The Complexity of Computing Optimum Labelings for Temporal Connectivity

Nina Klobas
Department of Computer Science, Durham University, UK

George B. Mertzios
Department of Computer Science, Durham University, UK

Hendrik Molter
Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel

Paul G. Spirakis
Department of Computer Science, University of Liverpool, UK
Computer Engineering & Informatics Department, University of Patras, Greece

Abstract

A graph is temporally connected if there exists a strict temporal path, i.e., a path whose edges have *strictly increasing* labels, from every vertex u to every other vertex v. In this paper we study *temporal design* problems for undirected temporally connected graphs. The basic setting of these optimization problems is as follows: given a connected undirected graph G, what is the smallest number $|\lambda|$ of time-labels that we need to add to the edges of G such that the resulting temporal graph (G, λ) is temporally connected? As it turns out, this basic problem, called Minimum Labeling (ML), can be optimally solved in polynomial time. However, exploiting the temporal dimension, the problem becomes more interesting and meaningful in its following variations, which we investigate in this paper. First we consider the problem Min. Aged Labeling (MAL) of temporally connecting the graph when we are given an upper-bound on the allowed *age* (i.e., maximum label) of the obtained temporal graph (G, λ). Second we consider the problem Min. Steiner Labeling (MSL), where the aim is now to have a temporal path between any pair of "important" vertices which lie in a subset $R \subseteq V$, which we call the *terminals*. This relaxed problem resembles the problem Steiner Tree in static (i.e., non-temporal) graphs. However, due to the requirement of *strictly* increasing labels in a temporal path, Steiner Tree is *not* a special case of MSL. Finally we consider the age-restricted version of MSL, namely Min. Aged Steiner Labeling (MASL). Our main results are threefold: we prove that (i) MAL becomes NP-complete on undirected graphs, while (ii) MASL becomes W[1]-hard with respect to the number $|R|$ of terminals. On the other hand we prove that (iii) although the age-unrestricted problem MSL remains NP-hard, it is in FPT with respect to the number $|R|$ of terminals. That is, adding the age restriction, makes the above problems *strictly harder* (unless P=NP or W[1]=FPT).

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Mathematics of computing → Discrete mathematics

Keywords and phrases Temporal graph, graph labeling, foremost temporal path, temporal connectivity, Steiner Tree

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.62

Related Version *Full Version*: https://arxiv.org/abs/2202.05880

Funding *George B. Mertzios*: Supported by the EPSRC grant EP/P020372/1.
Hendrik Molter: Supported by the ISF, grant No. 1070/20.
Paul G. Spirakis: Supported by the NeST initiative of the School of EEE and CS at the University of Liverpool and by the EPSRC grant EP/P02002X/1.

© Nina Klobas, George B. Mertzios, Hendrik Molter, and Paul G. Spirakis;
licensed under Creative Commons License CC-BY 4.0

47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022).
Editors: Stefan Szeider, Robert Ganian, and Alexandra Silva; Article No. 62; pp. 62:1–62:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A temporal (or dynamic) graph is a graph whose underlying topology is subject to discrete changes over time. This paradigm reflects the structure and operation of a great variety of modern networks; social networks, wired or wireless networks whose links change dynamically, transportation networks, and several physical systems are only a few examples of networks that change over time [23, 33, 35]. Inspired by the foundational work of Kempe et al. [25], we adopt here a simple model for temporal graphs, in which the vertex set remains unchanged while each edge is equipped with a set of integer time-labels.

▶ **Definition 1** (temporal graph [25]). *A* temporal graph *is a pair (G, λ), where $G = (V, E)$ is an underlying (static) graph and $\lambda : E \to 2^{\mathbb{N}}$ is a* time-labeling *function which assigns to every edge of G a set of discrete time-labels.*

Here, whenever $t \in \lambda(e)$, we say that the edge e is *active* or *available* at time t. Throughout the paper we may refer to "time-labels" simply as "labels" for brevity. Furthermore, the *age* (or *lifetime*) $\alpha(G, \lambda)$ of the temporal graph (G, λ) is the largest time-label used in it, i.e., $\alpha(G, \lambda) = \max\{t \in \lambda(e) : e \in E\}$. One of the most central notions in temporal graphs is that of a *temporal path* (or *time-respecting path*) which is motivated by the fact that, due to causality, entities and information in temporal graphs can "flow" only along sequences of edges whose time-labels are strictly increasing, or at least non-decreasing.

▶ **Definition 2** (temporal path). *Let (G, λ) be a temporal graph, where $G = (V, E)$ is the underlying static graph. A* temporal path *in (G, λ) is a sequence $(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)$, where $(e_1, e_2, \ldots, e_k)$ is a path in G, $t_i \in \lambda(e_i)$ for every $i = 1, 2, \ldots, k$, and $t_1 < t_2 < \ldots < t_k$.*

A vertex v is *temporally reachable* (or *reachable*) from vertex u in (G, λ) if there exists a temporal path from u to v. If every vertex v is reachable by every other vertex u in (G, λ), then (G, λ) is called *temporally connected*. Note that, for every temporally connected temporal graph (G, λ), we have that its age is at least as large as the diameter d_G of the underlying graph G. Indeed, the largest label used in any temporal path between two anti-diametrical vertices cannot be smaller than d_G. Temporal paths have been introduced by Kempe et al. [25] for temporal graphs which have only one label per edge, i.e., $|\lambda(e)| = 1$ for every edge $e \in E$, and this notion has later been extended by Mertzios et al. [28] to temporal graphs with multiple labels per edge. Furthermore, depending on the particular application, both variations of temporal paths with non-decreasing [6, 25, 26] and with strictly increasing [15, 28] labels have been studied. In this paper we focus on temporal paths with *strictly increasing* labels. Due to the very natural use of temporal paths in various contexts, several path-related notions, such as temporal analogues of distance, diameter, reachability, exploration, and centrality have also been studied [1–3, 6, 8, 10, 11, 13, 15–18, 20, 26, 28, 32, 34, 36].

Furthermore, some non-path temporal graph problems have been recently introduced too, including for example temporal variations of maximal cliques [7, 37], vertex cover [4, 21], vertex coloring [31], matching [29], and transitive orientation [30]. Motivated by the need of restricting the spread of epidemic, Enright et al. [15] studied the problem of removing the smallest number of time-labels from a given temporal graph such that every vertex can only temporally reach a limited number of other vertices. Deligkas et al. [12] studied the problem of accelerating the spread of information for a set of sources to all vertices in a temporal graph, by only using delaying operations, i.e., by shifting specific time-labels to a later time slot. The problems studied in [12] are related but orthogonal to our temporal connectivity problems. Various other temporal graph modification problems have been also studied, see for example [6, 11, 13, 16, 34].

The time-labels of an edge e in a temporal graph indicate the discrete units of time (e.g., days, hours, or even seconds) in which e is active. However, in many real dynamic systems, e.g., in synchronous mobile distributed systems that operate in discrete rounds, or in unstable chemical or physical structures, maintaining an edge over time requires energy and thus comes at a cost. One natural way to define the *cost* of the whole temporal graph (G, λ) is the *total number* of time-labels used in it, i.e., the total cost of (G, λ) is $|\lambda| = \sum_{e \in E} |\lambda_e|$.

In this paper we study *temporal design* problems of undirected temporally connected graphs. The basic setting of these optimization problems is as follows: given an undirected graph G, what is the smallest number $|\lambda|$ of time-labels that we need to add to the edges of G such that (G, λ) is temporally connected? As it turns out, this basic problem can be optimally solved in polynomial time, thus answering to a conjecture made in [2]. However, exploiting the temporal dimension, the problem becomes more interesting and meaningful in its following variations, which we investigate in this paper. First we consider the problem variation where we are given along with the input also an upper bound of the allowed *age* (i.e., maximum label) of the obtained temporal graph (G, λ). This age restriction is sensible in more pragmatic cases, where delaying the latest arrival time of any temporal path incurs further costs, e.g., when we demand that all agents in a safety-critical distributed network are synchronized as quickly as possible, and with the smallest possible number of communications among them. Second we consider problem variations where the aim is to have a temporal path between any pair of "important" vertices which lie in a subset $R \subseteq V$, which we call the *terminals*. For a detailed definition of our problems we refer to Section 2.

Here it is worth noting that the latter relaxation of temporal connectivity resembles the problem Steiner Tree in static (i.e., non-temporal) graphs. Given a connected graph $G = (V, E)$ and a set $R \subseteq V$ of terminals, Steiner Tree asks for a smallest-sized subgraph of G which connects all terminals in R. Clearly, the smallest subgraph sought by Steiner Tree is a tree. As it turns out, this property does not carry over to the temporal case. Consider for example an arbitrary graph G and a terminal set $R = \{a, b, c, d\}$ such that G contains an induced cycle on four vertices a, b, c, d; that is, G contains the edges ab, bc, cd, da but not the edges ac or bd. Then, it is not hard to check that only way to add the smallest number of time-labels such that all vertices of R are temporally connected is to assign one label to each edge of the cycle on a, b, c, d, e.g., $\lambda(ab) = \lambda(cd) = 1$ and $\lambda(bc) = \lambda(cd) = 2$. The main underlying reason for this difference with the static problem Steiner Tree is that temporal connectivity is *not transitive* and *not symmetric:* if there exists temporal paths from u to v, and from v to w, it is not a priori guaranteed that a temporal path from v to u, or from u to w exists.

Temporal network design problems have already been considered in previous works. Mertzios et al. [28] proved that it is APX-hard to compute a minimum-cost labeling for temporally connecting an input *directed* graph G, where the age of the graph is upper-bounded by the diameter of G. This hardness reduction was strongly facilitated by the careful placement of the edge directions in the constructed instance, in which every vertex was reachable in the static graph by only constantly many vertices. Unfortunately this cannot happen in an undirected connected graph, where every vertex is reachable by all other vertices. Later, Akrida et al. [2] proved that it is also APX-hard to *remove* the largest number of time-labels from a given temporally connected (undirected) graph (G, λ), while still maintaining temporal connectivity. In this case, although there are no edge directions, the hardness reduction was strongly facilitated by the careful placement of the initial time-labels of λ in the input temporal graph, in which every pair of vertices could be connected by only a few different temporal paths, among which the solution had to choose. Unfortunately this cannot happen when the goal is to add time-labels to an undirected connected graph, where there are potentially multiple ways to temporally connect a pair of vertices (even if we upper-bound the largest time-label by the diameter).

Summarizing, the above technical difficulties seem to be the reason why the problem of *adding* the minimum number of time-labels with an age-restriction to an *undirected* graph to achieve temporal connectivity remained open until now for the last decade. In this paper we overcome these difficulties by developing a hardness reduction from a variation of the problem Max XOR SAT (see Theorem 12 in Section 3) where we manage to add the appropriate (undirected) edges among the variable-gadgets such that simultaneously (i) the distance between any two vertices from different variable gadgets remains small (constant) and (ii) there is no shortest path between two vertices of the *same* variable gadget that leaves this gadget.

Our contribution and road-map. In the first part of our paper, in Section 3, we present our results on Min. Aged Labeling (MAL). This problem is the same as ML, with the additional restriction that we are given along with the input an upper bound on the allowed *age* of the resulting temporal graph (G, λ). Using a technically involved reduction from a variation of Max XOR SAT, we prove that MAL is NP-complete on undirected graphs, even when the required maximum age is equal to the diameter d_G of the input static graph G.

In the second part of our paper, in Section 4, we present our results on the Steiner-tree versions of the problem, namely on Min. Steiner Labeling (MSL) and Min. Aged Steiner Labeling (MASL). The difference of MSL from ML is that, here, the goal is to have a temporal path between any pair of "important" vertices which lie in a given subset $R \subseteq V$ (the *terminals*). In Section 4.1 we prove that MSL is NP-complete by a reduction from Vertex Cover, the correctness of which requires showing structural properties of MSL. Here it is worth recalling that, as explained above, the classical problem Steiner Tree on static graphs is *not* a special case of MSL, due to the requirement of strictly increasing labels in a temporal path. Furthermore, we would like to emphasize here that, as temporal connectivity is neither transitive nor symmetric, a straightforward NP-hardness reduction from Steiner Tree to MSL does not seem to exist. For example, as explained above, in a graph that contains a C_4 with its four vertices as terminals, labeling a Steiner tree is sub-optimal for MSL.

In Section 4.2 we provide a fixed-parameter tractable (FPT) algorithm for MSL with respect to the number $|R|$ of terminal vertices, by providing a parameterized reduction to Steiner Tree. The proof of correctness of our reduction, which is technically quite involved, is of independent interest, as it proves crucial graph-theoretical properties of minimum temporal Steiner labelings. In particular, for our algorithm we prove (see Lemma 14) that, for any undirected graph G with a set R of terminals, there always exists at least one minimum temporal Steiner labeling (G, λ) which labels edges either from (i) a tree or from (ii) a tree with one extra edge that builds a C_4.

In Section 4.3 we prove that MASL is W[1]-hard with respect to the number $|R|$ of terminals. Our results actually imply the stronger statement that MASL is W[1]-hard even with respect to the number of time-labels of the solution (which is a larger parameter than the number $|R|$ of terminals).

Finally, we complete the picture by providing some auxiliary results in our preliminary Section 2. More specifically, in Section 2.1 we prove that ML can be solved in polynomial time, and in Section 2.2 we prove that the analogue minimization versions of ML and MAL on directed acyclic graphs are solvable in polynomial time.

Due to space constraints, proofs of results marked with $\star$ are (partially) deferred to a full version on arXiv [27].

2 Preliminaries and notation

Given a (static) undirected graph $G = (V, E)$, an edge between two vertices $u, v \in V$ is denoted by uv, and in this case the vertices u, v are said to be *adjacent* in G. If the graph is directed, we will use the ordered pair (u, v) (resp. (v, u)) to denote the oriented edge from u to v (resp. from v to u). The *age* of a temporal graph (G, λ) is denoted by $\alpha(G, \lambda) = \max\{t \in \lambda(e) : e \in E\}$. A temporal path $(e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)$ from vertex u to vertex v is called *foremost*, if it has the smallest arrival time t_k among all temporal paths from u to v. Note that there might be another temporal path from u to v that uses fewer edges than a foremost path. A temporal graph (G, λ) is *temporally connected* if, for every pair of vertices $u, v \in V$, there exists a temporal path (see Definition 2) P_1 from u to v and a temporal path P_2 from v to u. Furthermore, given a set of terminals $R \subseteq V$, the temporal graph (G, λ) is *R-temporally connected* if, for every pair of vertices $u, v \in R$, there exists a temporal path from u to v and a temporal path from v to u; note that P_1 and P_2 can also contain vertices from $V \setminus R$. Now we provide our formal definitions of our four decision problems.

Min. Labeling (ML)

Input: A static graph $G = (V, E)$ and a $k \in \mathbb{N}$.
Question: Does there exist a temporally connected temporal graph (G, λ), where $|\lambda| \leq k$?

Min. Aged Labeling (MAL)

Input: A static graph $G = (V, E)$ and two integers $a, k \in \mathbb{N}$.
Question: Does there exist a temporally connected temporal graph (G, λ), where $|\lambda| \leq k$ and $\alpha(\lambda) \leq a$?

Min. Steiner Labeling (MSL)

Input: A static graph $G = (V, E)$, a subset $R \subseteq V$ and a $k \in \mathbb{N}$.
Question: Does there exist a temporally R-connected temporal graph (G, λ), where $|\lambda| \leq k$?

Min. Aged Steiner Labeling (MASL)

Input: A static graph $G = (V, E)$, a subset $R \subseteq V$, and two integers $a, k \in \mathbb{N}$.
Question: Does there exist a temporally R-connected temporal graph (G, λ), where $|\lambda| \leq k$ and $\alpha(\lambda) \leq a$?

Note that, for both problems MAL and MASL, whenever the input age bound a is strictly smaller than the diameter d of G, the answer is always NO. Thus, we always assume in the remainder of the paper that $a \geq d$, where d is the diameter of the input graph G. For simplicity of the presentation, we denote next by $\kappa(G, d)$ the smallest number k for which (G, k, d) is a YES instance for MAL.

▶ **Observation 3** ($\star$). *For every graph G with n vertices and diameter d, we have that $\kappa(G, d) \leq n(n-1)$.*

The next lemma shows that the upper bound of Observation 3 is asymptotically tight as, for cycle graphs C_n with diameter d, we have that $\kappa(C_n, d) = \Theta(n^2)$.

▶ **Lemma 4** ($\star$). *Let C_n be a cycle on n vertices, where $n \neq 4$, and let d be its diameter. Then*

$$\kappa(C_n, d) = \begin{cases} d^2, & \text{when } n = 2d \\ 2d^2 + d, & \text{when } n = 2d + 1. \end{cases}$$

2.1 A polynomial-time algorithm for ML

As a first warm-up, we study the problem ML, where no restriction is imposed on the maximum allowed age of the output temporal graph. It is already known by Akrida et al. [2] that any undirected graph can be made temporally connected by adding at most $2n-3$ time-labels, while for trees $2n-3$ labels are also necessary. Moreover, it was conjectured that every graph needs at least $2n-4$ time-labels [2]. Here we prove their conjecture true by proving that, if G contains (resp. does not contain) the cycle C_4 on four vertices as a subgraph, then (G,k) is a YES instance of ML if and only if $k \geq 2n-4$ (resp. $k \geq 2n-3$). The proof is done via a reduction to the gossip problem [9] (for a survey on gossiping see also [22]).

The related problem of achieving temporal connectivity by assigning to every edge of the graph at most one time-label, has been studied by Göbel et al. [19], where the relationship with the gossip problem has also been drawn. Contrary to ML, this problem is NP-hard [19]. That is, the possibility of assigning two or more labels to an edge makes the problem computationally much easier. Indeed, in a C_4-free graph with n vertices, an optimal solution to ML consists in assigning in total $2n-3$ time-labels to the $n-1$ edges of a spanning tree. In such a solution, one of these $n-1$ edges receives one time-label, while each of the remaining $n-2$ edges receives two time-labels. Similarly, when the graph contains a C_4, it suffices to span the graph with four trees tooted at the vertices of the C_4, where each of the edges of the C_4 receives one time-label and each edge of the four trees receives two labels. That is, a graph containing a C_4 can be temporally connected using $2n-4$ time-labels.

In the gossip problem we have n agents from a set A. At the beginning, every agent $x \in A$ holds its own secret. The goal is that each agent eventually learns the secret of every other agent. This is done by producing a sequence of unordered pairs (x,y), where $x, y \in A$ and each such pair represents one phone call between the agents involved, during which the two agents exchange all the secrets they currently know.

The above gossip problem is naturally connected to ML. The only difference between the two problems is that, in gossip, all calls are non-concurrent, while in ML we allow concurrent temporal edges, i.e., two or more edges can appear at the same time slot t. Therefore, in order to transfer the known results from gossip to ML, it suffices to prove that in ML we can equivalently consider solutions with non-concurrent edges.

▶ **Theorem 5** ($\star$). *Let $G=(V,E)$ be a connected graph. Then the smallest $k \in \mathbb{N}$ for which (G,k) is a YES instance of ML is:*

$$k = \begin{cases} 2n-4, & \textit{if } G \textit{ contains } C_4 \textit{ as a subgraph,} \\ 2n-3, & \textit{otherwise.} \end{cases}$$

2.2 A polynomial-time algorithm for directed acyclic graphs

As a second warm-up, we show that the minimization analogues of ML and MAL on directed acyclic graphs (DAGs) are solvable in polynomial time. More specifically, for the minimization analogue of ML we provide an algorithm which, given a DAG $G=(V,A)$ with diameter d_G, computes a temporal labeling function λ which assigns the smallest possible number of time-labels on the arcs of G with the following property: for every two vertices $u, v \in V$, there exists a directed temporal path from u to v in (G,λ) if and only if there exists a directed path from u to v in G. Moreover, the age $\alpha(G,\lambda)$ of the resulting temporal graph is *equal* to d_G. Therefore, this immediately implies a polynomial-time algorithm for the minimization analogue of MAL on DAGs. For notation uniformity, we call these minimization problems $\text{ML}_{directed}$ and $\text{MAL}_{directed}$, respectively.

▶ **Theorem 6** (⋆). *Let $G = (V, E)$ be a DAG with n vertices and m arcs. Then $ML_{directed}(G)$ and $MAL_{directed}(G)$ can be both computed in $O(n(n + m))$ time.*

3 MAL is NP-complete

In this section we prove that it is NP-hard to determine the number of labels in an optimal labeling of a static, undirected graph G, where the age, i.e., the maximum label used, is not larger than the diameter of the input graph.

To prove this we provide a reduction from the NP-hard problem Monotone Max XOR(3) (or MonMaxXOR(3) for short). This is a special case of the classical Boolean satisfiability problem, where the input formula ϕ consists of the conjunction of *monotone* XOR clauses of the form $(x_i \oplus x_j)$, i.e., variables x_i, x_j are non-negated. If each variable appears in exactly r clauses, then ϕ is called a *monotone* Max XOR(r) formula. A clause $(x_i \oplus x_j)$ is *XOR-satisfied* (or simply *satisfied*) if and only if $x_i \neq x_j$. In Monotone Max XOR(r) we are trying to find a truth assignment τ of ϕ which satisfies the maximum number of clauses. As it can be easily checked, MonMaxXOR(3) encodes the problem Max-Cut on cubic graphs, which is known to be NP-hard [5]. Therefore we conclude the following.

▶ **Theorem 7** ([5]). *MonMaxXOR(3) is NP-hard.*

Now we explain our reduction from MonMaxXOR(3) to the problem Minimum Aged Labeling (MAL), where the input static graph G is undirected and the desired age of the output temporal graph is the diameter d of G . Let ϕ be a monotone Max XOR(3) formula with n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$. Note that $m = \frac{3}{2}n$, since each variable appears in exactly 3 clauses. From ϕ we construct a static undirected graph G_ϕ with diameter $d = 10$, and prove that there exists a truth assignment τ which satisfies at least k clauses in ϕ, if and only if there exists a labeling λ_ϕ of G_ϕ, with $|\lambda_\phi| \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$ labels and with age $\alpha(G, \lambda) \leq 10$.

High-level construction

For each variable x_i, $1 \leq i \leq n$, we construct a variable gadget X_i that consists of a "starting" vertex s_i and three "ending" vertices t_i^ℓ (for $\ell \in \{1, 2, 3\}$); these ending vertices correspond to the appearances of x_i in three clauses of ϕ. In an optimum labeling $\lambda(\phi)$, in each variable gadget there are exactly two labelings that temporally connect starting and ending vertices, which correspond to the True or False truth assignment of the variable in the input formula ϕ. For every clause $(x_i \oplus x_j)$ we identifying corresponding ending vertices of X_i and X_j (as well as some other auxiliary vertices and edges). Whenever $(x_i \oplus x_j)$ is satisfied by a truth assignment of ϕ, the labels of the common edges of X_i and X_j in an optimum labeling coincide (thus using few labels); otherwise we need additional labels for the common edges of X_i and X_j.

Detailed construction of G_ϕ

For each variable x_i from ϕ we create a variable gadget X_i, that consists of a *base* BX_i on 11 vertices, $BX_i = \{s_i, a_i, b_i, c_i, d_i, e_i, \overline{a_i}, \overline{b_i}, \overline{c_i}, \overline{d_i}, \overline{e_i}\}$, and three *forks* F^1X_i, F^2X_i, F^3X_i, each on 9 vertices, $F^\ell X_i = \{t_i^\ell, f_i^\ell, g_i^\ell, h_i^\ell, m_i^\ell, \overline{f_i}^{\ell}, \overline{g_i}^{\ell}, \overline{h_i}^{\ell}, \overline{m_i}^{\ell}\}$, where $\ell \in \{1, 2, 3\}$. Vertices in the base BX_i are connected in the following way: there are two paths of length 5: $s_i a_i b_i c_i d_i e_i$ and $s_i \overline{a_i}\,\overline{b_i}\,\overline{c_i}\,\overline{d_i}\,\overline{e_i}$, and 5 extra edges of form $y_i \overline{y_i}$, where $y \in \{a, b, c, d, e\}$. Vertices in each fork $F^\ell X_i$ (where $\ell \in \{1, 2, 3\}$) are connected in the following way: there are two paths of length

4: $t_i^\ell m_i^\ell h_i^\ell g_i^\ell f_i^\ell$ and $t_i^\ell \overline{m_i}^\ell \overline{h_i}^\ell \overline{g_i}^\ell \overline{f_i}^\ell$, and 4 extra edges of form $y_i\overline{y_i}^\ell$, where $y \in \{m, h, g, f\}$. The base BX_i of the variable gadget X_i is connected to each of the three forks $F^\ell X_i$ via two edges $e_i f_i^\ell$ and $\overline{e_i}\,\overline{f_i}^\ell$, where $\ell \in \{1, 2, 3\}$. For an illustration see Figure 1.

For an easier analysis we fix the following notation. The vertex $s_i \in BX_i$ is called a *start vertex* of X_i, vertices t_i^ℓ ($\ell \in \{1, 2, 3\}$) are called *ending vertices* of X_i, a path connecting s_i, t_i^ℓ that passes through vertices $a_i b_i c_i d_i e_i f_i^\ell g_i^\ell h_i^\ell m_i^\ell$ (resp. $\overline{a_i}\,\overline{b_i} \ldots \overline{m_i}^\ell$) is called the *left* (resp. *right*) s_i, t_i^ℓ-path. The left (resp. right) s_i, t_i^ℓ-path is a disjoint union of the left (resp. right) path on vertices of the base BX_i of X_i, an edge of form $e_i f_i^\ell$ (resp. $\overline{e_i}\,\overline{f_i}^\ell$) called the left (resp. right) *bridge edge* and the left (resp. right) path on vertices of the ℓ-th fork $F^\ell X_i$ of X_i. The edges $y_i\overline{y_i}$, where $y \in \{a, b, c, d, e, f^\ell, g^\ell, h^\ell, m^\ell\}$, $\ell \in \{1, 2, 3\}$, are called *connecting edges.*

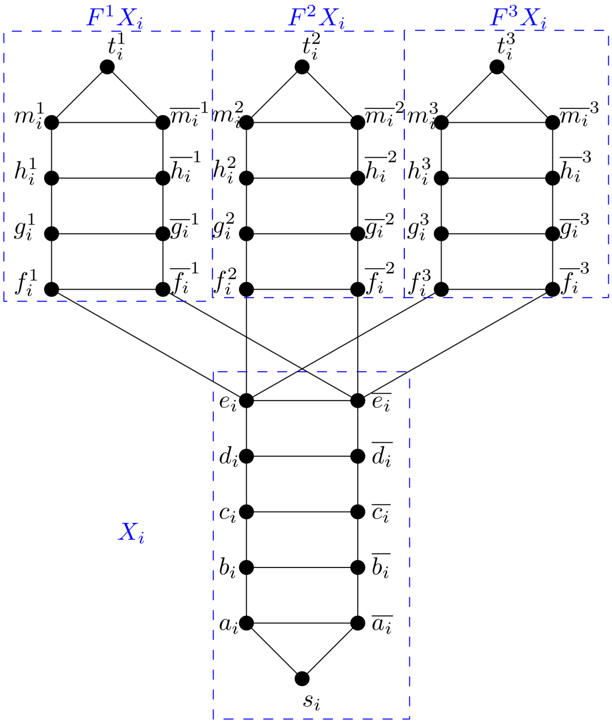

Figure 1 An example of a variable gadget X_i in G_ϕ, corresponding to the variable x_i from ϕ.

Connecting variable gadgets

There are two ways in which we connect two variable gadgets, depending whether they appear in the same clause in ϕ or not.

1. Two variables x_i, x_j do not appear in any clause together. In this case we add the following edges between the variable gadgets X_i and X_j:
   - from e_i (resp. $\overline{e_i}$) to $f_j^{\ell'}$ and $\overline{f_j}^{\ell'}$, where $\ell' \in \{1, 2, 3\}$,
   - from e_j (resp. $\overline{e_j}$) to f_i^ℓ and $\overline{f_i}^\ell$, where $\ell \in \{1, 2, 3\}$,
   - from d_i (resp. $\overline{d_i}$) to d_j and $\overline{d_j}$.

   We call these edges the *variable edges.* For an illustration see Figure 2.

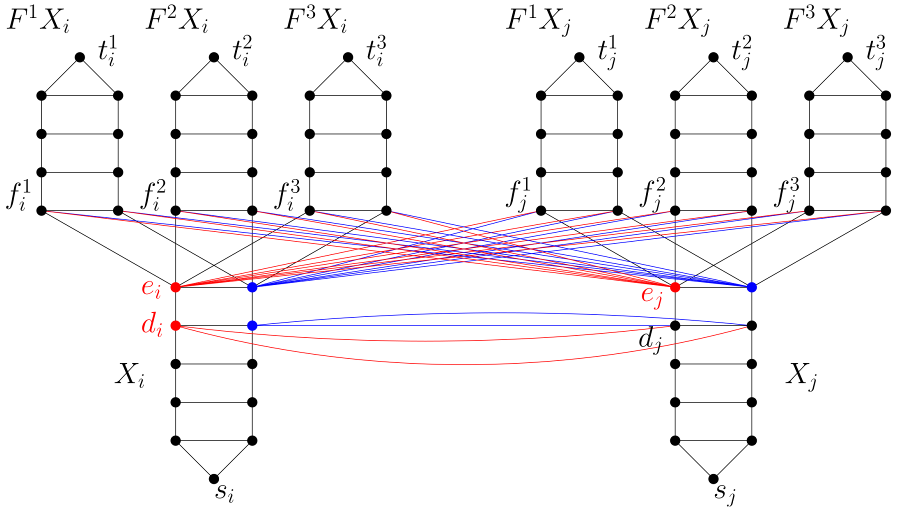

Figure 2 An example of two non-intersecting variable gadgets and variable edges among them.

2. Let $C = (x_i \oplus x_j)$ be a clause of ϕ, that contains the r-th appearance of the variable x_i and r'-th appearance of the variable x_j. In this case we identify the r-th fork $F^r X_i$ of X_i with the r'-th fork $F^{r'} X_j$ of X_j in the following way:
   - $t_i^r = t_j^{r'}$,
   - $\{f_i^r, g_i^r, h_i^r, m_i^r\} = \{\overline{f_j}^{r'}, \overline{g_j}^{r'}, \overline{h_j}^{r'}, \overline{m_j}^{r'}\}$ respectively, and
   - $\{\overline{f_i}^r, \overline{g_i}^r, \overline{h_i}^r, \overline{m_i}^r\} = \{f_j^{r'}, g_j^{r'}, h_j^{r'}, m_j^{r'}\}$ respectively.

   Besides that we add the following edges between the variable gadgets X_i and X_j:
   - from e_i (resp. $\overline{e_i}$) to $f_j^{\ell'}$ and $\overline{f_j}^{\ell'}$, where $\ell' \in \{1, 2, 3\} \setminus \{r'\}$,
   - from e_j (resp. $\overline{e_j}$) to f_i^{ℓ} and $\overline{f_i}^{\ell}$, where $\ell \in \{1, 2, 3\} \setminus \{r\}$,
   - from d_i (resp. $\overline{d_i}$) to d_j and $\overline{d_j}$.

   For an illustration see Figure 3.

This finishes the construction of G_ϕ. Before continuing with the reduction, we prove the following structural property of G_ϕ.

▶ **Lemma 8** (⋆). *The diameter d_ϕ of G_ϕ is* 10.

▶ **Theorem 9** (⋆). *If $OPT_{\text{MonMaxXOR(3)}}(\phi) \geq k$ then $OPT_{MAL}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$, where n is the number of variables in the formula ϕ.*

Before proving the statement in the other direction, we have to show some structural properties. Let us fix the following notation. If a labeling λ_ϕ labels all left (resp. right) paths of the variable gadget X_i (i.e., both bottom-up from s_i to t_i^1, t_i^2, t_i^3 and top-down from t_i^1, t_i^2, t_i^3 to s_i with labels $1, 2 \ldots, 10$ in this order), then we say that the variable gadget X_i is *left-aligned* (resp. *right-aligned*) in the labeling λ_ϕ. Note, if at least one edge on any of these left (resp. right) paths of X_i is not labeled with the appropriate label between 1 and 10, then the variable gadget is *not* left-aligned (resp. *not* right-aligned). Every temporal path from s_i to t_i^ℓ (resp. from t_i^ℓ to s_i) of length 10 in X_i is called an *upward path* (resp. a *downward path*) in X_i. Any part of an upward (resp. downward) path is called a *partial* upward (resp. downward) path. Note that, for any $\ell, \ell' \in \{1, 2, 3\}$, $\ell \neq \ell'$, a temporal path from t_i^ℓ to $t_i^{\ell'}$ of length 10 is the union of a partial downward path on the fork F_i^ℓ and a partial upward path on $F_i^{\ell'}$. Moreover, note that these two partial downward/upward paths

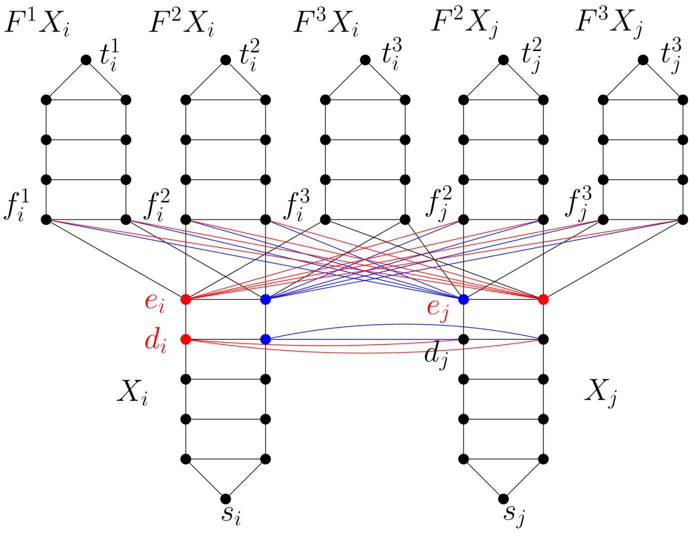

Figure 3 An example of two intersecting variable gadgets X_i, X_j corresponding to variables x_i, x_j, that appear together in some clause in ϕ, where it is the third appearance of x_i and the first appearance of x_j.

must be either both parts of a left temporal path or both parts of a right temporal path between s_i and $t_i^\ell, t_i^{\ell'}$. The following technical lemma will allow us to prove the correctness of our reduction.

▶ **Lemma 10** (⋆). *Let λ_ϕ be a minimum labeling of G_ϕ. Then λ_ϕ can be modified in polynomial time to a minimum labeling of G_ϕ in which each variable gadget X_i is either left-aligned or right-aligned.*

▶ **Theorem 11** (⋆). *If $OPT_{\text{MAL}}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$ then $OPT_{\text{MonMaxXOR(3)}}(\phi) \geq k$, where n is the number of variables in the formula ϕ.*

Since MAL is clearly in NP, the next theorem follows directly by Theorems 7, 9, and 11.

▶ **Theorem 12.** *MAL is NP-complete on undirected graphs, even when the required maximum age is equal to the diameter of the input graph.*

4 The Steiner-Tree variations of the problem

In this section we investigate the computational complexity of the Steiner-Tree variations of the problem, namely MSL and MASL. First, we prove in Section 4.1 that the age-unrestricted problem MSL remains NP-hard, using a reduction from Vertex Cover. In Section 4.2 we prove that this problem is in FPT, when parameterized by the number $|R|$ of terminals. Finally, using a parameterized reduction from Multicolored Clique, we prove in Section 4.3 that the age-restricted version MASL is W[1]-hard with respect to $|R|$, even if the maximum allowed age is a constant.

4.1 MSL is NP-complete

▶ **Theorem 13** (⋆). *MSL is NP-complete.*

Proof sketch. MSL is clearly contained in NP. To prove that the MSL is NP-hard we provide a polynomial-time reduction from the NP-complete Vertex Cover problem [24].

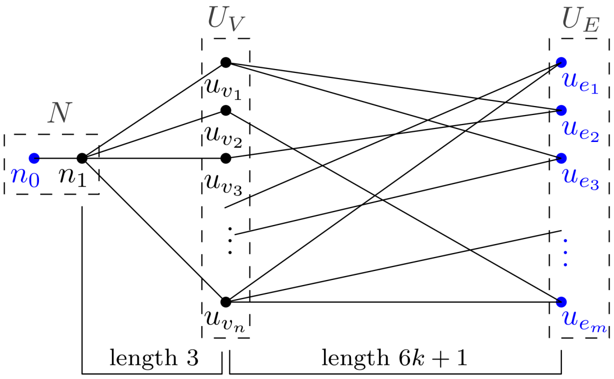

Figure 4 An example of construction of the input graph for MSL.

Vertex Cover
Input: A static graph $G = (V, E)$, a positive integer k.
Question: Does there exist a subset of vertices $S \subseteq V$ such that $|S| = k$ and $\forall e \in E, e \cap S \neq \emptyset$.

Let (G, k) be an input of the Vertex Cover problem and denote $|V(G)| = n, |E(G)| = m$. We assume w.l.o.g. that G does not admit a vertex cover of size $k - 1$. We construct (G^*, R^*, k^*), the input of MSL using the following procedure. The vertex set $V(G^*)$ consists of the following vertices:

- two starting vertices $N = \{n_0, n_1\}$,
- a "vertex-vertex" corresponding to every vertex of G: $U_V = \{u_v | v \in V(G)\}$,
- an "edge-vertex" corresponding to every edge of G: $U_E = \{u_e | e \in E(G)\}$,
- $2n + 12m \cdot k$ "dummy" vertices.

The edge set $E(G^*)$ consists of the following edges:

- an edge between starting vertices, i.e., n_0n_1,
- a path of length 3 between a starting vertex n_1 and every vertex-vertex $u_v \in U_V$ using 2 dummy vertices, and
- for every edge $e = vw \in E(G)$ we connect the corresponding edge-vertex u_e with the vertex-vertices u_v and u_w, each with a path of length $6k + 1$ using $6k$ dummy vertices.

We set $R^* = \{n_0\} \cup U_E$ and $k^* = 6k + 2m(6k + 1) + 1$. This finishes the construction. It is not hard to see that this construction can be performed in polynomial time. For an illustration see Figure 4. Note that any two paths in G^* can intersect only in vertices from $N \cup U_V \cup U_E$ and not in any of the dummy vertices. At the end G^* is a graph with $3n + m(12k + 1) + 2$ vertices and $1 + 3n + 2m(6k + 1)$ edges.

In the full proof we prove that (G, k) is a YES instance of the Vertex Cover if and only if (G^*, R^*, k^*) is a YES instance of the MSL. ◀

4.2 An FPT-algorithm for MSL with respect to the number of terminals

In this section we provide an FPT-algorithm for MSL, parameterized by the number $|R|$ of terminals. The algorithm is based on a crucial structural property of minimum solutions for MSL: there always exists a minimum labeling λ that labels the edges of a subtree of the input graph (where every leaf is a terminal vertex), and potentially one further edge that forms a C_4 with three edges of the subtree.

Intuitively speaking, we can use an FPT-algorithm for Steiner Tree parameterized by the number of terminals [14] to reveal a subgraph of the MSL instance that we can optimally label using Theorem 5. Since the number of terminals in the created Steiner Tree instance is larger than the number of terminals in the MSL instance by at most a constant, we obtain an FPT-algorithm for MSL parameterized by the number of terminals.

▶ **Lemma 14** ($\star$). *Let $G = (V, E)$ be a graph, $R \subseteq V$ a set of terminals, and k be an integer such that (G, R, k) is a YES instance of MSL and $(G, R, k-1)$ is a NO instance of MSL.*

- *If k is odd, then there is a labeling λ of size k for G such that the edges labeled by λ form a tree, and every leaf of this tree is a vertex in R.*
- *If k is even, then there is a labeling λ of size k for G such that the edges labeled by λ form a graph that is a tree with one additional edge that forms a C_4, and every leaf of the tree is a vertex in R.*

The main idea for the proof of Lemma 14 is as follows. Given a solution labeling λ, we fix one terminal r^* and then (i) we consider the minimum subtree in which r^* can reach all other terminal vertices and (ii) we consider the minimum subtree in which all other terminal vertices can reach r^*. Intuitively speaking, we want to label the smaller one of those subtrees using Theorem 5 and potentially adding an extra edge to form a C_4; we then argue that the obtained labeling does not use more labels than λ. To do that, and to detect whether it is possible to add an edge to create a C_4, we make a number of modifications to the trees until we reach a point where we can show that our solution is correct.

Having Lemma 14, we can now give our algorithm for MSL. As mentioned before, it uses an FPT-algorithm for Steiner Tree parameterized by the number of terminals [14] as a subroutine.

▶ **Theorem 15** ($\star$). *MSL is in FPT when parameterized by the number of terminals.*

4.3 Parameterized Hardness of MASL

Note that, since MASL generalizes both MSL and MAL, NP-hardness of MASL is already implied by both Theorems 12 and 13. In this section, we prove that MASL is W[1]-hard when parameterized by the number $|R|$ of the terminals, even if the restriction a on the age is a constant. To this end, we provide a parameterized reduction from Multicolored Clique. This, together with Theorem 15, implies that MASL is strictly harder than MSL (parameterized by the number $|R|$ of terminals), unless FPT=W[1].

▶ **Theorem 16** ($\star$). *MASL is W[1]-hard when parameterized by the number $|R|$ of the terminals, even if the restriction a on the age is a constant.*

Note here that, in the constructed instance of MASL in the proof of Theorem 16, the number of labels is also upper-bounded by a function of the number of colors in the instance of Multicolored Clique. Therefore the proof of Theorem 16 implies also the next result, which is even stronger (since in every solution of MASL the number of time-labels is lower-bounded by a function of the number $|R|$ of terminals).

▶ **Corollary 17.** *MASL is W[1]-hard when parameterized by the number k of time-labels, even if the restriction a on the age is a constant.*

5 Concluding remarks

Several open questions arise from our results. As we pointed out in Lemma 4, $\kappa(C_n, d) = \Theta(n^2)$, while $\kappa(G, d) = O(n^2)$ for every graph G by Observation 3. For which graph classes $\mathcal{G}$ do we have $\kappa(G, d) = o(n^2)$ (resp. $\kappa(G, d) = O(n)$) for every $G \in \mathcal{G}$?

As we proved in Theorem 12, MAL is NP-complete when the upper age bound is equal to the diameter d of the input graph G. In other words, it is NP-hard to compute $\kappa(G, d)$. On the other hand, $\kappa(G, 2r)$ can be easily computed in polynomial time, where r is the *radius* of G. Indeed, using the results of Section 2.1, it easily follows that, if G contains (resp. does not contain) a C_4 then $\kappa(G, 2r) = 2n - 4$ (resp. $\kappa(G, 2r) = 2n - 3$). For which values of an upper age bound a, where $d \leq a \leq 2r$, can $\kappa(G, a)$ computed efficiently? In particular, can $\kappa(G, d + 1)$ or $\kappa(G, 2r - 1)$ be computed in polynomial time for every undirected graph G?

With respect to parameterized algorithmics, is MAL FPT with respect to the number k of time-labels?

References

1 Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. Ephemeral networks with random availability of links: The case of fast networks. *Journal of Parallel and Distributed Computing*, 87:109–120, 2016.

2 Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017.

3 Eleni C. Akrida, George B. Mertzios, Sotiris E. Nikoletseas, Christoforos L. Raptopoulos, Paul G. Spirakis, and Viktor Zamaraev. How fast can we reach a target vertex in stochastic temporal graphs? In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 132, pages 131:1–131:14, 2019.

4 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 148:1–148:14, 2018.

5 Paola Alimonti and Viggo Kann. Hardness of approximating problems on cubic graphs. In *Proceedings of the 3rd Italian Conference on Algorithms and Complexity (CIAC)*, pages 288–298, 1997.

6 Kyriakos Axiotis and Dimitris Fotakis. On the size and the approximability of minimum temporally connected subgraphs. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 149:1–149:14, 2016.

7 Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, and René Saitenmacher. Listing all maximal k-plexes in temporal graphs. *ACM Journal of Experimental Algorithmics*, 24(1):13:1–13:27, 2019.

8 Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285, 2003.

9 Richard T. Bumby. A problem with telephones. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):13–18, 1981.

10 Sebastian Buß, Hendrik Molter, Rolf Niedermeier, and Maciej Rymar. Algorithmic aspects of temporal betweenness. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2084–2092, 2020.

11 Arnaud Casteigts, Joseph G. Peters, and Jason Schoeters. Temporal cliques admit sparse spanners. *Journal of Computer and System Sciences*, 121:1–17, 2021.

12 Argyrios Deligkas, Eduard Eiben, and George Skretas. Minimizing reachability times on temporal graphs via shifting labels. *CoRR*, abs/2112.08797, 2021. arXiv:2112.08797.

13 Argyrios Deligkas and Igor Potapov. Optimizing reachability sets in temporal graphs by delaying. In *Proceedings of the 34th Conference on Artificial Intelligence (AAAI)*, pages 9810–9817, 2020.

14 S.E. Dreyfus and R.A. Wagner. The steiner problem in graphs. *Networks*, 1:195–207, 1971.

15 Jessica Enright, Kitty Meeks, George B. Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. *Journal of Computer and System Sciences*, 119:60–77, 2021.

16 Jessica Enright, Kitty Meeks, and Fiona Skerman. Assigning times to minimise reachability in temporal graphs. *Journal of Computer and System Sciences*, 115:169–186, 2021.

17 Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On temporal graph exploration. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 444–455, 2015.

18 Thomas Erlebach and Jakob T. Spooner. Faster exploration of degree-bounded temporal graphs. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 36:1–36:13, 2018.

19 F. Göbel, J.Orestes Cerdeira, and H.J. Veldman. Label-connected graphs and the gossip problem. *Discrete Mathematics*, 87(1):29–40, 1991.

20 Roman Haag, Hendrik Molter, Rolf Niedermeier, and Malte Renken. Feedback edge sets in temporal graphs. *Discrete Applied Mathematics*, 307:65–78, 2022.

21 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Proceedings of the 36th Conference on Artificial Intelligence (AAAI)*, 2022. To appear.

22 Sandra M. Hedetniemi, Stephen T. Hedetniemi, and Arthur L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

23 Petter Holme and Jari Saramäki. *Temporal network theory*, volume 2. Springer, 2019.

24 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

25 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.

26 Nina Klobas, George B. Mertzios, Hendrik Molter, Rolf Niedermeier, and Philipp Zschoche. Interference-free walks in time: Temporally disjoint paths. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4090–4096, 2021.

27 Nina Klobas, George B. Mertzios, Hendrik Molter, and Paul G. Spirakis. The complexity of computing optimum labelings for temporal connectivity. *CoRR*, abs/2202.0588, 2022. `arXiv:2202.0588`.

28 George B. Mertzios, Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. Temporal network optimization subject to connectivity constraints. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 657–668, 2013.

29 George B Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. In *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 154, pages 27:1–27:14, 2020.

30 George B. Mertzios, Hendrik Molter, Malte Renken, Paul G. Spirakis, and Philipp Zschoche. The complexity of transitively orienting temporal graphs. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 75:1–75:18, 2021.

31 George B. Mertzios, Hendrik Molter, and Viktor Zamaraev. Sliding window temporal graph coloring. *Journal of Computer and System Sciences*, 120:97–115, 2021.

32 Othon Michail and Paul G. Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016.

33 Othon Michail and Paul G. Spirakis. Elements of the theory of dynamic networks. *Communications of the ACM*, 61(2):72–72, January 2018.

34 Hendrik Molter, Malte Renken, and Philipp Zschoche. Temporal reachability minimization: Delaying vs. deleting. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS '21)*, pages 76:1–76:15, 2021.

35 Vincenzo Nicosia, John Tang, Cecilia Mascolo, Mirco Musolesi, Giovanni Russo, and Vito Latora. Graph metrics for temporal networks. In *Temporal Networks*. Springer, 2013.

36 Suhas Thejaswi, Juho Lauri, and Aristides Gionis. Restless reachability in temporal graphs. *CoRR*, abs/2010.08423, 2021. `arXiv:2010.08423`.

37 Tiphaine Viard, Matthieu Latapy, and Clémence Magnien. Computing maximal cliques in link streams. *Theoretical Computer Science*, 609:245–252, 2016.