time of this algorithm also implies that Multistage 2-Coloring is fixed-parameter tractable with respect to $\tau + d + \mathrm{tw}_{U+\tau}$.

▶ **Proposition 18 (★).** Multistage 2-Coloring *is NP-hard even if* $\tau = 3$ *and* $\Delta(G) = 3$. *Hence, the problem is para-NP-hard with respect to* $\Delta_{U+\tau}$.

▶ **Proposition 19 (★).** Multistage 2-Coloring *is NP-hard even if* $\tau = 3$ *and* $\mathrm{dbi}_{U+\tau} = 2$.

Next, we will prove that Multistage 2-Coloring is W[1]-hard with respect to $\mathrm{fes}_{U+\tau}$. In fact, we will prove the following slightly stronger statement:

▶ **Proposition 20 (★).** Multistage 2-Coloring *is* W[1]*-hard when parameterized by* τ, *even if the feedback edge number* $\mathrm{fes}(\mathcal{G}_U)$ *of the underlying graph is one.*

We already showed that MS2C is XP regarding $\mathrm{tw}_{U+\tau}$, so Proposition 20 implies that it is XP and W[1]-hard when parameterized by $\mathrm{tw}_{U+\tau}$, $\mathrm{fvs}_{U+\tau}$, and $\mathrm{fes}_{U+\tau}$, since $\mathrm{tw} \preceq \mathrm{fvs} \preceq \mathrm{fes}$. The proof of Proposition 20 is a little more involved than most of the previous hardness proofs. Our reduction is from the following:

▶ **Problem 3.** Multicolored Clique (MC)
Input: A k-colored static graph $G = (V, E)$ with $V = V_1 \uplus \ldots \uplus V_k$.
Question: Does G contain a clique $X \subseteq V$ such that $|X \cap V_i| = 1$ for all $i \in \{1, \ldots, k\}$?

Multicolored Clique is W[1]-hard when parameterized by k [11, 28].

▶ **Construction 2.** Let $(G = (V, E), k)$ with $V = V_1 \uplus \ldots \uplus V_k$ be an instance of Multicolored Clique. We may assume that $|V_1| = \ldots = |V_k| = n$ (if color classes do not have the same size, we can add isolated vertices), that all V_i are independent, and that $|E| \geq \binom{k}{2}$ (otherwise, this is clearly a `no`-instance). Let $V_i = \{v_0^i, \ldots, v_{n-1}^i\}$.

We will now describe an instance $(\mathcal{G} = (V', (E_t)_{t=1}^\tau), d)$ of Multistage 2-Coloring with $\mathrm{fes}(\mathcal{G}_U) = 2$ (see Figure 3 for an illustration). We let $\tau := 2k(k-1)+3$ and $d := |E|$.

The general idea behind the reduction is as follows. We consider the steps between consecutive layers and the number of changes to the coloring in those steps. The value of τ implies that there are $2k(k-1)+2$ steps in total. There are $2k-2$ such steps for each color class in G, while the final two steps do not correspond to any color class. Of the $2k-2$ steps that correspond to $c \in \{1, \ldots, k\}$, two will be used to verify adjacency to each of the $k-1$ other color classes. In order to be able to refer to these steps easily, we will use the following notation for any $c, c' \in \{1, \ldots, k\}$, $c \neq c'$:

$$T(c \to c') := \begin{cases} 2(c-1)(k-1) + c', & \text{if } c > c', \\ 2(c-1)(k-1) + c' - 1, & \text{if } c < c', \end{cases} \quad \text{and}$$
$$T(c \Rightarrow c') := T(c \to c') + k - 1$$

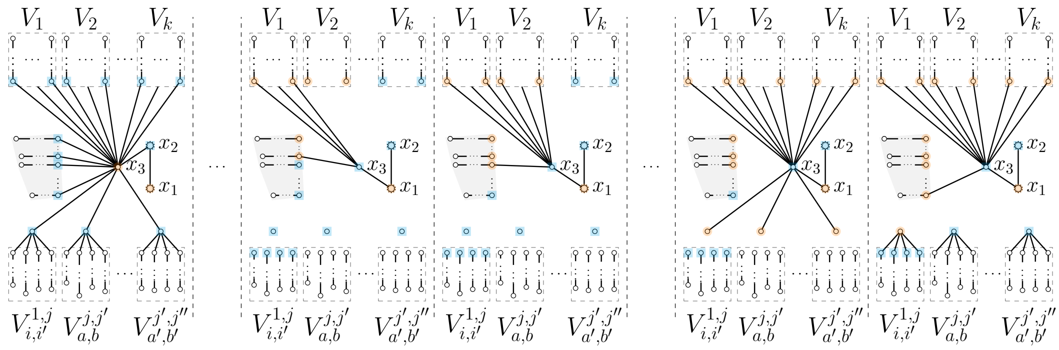

Figure 3 Illustration of Construction 2. Shown are the first layer (left), the two layers when transitioning from the phase regarding V_2 to the phase regarding V_3 (middle), the last two layers (right). In the gray area, the waste-budget gadget is depicted. In this example, the edge $\{v_i^1, v_{i'}^j\}$ is chosen into the clique. Note that many vertices (those from paths and stars) are not depicted.

We will use several gadgets. The first gadget maintains its coloring throughout most of the lifetime of the instance. We use it to enforce a particular, predictable coloring on vertices in other gadgets at certain points. The second type of gadget represents the selection of a vertex in a certain color class. If the vertex v_j^i is to be added to the clique, it forces any multistage 2-coloring to make j changes in the first $k-1$ steps corresponding to the color class i and $n-j-1$ changes in the following $k-1$ steps corresponding to this class. There is a third type of gadget. Its purpose is to verify that the vertices selected by the first gadget type are pairwise adjacent. There are numerous additional vertices whose sole purpose is to ensure that the coloring of vertices cannot change in unexpected ways. More specifically, when we say that a vertex v is *blocked* in time step t, we mean that we add d vertices that are adjacent to v in the $(t-1)$-st and t-th layers and isolated in all other layers. There are also further vertices designed to use up extraneous budget for changes during certain time steps.

We start by describing the first gadget, whose purpose is to maintain a predictable coloring so it can be used to enforce a certain coloring on other parts of the instance at particular points in time. This gadget contains the vertices x_1, x_2, x_3. The edge $\{x_1, x_2\}$ is present in every layer of $\mathcal{G}$. The edge $\{x_2, x_3\}$ exists only in the first layer, while $\{x_1, x_3\}$ is present in all layers but the first. The vertices x_1 and x_2 are blocked in every step.

Next, we define the second type of gadget, which models the selection of a vertex in a color class. The gadget representing a certain color class V_c, $c \in \{1, \dots, k\}$, consists of $(n-1)(k-1)$ vertices $w_{i,j}^c$ for $i \in \{1, \dots, n-1\}$, $j \in \{1, \dots, k-1\}$. The vertex $w_{i,j}^c$ is blocked in all time steps except for the step $T(c \to j)$ and the step $T(c \Rightarrow j)$. There is an edge between $w_{i,j}^c$ and $w_{i,j+1}^c$ in the layers from $T(c \to j+1)$ to $T(c \Rightarrow 1)$ and from $T(c \Rightarrow j)$ to $T(c+1 \to 1)$. Additionally, in the very first and in the final layer of $\mathcal{G}$, all edges $\{w_{i,j}^c, w_{i,j+1}^c\}$ are present and there is an edge from x_3 to $w_{i,1}^c$ for all $c \in \{1, \dots, k\}$ and $i \in \{1, \dots, n-1\}$. Moreover, for every $c \in \{1, \dots, k\}$, there is an edge from x_3 to $w_{i,1}^c$ for all $i \in \{1, \dots, n-1\}$ in all layers of index larger than $T(c \Rightarrow c')$, with $c' = \max\{1, \dots, k\} \setminus \{c\}$. This gadget is illustrated in the top part of Figure 4.

Next, we will describe the gadget that verifies that vertices selected in the previous gadget are pairwise adjacent. There is one such gadget for every edge $e = \{v_j^c, v_{j'}^{c'}\} \in E$, $1 \le c < c' \le k$, $j, j' \in \{0, \dots, n-1\}$. The gadget consists of a root vertex u_0^e and four paths. The root is blocked in every step except for the final two. There is an edge between u_0^e and

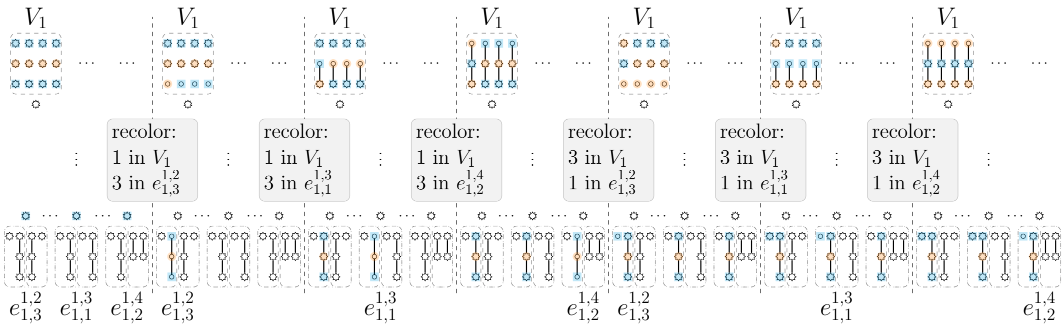

Figure 4 Illustrative example of the recolorings in Construction 2. Here, $n = 5$, $k = 4$, $e_{1,3}^{1,2} \coloneqq \{v_1^1, v_3^2\}$, $e_{1,1}^{1,3} \coloneqq \{v_1^1, v_1^3\}$, and $e_{1,2}^{1,4} \coloneqq \{v_1^1, v_2^4\}$. The recolorings here represents the case that vertex v_1^1 is chosen into the clique, together with its incident edges to v_3^2, v_1^3, and v_2^4.

x_3 in the first and the $(\tau - 2)$nd layer. The first vertex of each of the four paths is adjacent to u_0^e in the first and in the final layer. The edges of the paths are present in every layer. These paths consist of $n - 1 - j$, j, $n - 1 - j'$, and j' vertices, respectively. The vertices on the path of size $n - 1 - j$ are blocked in every time step except for step $T(c \to c')$. Those on the path of size j are blocked except for step $T(c \Rightarrow c')$. The vertices on the path of size $n - 1 - j'$ are blocked except for step $T(c' \to c)$. Finally, those on the path of size j' are blocked except for step $T(c' \Rightarrow c)$.

Finally, there is a gadget whose purpose is to waste extraneous budget for changes. It consists of $\tau - 2$ paths. There are $\tau - 4$ paths $P_3, \ldots, P_{\tau-2}$ containing $d - (n - 1)$ vertices each, one path P_2 that consists of $d - n$ vertices, and one path P_τ that consists of $\binom{k}{2}$ vertices. For each $i \in \{2, \ldots, \tau\} \setminus \{\tau - 1\}$, the first vertex in P_i is adjacent to x_3 exactly in the first and ith layer, where in all but the ith layer, all vertices from P_i are blocked. ⌟

The proof of the correctness of this reduction is deferred to the full version. We will briefly sketch a high-level description of this proof. All vertices in the gadget for a color class c must be re-colored at some point. Some number $i(k - 1)$ is re-colored in the first $k - 1$ steps corresponding to the color class and the remaining $(n - i - 1)(k - 1)$ are re-colored during the subsequent $k - 1$ steps (see Figure 4 for an illustration). That is, vertex v_i^c from color class c is added to the clique. In the final step, only $|E| - \binom{k}{2}$ vertices u_0^e can be re-colored. The other $\binom{k}{2}$ vertices correspond to edges that have both endpoints in the clique. The adjacency verification gadget ensures that, if u_0^e is not re-colored in the final step, then its endpoints must be selected to be part of the clique. This works because the four paths in this gadget must be re-colored in steps that belong to the color classes of the edge's endpoints.

▶ **Lemma 21** (★). *The input instance to Construction 2 is a* **yes***-instance for* Multicolored Clique *if and only if the output instance is a* **yes***-instance for* Multistage 2-Coloring.

Finally, fixed-parameter tractability with respect to $\mathrm{vc}_{U+\tau}$ can be proved using Theorem 27 (see Section 4.3) and the interplay between the different parameters (cf. Propositions 12 and 13).

▶ **Proposition 22** (★). Multistage 2-Coloring *is fixed-parameter tractable regarding* $\mathrm{vc}_{U+\tau}$.

4.2 Maximum parameterization

We turn our attention to the parameterized complexity of Multistage 2-Coloring with respect to several structural parameters under the maximum parameterization. We begin with ncc_∞, the maximum number of connected components over all layers. Observe that under any proper 2-coloring the color of a single vertex determines the coloring of its entire connected component.

▶ **Observation 23.** *Every 2-colorable static graph with N connected components admits exactly 2^N different proper 2-colorings.*

This implies that MS2C is fixed-parameter tractable with respect to ncc_∞.

▶ **Proposition 24.** Multistage 2-Coloring *admits an $\mathcal{O}(4^{\mathrm{ncc}_\infty(\mathcal{G})}\tau)$-time algorithm.*

Proof. Let $N \coloneqq \mathrm{ncc}_\infty(\mathcal{G})$. We create an auxiliary static directed graph in the following manner. For each layer of $\mathcal{G}$, we include a node for every one of the at most 2^N many 2-colorings of this layer. There is a directed edge from a node representing a coloring of $\mathcal{G}_t$ to a node representing a coloring of $\mathcal{G}_{t+1}$ if the recoloring cost between the two is at most d. Finally, add two nodes s, t and connect s to every node corresponding to a coloring of the first layer and connect every node that corresponds to a coloring of the final layer to t. Then, $(\mathcal{G}, d)$ is a `yes`-instance if and only if the auxiliary graph contains a path from s to t. Moreover, the auxiliary graph contains at most $\mathcal{O}(4^{\mathrm{ncc}_\infty(\mathcal{G})}\tau)$ edges. ◀

This result is essentially a stronger version of the statement in Corollary 2 that Multistage 2-Coloring is fixed-parameter tractable with respect to n, the number of vertices. However, ncc and larger parameters are the only structural parameters that yield fixed-parameter tractability with respect to the maximum parameterization.

▶ **Proposition 25 (★).** Multistage 2-Coloring *is NP-hard even for constant values of* vc_∞, fes_∞, *and* bw_∞.

We note that Proposition 11 implies that MS2C does not admit a polynomial kernel for any parameter p listed in Figure 1 under the maximum parameterization, since $n \succeq p_\infty$ for all of these parameters.

4.3 Sum parameterization

We start with the parameterized complexity of Multistage 2-Coloring with respect to several structural parameters under the sum parameterization. For ncc_Σ, fixed-parameter tractability follows from that for ncc_∞.

We start by proving that MS2C is fixed-parameter tractable with respect to the distance to co-cluster under the sum parameterization. This stands in contrast to the maximum parameterization (see Proposition 25). A graph is a co-cluster if and only if it does not contain $K_2 + K_1$ as an induced subgraph. By a general result obtained by Cai [6], this implies that the problem of determining whether $\mathrm{dcc}(G) \le k$ for a static graph G is fixed-parameter tractable with respect to k. We will make use of the following fact:

▶ **Observation 26.** *If G is a co-cluster, then G is edgeless or connected.*

▶ **Theorem 27.** Multistage 2-Coloring *is fixed-parameter tractable regarding* dcc_Σ.

If $G = (V, E)$ is a graph, then a function $\tilde{f}\colon V \to \{1, 2, \perp\}$ is a *proper partial 2-coloring* if the restriction of $\tilde{f}$ to $V' := \{v \in V \mid \tilde{f}(v) \neq \perp\}$ is a proper 2-coloring of $G[V']$. If $\tilde{f}$ is a proper partial and f is a proper 2-coloring of G, then f is an *extension* of $\tilde{f}$, if $\tilde{f}(v) \in \{\perp, f(v)\}$ for every $v \in V$. We will use the following as an intermediate problem.

▶ **Problem 4.** MULTISTAGE 2-COLORING EXTENSION (MS2CE)

Input: A temporal graph $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$, proper partial 2-colorings $\tilde{f}_1, \ldots, \tilde{f}_\tau\colon V \to \{1, 2, \perp\}$, and an integer $d \in \mathbb{N}_0$.

Question: Are there $f_1, \ldots, f_\tau\colon V \to \{1, 2\}$ such that f_t is an extension of $\tilde{f}_t$ and a proper 2-coloring of (V, E_t) for every $t \in \{1, \ldots, \tau\}$ and $\delta(f_t, f_{t+1}) \leq d$ for every $t \in \{1, \ldots, t-1\}$?

We have the following immediate reduction rule for MS2CE.

▶ **Reduction Rule 1.** If an edge e has two colored endpoints, then delete e.

▶ **Lemma 28.** MULTISTAGE 2-COLORING EXTENSION *is polynomial-time solvable if the input does not contain any edges.*

Proof. We reduce MULTISTAGE 2-COLORING EXTENSION with no edges to the following job scheduling problem:

▶ **Problem 5.** $(1 \mid r_j, p_j = 1 \mid L_{\max})$ SCHEDULING

Input: A list of jobs $j_1, \ldots, j_n$, where each job $j_i = (r_i, d_i)$ has a release date $r_i \in \mathbb{N}_0$ and a due date $d_i \in \mathbb{N}_0$, and a maximum lateness $L \in \mathbb{N}_0$.

Question: Is there a schedule $s\colon \{j_1, \ldots, j_n\} \to \mathbb{N}_0$ such that (i) $s(j_i) \neq s(j_{i'})$ if $i \neq i'$, (ii) $s(j_i) \geq r_i$ for all $i \in \{1, \ldots, n\}$, and (iii) $s(j_i) - d_i \leq L$ for all $i \in \{1, \ldots, n\}$?

Horn [20, Sect. 2] showed that this scheduling problem can be solved by a polynomial-time greedy algorithm that always schedules the available job with the earliest due date. Let $(\mathcal{G} = (V, (\emptyset)_{t=1}^{\tau}), \tilde{f}_1, \ldots, \tilde{f}_\tau, d)$ be an instance for MS2CE. We will say that vertex $v \in V$ between $t_1, t_2 \in \{1, \ldots, \tau\}$ is *forced to be re-colored* $i \in \{1, 2\}$ if: (i) $t_1 < t_2$ and there is no t_3 with $t_1 < t_3 < t_2$ such that $\tilde{f}_{t_3}(v) \neq \perp$, (ii) $\tilde{f}_{t_2}(v) = i \in \{1, 2\}$, and (iii) $\tilde{f}_{t_1}(v) = 3 - i$. Let $\mathcal{R} \subseteq V \times \{1, \ldots, \tau - 1\} \times \{2, \ldots, \tau\} \times \{1, 2\}$ be the set of all forced re-colorings. Specifically, $(v, t_1, t_2, i) \in \mathcal{R}$ if and only if v is forced to be re-colored i between t_1 and t_2.

In the machine scheduling model, only one job can be performed per time step, but, in a solution for an MS2C instance, up to d vertices can be re-colored. Hence, we fill each transition between two layers with d time slots. For $t \in \{1, \ldots, \tau - 1\}$, the time slots $d(t-1)+1, \ldots, dt$ correspond to changes in the coloring between the layers t and $t+1$. For any forced re-coloring $(v, t_1, t_2, c) \in \mathcal{R}$, we create a job j_i with release date $r_i = d(t_1 - 1) + 1$ and due date $d_i = dt_2$. We will show that the given instance of MS2CE admits a solution if and only if this set of jobs admits a schedule with maximum lateness 0.

($\Rightarrow$) Suppose that $f_1, \ldots, f_\tau$ is a solution to the instance that extends $\tilde{f}_1, \ldots, \tilde{f}_\tau$. It is easy to see that, if $(v, t_1, t_2, i) \in \mathcal{R}$, then $f_{t_1}(v) \neq f_{t_2}(v)$. Hence, there must be a t with $f_t(v) \neq f_{t+1}(v)$ and $t \in \{t_1, \ldots, t_2 - 1\}$. Then, a machine schedule for the instance described above can be constructed by scheduling the job corresponding to (v, t_1, t_2, i) in one of the slots $d(t-1)+1, \ldots, dt$. Since $\delta(f_t, f_{t+1}) \leq d$, there are enough slots.

($\Leftarrow$) Suppose that we are given a machine schedule with maximum lateness 0 for the aforementioned instance. We construct an initial coloring f_1 by assigning each vertex v the color i, if there is a $t \in \{1, \ldots, \tau\}$ such that $\tilde{f}_t(v) = i \in \{1, 2\}$ and $\tilde{f}_{t'}(v) = \perp$ for all $t' < t$. If $\tilde{f}_t(v) = \perp$ for all $t \in \{1, \ldots, \tau\}$, then we assign $f_1(v)$ arbitrarily. We iteratively

Algorithm 1 FPT-algorithm regarding dcc_Σ on input $\mathcal{G} = (V, (E_t)_{t=1}^{\tau}), d \in \mathbb{N}_0$.

$T^+, T^- \leftarrow \emptyset$;
foreach $t \in \{1, \ldots, \tau\}$ **do**
  $X_t \leftarrow$ a minimum set such that $G_t - X_t$ is a co-cluster;
  if $G_t - X_t$ is connected **then** $T^+ \leftarrow T^+ \cup \{t\}$ **else** $T^- \leftarrow T^- \cup \{t\}$;
foreach $g_1 : X_1 \to \{1, 2\}, \ldots, g_\tau : X_\tau \to \{1, 2\}$ **do** `// ≤` 2^{dcc_Σ} `many`
  foreach $t \in \{1, \ldots, \tau\}$ **do**
    if $t \in T^-$ **then** while $\exists \{u, v\} \in E_t$ s.t. $g_t(u) = i$ and $g_t(v)$ is undefined, let $g_t(v) \leftarrow 3 - i$;
    if $t \in T^+$ **then** $F_t \leftarrow \{g_t^1, g_t^2\}$ with the two possible proper 2-colorings g_t^1, g_t^2 of $G_t - X_t$;
  foreach $(g'_{t_1}, \ldots, g'_{t_{|T^-|}}) \in \bigtimes_{t \in T^-} F_t$ **do** `// ≤` 2^τ `many`
    Let $\tilde{f}_t \leftarrow g_t$ if $t \in T^-$ and $\tilde{f}_t \leftarrow g_t \cup g'_t$ if $t \in T^+$;
    if $\tilde{f}_1, \ldots, \tilde{f}_\tau$ *are proper partial colorings* **then**
      if $(\mathcal{G}, \tilde{f}_1, \ldots, \tilde{f}_\tau, d)$ *is a* `yes`*-instance for* MS2CE **then**
        return `yes` `// decidable in polynomial time (Lemma 28)`
return `no`

construct $f_2, \ldots, f_\tau$ as follows. We let $f_{t+1}(v) = 3 - f_t(v)$ if the given schedule assigns a job j_i corresponding to a forced re-coloring $(v, t_1, t_2, 3 - f_t(v)) \in \mathcal{R}$ to a slot between $d(t-1) + 1$ and dt. Otherwise, we let $f_{t+1}(v) = f_t(v)$. ◀

The idea in the proof of Theorem 27 is as follows. After computing a distance-to-co-cluster set for each layer, we check for all possible colorings of these sets, and then propagate the colorings. We finally arrive at an instance of MS2CE with no edges, which is decidable in polynomial time.

Proof of Theorem 27. Let $\mathcal{I} = (\mathcal{G}, d)$ be an instance of Multistage 2-Coloring. Let $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ and $G_t := (V, E_t)$ be the t-th layer of G. Let $k := \sum_{t=1}^{\tau} \mathrm{dcc}(G_t)$. The following algorithm is summarized in pseudocode in Algorithm 1.

For each $t \in \{1, \ldots, \tau\}$, using Cai's algorithm [6], we can compute in $2^{\mathcal{O}(k)} \cdot |G_t|^{\mathcal{O}(1)}$ time a minimum set $X_t \subseteq V$ such that $G_t - X_t$ is a co-cluster. Let (T^+, T^-) be a partition of $\{1, \ldots, \tau\}$ such that $t \in T^+$ if and only if $G_t - X_t$ is connected (see Observation 26). For $t \in T^+$, let $V_t := V(G_t - X_t)$, and for $t \in T^-$, let $V_t := \{v \in V(G_t - X_t) \mid \deg_{G_t}(v) > 0\}$ be the vertices in $G_t - X_t$ incident to at least one edge in G_t. We then iterate over all the at most 2^k possible partial 2-colorings of $(X_1, \ldots, X_\tau)$. For every $t \in T^+$ there are only two possible proper 2-colorings of $G_t - X_t$. We iterate over all the at most 2^τ possible 2-colorings of these layers. For every $t \in T^-$, if there is an uncolored vertex v with a neighbor w colored $i \in \{1, 2\}$, then color v with color $3 - i$. Note that this colors all vertices in V_t. Let $\tilde{f}_1, \ldots, \tilde{f}_\tau$ be the resulting partial coloring. The important thing to note is that for every $t \in \{1, \ldots, \tau\}$ and every edge in E_t both its endpoints are colored by $\tilde{f}_t$. If one of $\tilde{f}_1, \ldots, \tilde{f}_\tau$ is not proper, we reject the coloring, otherwise we proceed as follows.

Construct the instance $\tilde{\mathcal{I}} = (\mathcal{G}, (\tilde{f}_t)_{t=1}^{\tau}, d)$ of Multistage 2-Coloring Extension. Since every edge has two colored endpoints, applying Reduction Rule 1 exhaustively results in an instance $\tilde{\mathcal{I}}' = (\mathcal{G}', (\tilde{f}_t)_{t=1}^{\tau}, d)$ of Multistage 2-Coloring Extension where $\mathcal{G}'$ contains no edge. Hence, due to Lemma 28, we can solve $\tilde{\mathcal{I}}'$ in polynomial-time. Thus, the overall running time is in $\sum_{t=1}^{\tau} 2^{\mathcal{O}(k)} \cdot |G_t|^{\mathcal{O}(1)} + 2^{k+\tau} |\mathcal{G}|^{\mathcal{O}(1)}$.

Clearly, if $\tilde{\mathcal{I}}'$ is a yes-instance in one choice, then $\mathcal{I}$ is a yes-instance of MS2C. That the opposite direction is correct too is also not hard not see. Note that every solution $f_1, \dots, f_\tau$ induces a proper partial coloring $\tilde{f}_1, \dots, \tilde{f}_\tau$, where $\tilde{f}_t$ is induced on $V_t \cup X_t$ for every $t \in \{1, \dots, \tau\}$, that we will eventually check. Moreover, the resulting input to MS2CE is clearly a yes-instance: $f_1, \dots, f_\tau$ is a solution to $(\mathcal{G}, (\tilde{f}_t)_{t=1}^{\tau}, d)$. ◀

▶ **Proposition 29** (★). Multistage 2-Coloring *is* NP*-hard even for constant values of (i)* dco_Σ*, (ii)* fes_Σ*, and (iii)* Δ_Σ*.*

Our final result on structural parameters concerns bw_Σ, that is, bandwidth with the sum parameterization. We first briefly note the following:

▶ **Observation 30.** *Let G be an undirected graph. If every connected component in G contains at most k vertices, then* $\mathrm{bw}(G) \leq k - 1$*.*

We can use this observation to show that Multistage 2-Coloring is para-NP-hard when parameterized by bw_Σ.

▶ **Proposition 31** (★). Multistage 2-Coloring *is* NP*-hard even if* bw_Σ *is constant.*

5 Global budget

The problem we have considered so far is the multistage version of 2-Coloring with a *local* budget. Heeger et al. [19] started the parameterized research of multistage graph problems on a *global* budget where there is no restriction on the number of changes between any two consecutive layers, but instead a restriction on the total number of changes made throughout the lifetime of the instance. All graph problems studied by Heeger et al. are NP-hard even for constant values of the global budget parameter. By contrast, we will show that a global budget version of Multistage 2-Coloring is fixed-parameter tractable with respect to the budget. Formally, the global budget version of Multistage 2-Coloring is:

▶ **Problem 6.** Multistage 2-Coloring on a Global Budget (MS2CGB)
Input: A temporal graph $\mathcal{G} = (V, (E_t)_{t=1}^{\tau})$ and an integer $D \in \mathbb{N}_0$.
Question: Are there $f_1, \dots, f_\tau \colon V \to \{1, 2\}$ such that f_t is a proper 2-coloring of (V, E_t) for every $t \in \{1, \dots, \tau\}$ and $\sum_{t=1}^{\tau-1} \delta(f_t, f_{t+1}) \leq D$?

We start by pointing out that MS2CGB is NP-hard. This follows from Theorem 7, since there is no distinction between a local and a global budget if $\tau = 2$.

▶ **Observation 32.** Multistage 2-Coloring on a Global Budget *is* NP*-hard.*

In order to show that Multistage 2-Coloring on a Global Budget is fixed-parameter tractable, we will prove the existence of a parameter-preserving transformation to the Almost 2-SAT problem, which is defined by:

▶ **Problem 7.** Almost 2-SAT (A2SAT)
Input: A Boolean formula φ in 2-CNF and an integer k.
Question: Can φ be made satisfiable by removing at most k clauses?

Razgon and O'Sullivan [29] prove that A2SAT is fixed-parameter tractable when parameterized by k, but the fastest presently known algorithm runs in $\mathcal{O}^*(2.3146^k)$ and is due to Lokshtanov et al. [25]. Kratsch and Wahlström [23] show that this problem admits a randomized polynomial kernel.

▶ **Proposition 33** (★). Multistage 2-Coloring on a Global Budget *parameterized by D admits a parameter-preserving transformation to* Almost 2-SAT *parameterized by k.*

The proof is deferred to the full version. The basic idea behind the reduction is that we use $D+1$ copies of the same two clauses to express that no edge should be monochromatic. At least one of these clause pairs must survive the deletion. Moreover, we add clauses stating that vertices are not re-colored. At most D of these clauses can be deleted. This directly implies the following:

▶ **Corollary 34.** Multistage 2-Coloring on a Global Budget *parameterized by D is fixed-parameter tractable and admits a randomized polynomial kernel.*

We note that the approach described here for MS2C can be used to reduce a global budget version of the more general Multistage 2-SAT to Almost 2-SAT, proving the following:

▶ **Observation 35.** Multistage 2-SAT on a Global Budget *parameterized by the number of changes is fixed-parameter tractable and admits a randomized polynomial kernel.*

References

1 Shorouq Al-Eidi, Yuanzhu Chen, Omar Darwishand, and Ali M. S. Alfosool. Time-ordered bipartite graph for spatio-temporal social network analysis. In *Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 833–838, 2020. doi:10.1109/ICNC47757.2020.9049668.

2 Evripidis Bampis, Bruno Escoffier, and Alexander V. Kononov. LP-based algorithms for multistage minimization problems. In *Proceedings of the 18th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 1–15, 2020. doi:10.1007/978-3-030-80879-2_1.

3 Evripidis Bampis, Bruno Escoffier, Michael Lampis, and Vangelis Th. Paschos. Multistage matchings. In *Proceedings of the 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 7:1–7:13, 2018. doi:10.4230/LIPIcs.SWAT.2018.7.

4 Evripidis Bampis, Bruno Escoffier, and Alexandre Teiller. Multistage knapsack. In *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 22:1–22:14, 2019. doi:10.4230/LIPIcs.MFCS.2019.22.

5 Robert Bredereck, Till Fluschnik, and Andrzej Kaczmarczyk. Multistage committee election. *arXiv*, 2020. arXiv:2005.02300.

6 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.

7 Markus Chimani, Niklas Troost, and Tilo Wiedera. Approximating multistage matching problems. In *Proceedings of the 32nd International Workshop on Combinatorial Algorithms (IWOCA)*, pages 558–570, 2021. doi:10.1007/978-3-030-79987-8_39.

8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

9 Reinhard Diestel. *Graph Theory*. Springer, 5th edition, 2017. doi:10.1007/978-3-662-53622-3.

10 Michael Fellows, Daniel Lokshtanov, Neeldhara Misra, Matthias Mnich, Frances Rosamond, and Saket Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory of Computing Systems*, 45(4):822–848, 2009. doi:10.1007/s00224-009-9167-9.

11 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. `doi:10.1016/j.tcs.2008.09.065`.

12 Michael R. Fellows, Bart M.P. Jansen, and Frances Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. `doi:10.1016/j.ejc.2012.04.008`.

13 Till Fluschnik. A multistage view on 2-satisfiability. In *Proceedings of the 12th International Conference on Algorithms and Complexity (CIAC)*, pages 231–244, 2021. `doi:10.1007/978-3-030-75242-2_16`.

14 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. As time goes by: Reflections on treewidth for temporal graphs. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms: Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, pages 49–77. Springer International Publishing, Cham, 2020. `doi:10.1007/978-3-030-42071-0_6`.

15 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020. `doi:10.1016/j.tcs.2019.03.031`.

16 Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. Multistage vertex cover. In *Proceedings of the 14th International Symposium on Parameterized and Exact Computation (IPEC)*, pages 14:1–14:14, 2019. `doi:10.4230/LIPIcs.IPEC.2019.14`.

17 Till Fluschnik, Rolf Niedermeier, Carsten Schubert, and Philipp Zschoche. Multistage s-t path: Confronting similarity with dissimilarity in temporal graphs. In *Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC)*, pages 43:1–43:16, 2020. `doi:10.4230/LIPIcs.ISAAC.2020.43`.

18 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 563–575, 2014. `doi:10.1007/978-3-662-43948-7_47`.

19 Klaus Heeger, Anne-Sophie Himmel, Frank Kammer, Rolf Niedermeier, Malte Renken, and Andrej Sajenko. Multistage graph problems on a global budget. *Theoretical Computer Science*, 868:46–64, 2021. `doi:10.1016/j.tcs.2021.04.002`.

20 W. A. Horn. Some simple scheduling algorithms. *Naval Research Logistics Quarterly*, 21(1):177–185, 1974. `doi:10.1002/nav.3800210113`.

21 Bart M. P. Jansen. *The Power of Data Reduction: Kernels for Fundamental Graph Problems*. PhD thesis, Utrecht University, 2013. URL: `http://dspace.library.uu.nl/handle/1874/276438`.

22 Leon Kellerhals, Malte Renken, and Philipp Zschoche. Parameterized algorithms for diverse multistage problems. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA)*, pages 55:1–55:17, 2021. `doi:10.4230/LIPIcs.ESA.2021.55`.

23 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *Journal of the ACM*, 67(3), June 2020. `doi:10.1145/3390887`.

24 Matthieu Latapy, Clémence Magnien, and Tiphaine Viard. Weighted, bipartite, or directed stream graphs for the modeling of temporal networks. In Petter Holme and Jari Saramäki, editors, *Temporal Network Theory*, pages 49–64. Springer International Publishing, Cham, 2019. `doi:10.1007/978-3-030-23495-9_3`.

25 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms*, 11(2):1–31, 2014. `doi:10.1145/2566616`.

26 Hendrik Molter. *Classic Graph Problems Made Temporal: A Parameterized Complexity Analysis*. PhD thesis, Technische Universität Berlin, 2020. doi:10.14279/depositonce-10551.

27 Karolina Okrasa and Paweł Rzążewski. Subexponential algorithms for variants of the homomorphism problem in string graphs. *Journal of Computer and System Sciences*, 109:126–144, 2020. doi:10.1016/j.jcss.2019.12.004.

28 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003. doi:10.1016/S0022-0000(03)00078-3.

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

30 Róbert Sasák. Comparing 17 graph parameters. Master's thesis, University of Bergen, 2010. URL: https://bora.uib.no/bora-xmlui/handle/1956/4329.

31 Johannes Schröder. Comparing graph parameters. Bachelor's thesis, Technische Universität Berlin, 2019. URL: http://fpt.akt.tu-berlin.de/publications/theses/BA-Schröder.pdf.

32 Manuel Sorge and Mathias Weller. The graph parameter hierarchy. Unpublished manuscript, 2019. URL: https://manyu.pro/assets/parameter-hierarchy.pdf.

33 Martin Vatshelle. *New Width Parameters of Graphs*. PhD thesis, University of Bergen, 2012. URL: https://www.ii.uib.no/~martinv/Papers/MartinThesis.pdf.

34 Tsunghan Wu, Sheau-Harn Yu, Wanjiun Liao, and Cheng-Shang Chang. Temporal bipartite projection and link prediction for online social networks. In *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)*, pages 52–59, 2014. doi:10.1109/BigData.2014.7004444.

35 Mihalis Yannakakis. Node- and edge-deletion NP-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–264, 1978. doi:10.1145/800133.804355.