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).

Due to lack of space, the full paper with all proofs is attached in a clearly marked Appendix to be read at the discretion of the Program Committee.

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.CVIT.2016.23

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.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:14
Leibniz International Proceedings in Informatics
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, 32, 34]. 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. [27] 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, 27] 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, 27, 31, 33, 35].

Furthermore, some non-path temporal graph problems have been recently introduced too, including for example temporal variations of maximal cliques [7, 36], vertex cover [4, 21], vertex coloring [30], matching [28], and transitive orientation [29]. 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, 33].

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. [27] 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.

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 an 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.** *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.** *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.** *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:*

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

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 $\text{ML}_{directed}(G)$ and $\text{MAL}_{directed}(G)$ 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*.

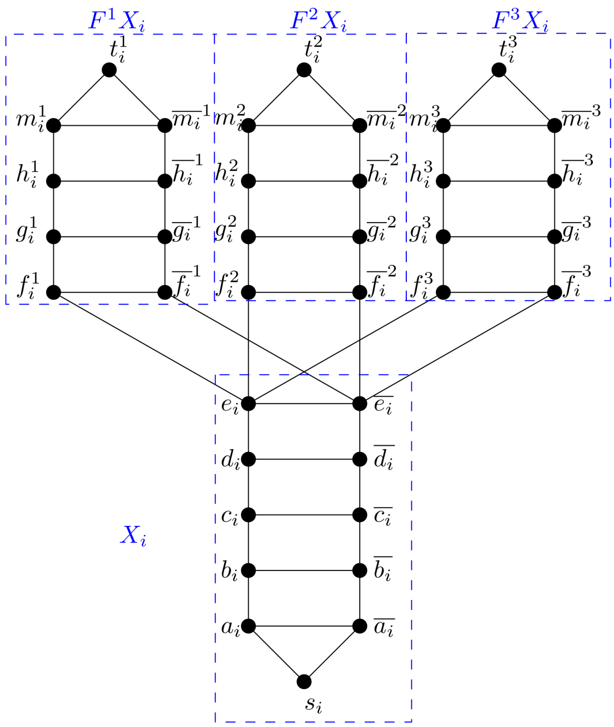


Figure 1 An example of a variable gadget X_i in G_ϕ, corresponding to the variable x_i from ϕ.

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 $\overline{d_j}$.

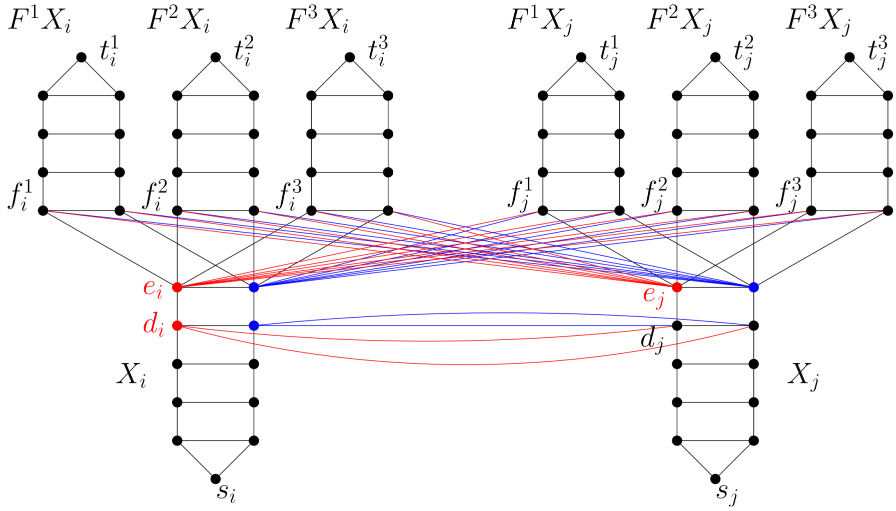


Figure 2 An example of two non-intersecting variable gadgets and variable edges among them.

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

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 $OPT_{\text{MAL}}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$, 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

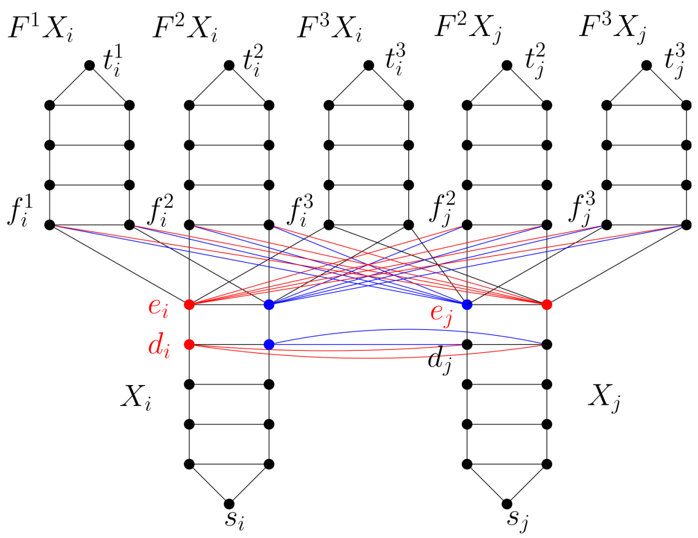


Figure 3 An example of two intersecting variable gadgets X_i, X_j corresponding to variables x_i, x_j, that appear together in some clause in ϕ, where it is the third appearance of x_i and the first appearance of x_j.

partial upward path on $F_i^{\ell'}$. Moreover, note that these two partial downward/upward paths 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.*

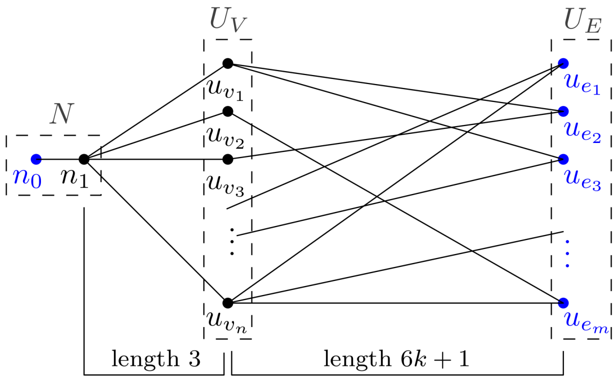


Figure 4 An example of construction of the input graph for MSL.

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].

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., $n_0 n_1$,
- 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.** *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.** 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.** 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.*

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. URL: `https://arxiv.org/abs/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 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.

28 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.

29 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.

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

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

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

33 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.

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

35 Suhas Thejaswi, Juho Lauri, and Aristides Gionis. Restless reachability in temporal graphs. *CoRR*, abs/2010.08423, 2021. URL: `https://arxiv.org/abs/2010.08423`.

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

APPENDIX

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.CVIT.2016.23

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.

APPENDIX

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 [24, 33, 35]. Inspired by the foundational work of Kempe et al. [26], 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 [26]). *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. [26] 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, 26, 27] 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, 21, 27, 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, 22], 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].

APPENDIX

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

APPENDIX

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 19 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 21) 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.

APPENDIX

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)	Min. Aged Labeling (MAL)
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 $\lvert\lambda\rvert \leq k$?	**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 $\lvert\lambda\rvert \leq k$ and $\alpha(\lambda) \leq a$?

Min. Steiner Labeling (MSL)	Min. Aged Steiner Labeling (MASL)
Input: A static graph $G = (V, E)$, a subset $R \subseteq V$ and a $k \in \mathbb{N}$. **Question:** Does there exist an temporally R-connected temporal graph (G, λ), where $\lvert\lambda\rvert \leq k$?	**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 $\lvert\lambda\rvert \leq k$ 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.** *For every graph G with n vertices and diameter d, we have that $\kappa(G, d) \leq n(n-1)$.*

Proof. For every vertex v of $G = (V, E)$, consider a BFS tree T_v rooted at v, while every edge from a vertex $u \neq v$ to its parent in T_v is assigned the time-label $dist(v, u)$, i.e., the length of the shortest path from v to u in G. Note that each of these time-labels is smaller than or equal to the diameter d of G. Clearly, each BFS tree T_v assigns in total $n - 1$ time-labels to the edges of G, and thus the union of all BFS trees T_v, where $v \in V$, assign in total at most $n(n-1)$ labels to the edges of G. ◀

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 $\kappa(C_n, d) = \Theta(n^2)$.

APPENDIX

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

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

Proof. Let $V(C_n) = \{v_1, v_2, \ldots, v_n\}$ be the vertices of C_n. In the following, if not specified otherwise, all subscripts are considered modulo n. We distinguish two cases, depending on the parity of n.

First, when n is odd, i.e., $n = 2d + 1$. In this case one can observe that for each vertex $v_i \in V(C_n)$ there are exactly two vertices on the distance d from it, namely v_{i+d} (on the *right side* of v_i) and v_{i-d} (on the *left side* of v_i). Therefore, the $(v_i, v_{i+d}/v_{i-d})$ and $(v_{i+d}/v_{i-d}, v_i)$-temporal paths must be labeled using all labels from 1 to d, one per each edge. Note also that each edge v_iv_{i+1} lies on the d temporal paths when the starting vertex v_j is on the left side of it ($j \in \{i, i-1, i-2, \ldots, i-d\}$) and on d temporal paths, when the starting vertex $v_{j'}$ is on the right side of it ($j' \in \{i, i+1, i+2, \ldots, i+d\}$). This results in edge v_iv_{i+1} admitting all labels. As this is true for any edge of C_n, each edge is labeled with all labels. Therefore we need $n \cdot d = 2d^2 + 1$ labels to ensure the existence of temporal paths among any two vertices in C_{2d+1}.

Now let us continue with the case when n is even, i.e., $n = 2d$. In this case each vertex $v_i \in V(C_n)$ has exactly one vertex, $v_{i-d} = v_{i+d}$, on the distance d from it and two on the distance $d-1$ from it (v_{i-d+1} and v_{i+d-1}). Therefore we have to label two disjoint paths starting in v_i, one of length d and the other of length $d-1$. Suppose that we chose the following labeling to label the edges of C_n. Let $i \in \{1, 2, \ldots, d\}$, if the edge is of form $v_{2i}v_{2i+1}$ then it is labeled with all even labels, $\{2, 4, 6, \ldots, j\}$, where $j \leq d$, and if the edge is of form v_{2i+1}, v_{2i} then it is labeled with all odd labels, $\{1, 3, 5, \ldots, j'\}$, where $j' \leq d$. Now vertices v_{2i-1} and v_{2i} use the same labels (i.e., the same temporal paths), to reach all other vertices from the cycle. Namely, the (v_{2i-1}, v_{2i+d-1})-temporal path is of length d, uses all labels from 1 to d and visits vertices $v_{2i-1}, v_{2i}, v_{2i+1}, \ldots, v_{2i+d-1}$. Therefore v_{2i-1} and v_{2i} can reach vertices $\{v_{2i+1}, v_{2i+2}, \ldots, v_{2i+d-1}\}$. Similarly, the (v_{2i}, v_{2i-d})-temporal path is of length d, uses all labels from 1 to d and visits vertices $v_{2i}, v_{2i-1}, v_{2i-2}, \ldots, v_{2i-d}$. So v_{2i-1} and v_{2i} can reach vertices $\{v_{2i-2}, v_{2i-3}, \ldots, v_{2i-d}\}$. This implies that that v_{2i} and v_{2i-1} reach all other vertices in the graph. This holds for any two endpoints of an edge in C_n. Therefore we need $d \cdot \frac{n}{2} = d^2$ labels to ensure the existence of temporal paths among any two vertices in C_{2d}. ◀

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 [23]).

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. [20], where the relationship with the gossip problem has also been drawn. Contrary to ML, this problem is NP-hard [20].

APPENDIX

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 (see Lemma 6).

From the set of agents A and a sequence of calls $\mathcal{C} = c(1), c(2), \ldots, c(m)$ we build a temporal graph $\mathcal{G}_\mathcal{C} = (G, \lambda)$ by the following procedure. For every agent $x \in A$ we create a vertex $v_x \in V(G)$. Every phone call $c(i)$ between two agents x, y gives rise to a time edge $(v_x v_y, i)$ of $\mathcal{G}_\mathcal{C}$. Therefore the labeling λ is defined by the sequence of phone calls. Since no two calls are concurrent, we can order them linearly: for every $1 \leq i \leq m$, the phone call $c(i)$ gives the label i to the edge between the two agents involved.

▶ **Observation 5.** *If the sequence $c(1), c(2), \ldots, c(m)$ of m phone calls results in all agents knowing all secrets, then the above construction produces a temporally connected temporal graph $\mathcal{G}_\mathcal{C} = (G, \lambda)$ with $|\lambda| = m$.*

Now note that the temporal graph $\mathcal{G}_\mathcal{C}$ produced by the above procedure has the special property that, for every time-label $t = 1, 2, \ldots, m$, there exists exactly one edge labeled with t. In the next lemma we prove the reverse statement of Observation 5.

▶ **Lemma 6.** *Let (G, λ) be an arbitrary temporally connected temporal graph with $|\lambda| = m$ time-labels in total. Then there exists a sequence $c(1), c(2), \ldots, c(m)$ of m phone calls that results in all agents knowing all secrets.*

Proof. Let (G, λ) be an arbitrary temporally connected temporal graph. W.l.o.g. we may assume that, for every $t = 1, 2, \ldots, \alpha(G, \lambda)$, there exists at least one edge e such that $t \in \lambda(e)$. Indeed, if such an edge does not exist in (G, λ), we can replace in (G, λ) every label $t' > t$ by $t' - 1$, thus obtaining another temporally connected graph with a smaller age.

Now we proceed as follows. Let $t \in \{1, 2, \ldots, \alpha(G, \lambda)\}$ be an arbitrary time step within the lifetime of (G, λ), and let $\{e_{i_k}\}_{k=1}^{t}$ be the edges of G such that $t \in \lambda(e_{i_k})$. Let $\varepsilon = \frac{1}{2t}$. For every $k = 1, \ldots, t$, we replace the label t of the edge e_{i_k} by the label $t + k\varepsilon$. Finally, we normalize the new time-labels of the edges of G such that they become the distinct consecutive natural numbers from 1 to m (since $|\lambda| = m$ by the assumption of the lemma). Denote the resulting temporal graph by (G, λ'). Note that every temporal path in (G, λ) corresponds to a temporal path in (G, λ') with the same sequence of edges, and vice versa.

Finally we create the required sequence of phone calls as follows: for every $i = 1, 2, \ldots, m$, if (G, λ') contains the edge e with time-label i, we add a phone call $c(i)$ between the two

APPENDIX

endpoints of the edge e. Since both (G, λ) aqnd (G, λ') are temporally connected, it follows that after the sequence $c(1), c(2), \ldots, c(m)$ of calls results in every agent knowing every secret. This completes the proof. ◀

Now denote with $f(n)$ the minimum number of calls needed to complete gossiping among a set A of n agents, where a specific set of pairs of vertices $B \subseteq A \times A$ are allowed to make a direct call between each other. Let $G_0 = (A, B)$ be the (static) graph having the agents in A as vertices and the pairs of B among them as edges. Then it is known by [9] that, if G_0 contains a C_4 as a subgraph then $f(n) = 2n - 4$, while otherwise $f(n) = 2n - 3$. Therefore the next theorem follows by Observation 5 and Lemma 6 and by the results of [9].

▶ **Theorem 7.** *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:*

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

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. First we define a *canonical layering* of a DAG, which is useful for our algorithm.

▶ **Definition 8.** *Let $G = (V, A)$ be a DAG with n vertices, m arcs, and diameter d. A partition $L_0, L_1, L_2, \ldots, L_d$ of V into $d+1$ sets is a* canonical layering *of G if, for every $0 \le i \le d$, the set L_i contains all the source vertices in the induced subgraph $G_i := G[\{L_i, L_{i+1}, \ldots, L_d\}]$.*

An example of a canonical layering of a DAG G is illustrated in Figure 1.

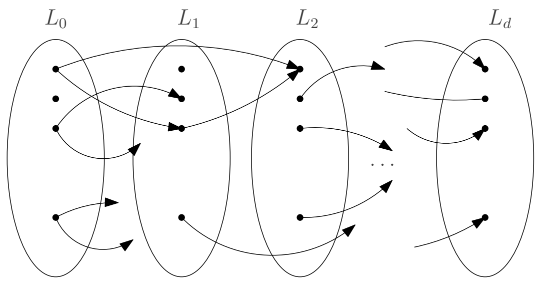


Figure 1 Example of a canonical layering.

▶ **Lemma 9.** *Let $G = (V, E)$ be a DAG with n vertices and m arcs. We can produce the canonical layering of G in linear $O(n + m)$ time.*

APPENDIX

Proof. First we initialize an auxiliary vertex subset $S = \emptyset$ and a counter $s_v = 0$ for every vertex v. We start by computing the vertices of L_0 in $O(n+m)$ time by just visiting all vertices and arcs of G; L_0 contains all vertices u such that $N^-(u) = \emptyset$. Now, for every $i \geq 0$ we proceed as follows. First we set $S = \emptyset$. Then, for every arc (u, v), where $u \in L_i$, we add v to S and we increase the counter s_v by 1. Then we set $L_{i+1} = \{v \in S : s_v = |N^-(v)|\}$. Before we continue to the next iteration $i+1$, we reset the set S to be $\emptyset$, and we iterate until we reach all vertices of G, i.e., until we add every vertex u to one of the sets $L_0, L_1, \ldots, L_d$.

It is easy to check that the above procedure is correct, as at every iteration $i+1$ (where $i \geq 0$) we include to L_i all vertices v which have zero in-degree in the graph induced by the vertices in $V \setminus \bigcup_{k=1}^{i} L_k$. Furthermore, the running time is clearly $O(n+m)$ as we visit each vertex and arc a constant number of times. ◀

The following observations will be useful when considering the canonical layering.

▶ **Observation 10.** *Each layer L_i is an independent set in G.*

▶ **Observation 11.** *For every $i = 0, 1, \ldots, d-1$ and every $u \in L_i$, there exists an arc $(u, v) \in A$ such that $v \in L_{i+1}$.*

▶ **Observation 12.** *For every arc $(u, v) \in A$, where $u \in L_i$ and $v \in L_{i+1}$ for some $i \in \{0, 1, \ldots, d-1\}$, there is no directed path of length two or more from u to v in G.*

We use the canonical layering to prove the following result.

▶ **Theorem 13.** *Let $G = (V, E)$ be a DAG with n vertices and m arcs. Then $\mathrm{ML}_{directed}(G)$ and $\mathrm{MAL}_{directed}(G)$ can be both computed in $O(n(n+m))$ time.*

Proof. For the purposes of simplicity of the proof, we denote by $\kappa(G)$ the optimum value of $\mathrm{ML}_{directed}$ with the DAG G as its input. First we calculate the canonical layering $L_0, L_1, \ldots, L_d$ of G in $O(n+m)$ time by Lemma 9. For simplicity of the presentation, denote by G_v the induced subgraph of G that contains v and all vertices that are reachable by v in G with a directed path. Let d_v be the diameter of G_v; note that d_v is the length of the longest shortest directed path in G that starts at v. For every vertex $u \in V$, we define the set $L_0^u = \{u\}$ and we initialize the set $S_u = N^+(u)$. Then, similarly to the proof of Lemma 9, we iterate over all vertices $v \in S_u = N^+(u)$ and over all vertices $w \in N^+(v)$. Whenever we encounter a vertex $w \in N^+(v) \cap N^+(u)$, we remove v from S_u. At the end of this procedure, the set S_u contains exactly those vertices of $v \in N^+(u)$, for which there is no directed path of length two or more from u to v in G. The above procedure can be completed in $O(n(n+m))$ time, as for every vertex u, we iterate at most over all arcs in G a constant number of times.

Now we define the labeling λ of G as follows: Every arc $(u, v) \in A$, where $u \in L_i$, $v \in L_j$, and $v \in S_u$, gets the label $\lambda((u, v)) = j$. Note here that $1 \leq \lambda((u, v)) \leq d$ for every arc of G, and thus the age $\alpha(G, \lambda)$ of the resulting temporal graph is equal to the diameter d of G. We will prove that $|\lambda| = \kappa(G)$. To prove that $|\lambda| \leq \kappa(G)$, it suffices to show that every label of λ must participate in every temporal labeling of G which preserves temporal reachability. In fact, this is true as the only arcs of G, which have a label in λ, are those arcs (u, v) such that there is no other directed path from u to v. That is, in order to preserve temporal reachability, we need to assign at least one label to all these arcs.

Conversely, to prove that $|\lambda| \geq \kappa(G)$, it suffices to show that λ preserves all temporal reachabilities. For this, observe first that, every directed path $P = (a, \ldots, b)$ in G can be transformed to a directed path $P' = (a, \ldots, b)$ such that, for every arc (u, v) in P', there is no other directed path from u to v in G apart from the arc (u, v) (i.e., there is no "shortcut"

APPENDIX

from u to v in G). Therefore, since every arc in P' is assigned a label in λ and these labels are increasing along P', it follows that λ preserves all temporal reachabilities, and thus $|\lambda| \geq \kappa(G)$. Summarizing, $|\lambda| = \kappa(G)$ and the labeling λ can be computed in $O(n(n+m))$ time.

Finally, since $\alpha(G, \lambda) = d$, the obtained optimum labeling for ML is also an optimum labeling for MAL (provided that the upper bound a in the input of MAL is at least d). ◀

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 14** ([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

APPENDIX

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 2.

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*.

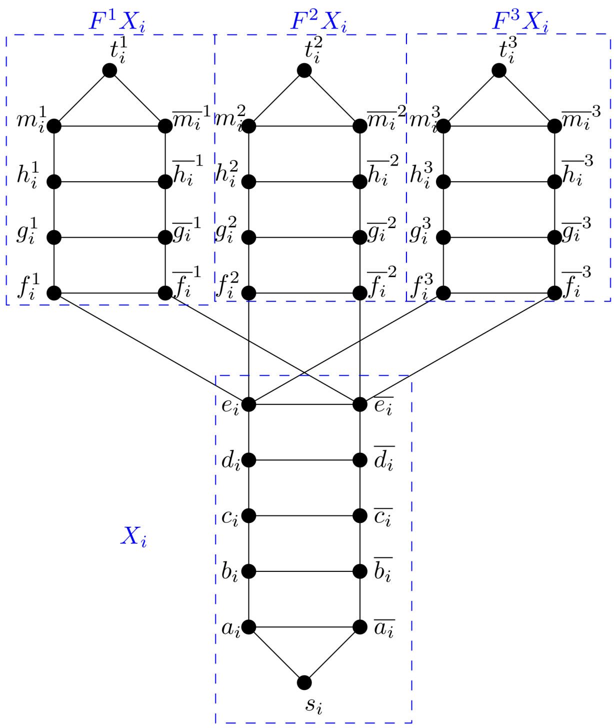


Figure 2 An example of a variable gadget X_i in G_ϕ, corresponding to the variable x_i from ϕ.

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 $\ell' \in \{1,2,3\}$,

APPENDIX

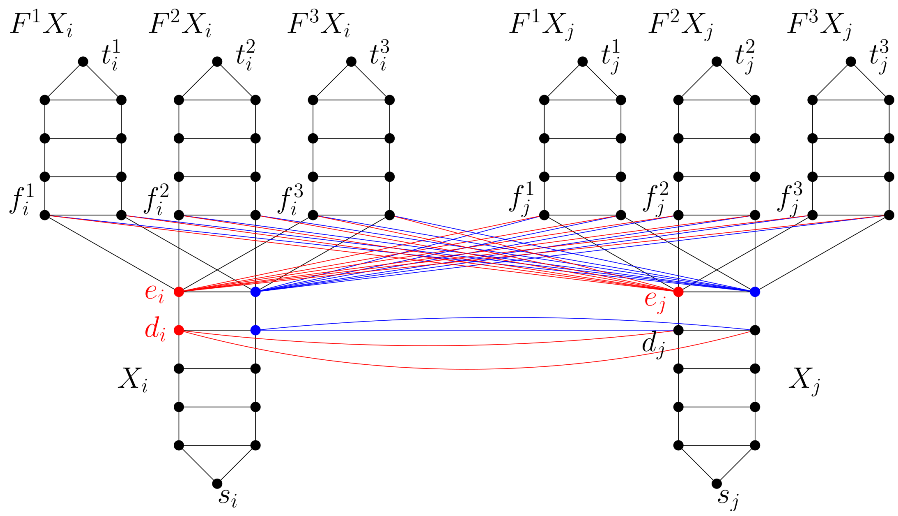


Figure 3 An example of two non-intersecting variable gadgets and variable edges among them.

- 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 3.

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 4.

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

▶ **Lemma 15.** *The diameter d_ϕ of G_ϕ is* 10.

Proof. We prove this in two steps. First we show that the diameter of any variable gadget is 10 and then show that the diameter does not increase, when the variable edges are introduced, i.e., vertices in any two variable gadgets are at most 10 apart.

Let us start with fixing a variable gadget X_i. A path from the starting vertex s_i to any ending vertex t_i^ℓ ($\ell \in \{1,2,3\}$) has to go through at least one of the vertices from $\{a_i, \overline{a_i}\}$, then through at least one of the vertices from $\{b_i, \overline{b_i}\}$, then through $\{c_i, \overline{c_i}\}$, $\{d_i, \overline{d_i}\}$, $\{e_i, \overline{e_i}\}$, $\{f_i^\ell, \overline{f_i}^\ell\}$, $\{g_i^\ell, \overline{g_i}^\ell\}$, $\{h_i^\ell, \overline{h_i}^\ell\}$ and finally through $\{m_i^\ell, \overline{m_i}^\ell\}$, before reaching the ending vertex. The shortest s_i, t_i^ℓ path will go through exactly one vertex from each of the above sets. Therefore it is of length 10. Because of the construction of X_i, there are exactly two s_i, t_i^ℓ paths of length 10, which are edge and vertex disjoint, as they share only the starting and ending vertices. One of this paths uses the vertices $a_i, b_i, c_i, d_i, e_i, f_i^\ell, g_i^\ell, h_i^\ell, m_i^\ell$ (i.e., the left

APPENDIX

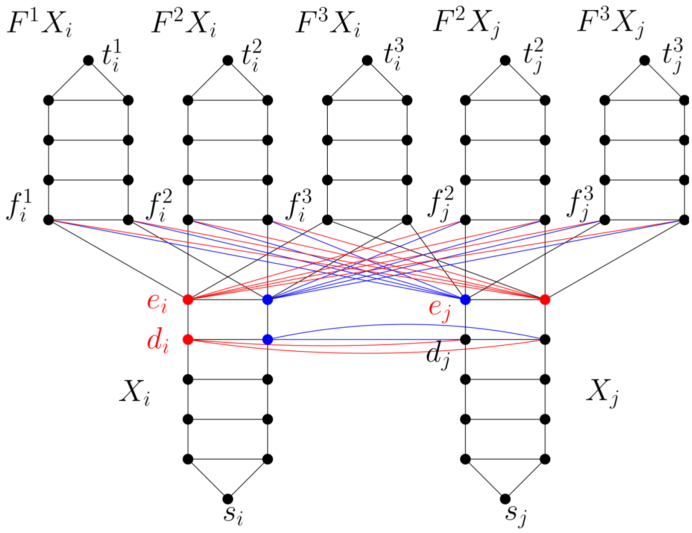


Figure 4 An example of two intersecting variable gadgets X_i, X_j corresponding to variables x_i, x_j, that appear together in some clause in ϕ, where it is the third appearance of x_i and the first appearance of x_j.

path) and the other uses vertices $\overline{a_i}, \overline{b_i}, \dots, \overline{m_i}^\ell$ (i.e., the right path). A path between any two ending vertices $t_i^\ell, t_i^{\ell'}$ (where $\ell, \ell' \in \{1,2,3\}$ and $\ell \neq \ell'$), has to go through the following sets of vertices, $\{m_i^\ell, \overline{m_i}^\ell\}$,$\{m_i^{\ell'}, \overline{m_i}^{\ell'}\}$, $\{h_i^\ell, \overline{h_i}^\ell\}$,$\{h_i^{\ell'}, \overline{h_i}^{\ell'}\}$, $\{g_i^\ell, \overline{g_i}^\ell\}$,$\{g_i^{\ell'}, \overline{g_i}^{\ell'}\}$, $\{f_i^\ell, \overline{f_i}^\ell\}$,$\{f_i^{\ell'}, \overline{f_i}^{\ell'}\}$, $\{e_i, \overline{e_i}\}$. Similarly as before, the shortest path uses exactly one vertex from each set and is of size 10. Even more, there are exactly two $t_i^\ell, t_i^{\ell'}$ paths of length 10. They are edge and vertex disjoint, as they share only the starting and ending vertices. One of this paths uses the vertices without the line in the label (i.e., the left path) and the other uses vertices with the line in the label (i.e., the right path). It is not hard to see that the distance between any other vertex in X_i and starting or ending vertices is at most 9, as that vertex lies on one of the s_i, t_i^ℓ or $t_i^\ell, t_i^{\ell'}$-paths, but it is not an endpoint of it. By the similar reasoning there exists a path between any two vertices in X_i (different than s_i, t_i^ℓ), of distance at most 9. Therefore the diameter of X_i is 10.

Now let us fix two variable gadgets X_i, X_j, that share no fork (i.e., x_i and x_j appear in no clause of ϕ). The shortest path from the starting vertex s_i of X_i to the starting vertex s_j of X_j has to reach vertex d_i (resp. $\overline{d_i}$), which is done in 4 steps, from where it connects to either d_j or $\overline{d_j}$, using a variable edge, and continues toward s_j, with 4 edges. Therefore, $d(s_i, s_j) = 9$. The shortest path connecting vertex s_i with $t_j^{\ell'}$, uses one of the vertices e_i or $\overline{e_i}$, that are on the distance 5 from s_i, then using one variable edge reaches $f_j^{\ell'}$ or $\overline{f_j}^{\ell'}$, which is on the distance 4 from the ending vertex $t_j^{\ell'}$. Therefore, $d(s_i, t_j^{\ell'}) = 10$, for all $\ell' \in \{1,2,3\}$. Lastly, the shortest path between an ending vertex t_i^ℓ of X_i and an ending vertex $t_j^{\ell'}$ uses 4 edges in the fork $F^\ell X_i$ to reach the vertex f_i^ℓ or $\overline{f_i^\ell}$, from where it uses a variable edge that connects it to the vertex e_j or $\overline{e_j}$, that is on the distance 5 from the $t_j^{\ell'}$. Therefore, $d(t_i^\ell, t_j^{\ell'}) = 10$, for all $\ell, \ell' \in \{1,2,3\}$. It is not hard to see that if two variable gadgets X_i, X_j share a fork the shortest path among any two vertices does not increase.

We proved that the distance among any two vertices in G_ϕ is at most 10 and thus its diameter is 10. ◀

APPENDIX

▶ **Theorem 16.** *If $OPT_{\text{MonMaxXOR}(3)}(\phi) \geq k$ then $OPT_{\text{MAL}}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$, where n is the number of variables in the formula ϕ.*

Proof. Let τ be an optimum truth assignment of ϕ, i.e., a truth assignment that satisfies at least k clauses of ϕ. We will prove that there exists a temporal labeling λ_ϕ of G_ϕ which uses $|\lambda_\phi| \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$ labels, such that (G, λ) is temporally connected and $\alpha(G, \lambda) = d_\phi = 10$. Recall that, since ϕ is an instance of MonMaxXOR(3) with n variables, it has $m = \frac{3}{2}n$ clauses. We build the labeling λ_ϕ using the following rules. For an illustration see Figure 5.

1. If a variable x_i from ϕ is set to be True by the truth assignment τ, we label the edges in X_i in the following way:
   - all three left (s_i, t_i^ℓ)-paths, for all $\ell \in \{1, 2, 3\}$, get the labels $1, 2, 3, \ldots, 10$, one on each edge,
   - similarly, all left (t_i^ℓ, s_i)-paths, get the labels $1, 2, 3, \ldots, 10$, one on each edge,
   - all connecting edges (i.e., edges of form $y_i\overline{y_i}$, where $y \in \{a, b, c, d, e, f^\ell, g^\ell, h^\ell, m^\ell\}$) get the labels 1 and 10.

   If a variable x_i from ϕ is set to be False by the truth assignment τ, we label the edges in X_i in the following way:
   - all three right (s_i, t_i^ℓ)-paths, for all $\ell \in \{1, 2, 3\}$, get the labels $1, 2, 3, \ldots, 10$, one on each edge,
   - similarly, all right (t_i^ℓ, s_i)-paths, get the labels $1, 2, 3, \ldots, 10$, one on each edge,
   - all connecting edges get the labels 1 and 10.

   Labeling λ_ϕ uses 10 labels on the left (resp. right) path of the base BX_i, 10 labels on the left (resp. right) path of each fork $F^\ell X_i$, where $\ell \in \{1, 2, 3\}$ and $10 + 3 \cdot 8$ labels on the connecting edges. All in total λ_ϕ uses 74 labels on the variable gadget X_i.

   We still need to prove that there exists a temporal path among any two vertices in X_i. There is a (unique) temporal path from the starting s_i vertex to all three ending vertices t_i^ℓ, where $\ell \in \{1, 2, 3\}$, using left (in case of x_i being True) or right (in case of x_i being False) paths of the base BX_i and forks $F^\ell X_i$. Similarly it holds for all temporal (t_i^ℓ, s_i)-paths. The temporal path connecting two ending vertices $t_i^{\ell_1}, t_i^{\ell_2}$, uses first the left (in case of x_i being True) or right (in case of x_i being False) path of the fork $F^{\ell_1} X_i$ to reach e_i (in case of x_i being True) or $\overline{e_i}$ (in case of x_i being False), using the labels 1 to 5, and then continues on the left (in case of x_i being True) or right (in case of x_i being False) path of the $F^{\ell_2} X_i$ from e_i or $\overline{e_i}$ to $t_i^{\ell_2}$, using labels 6 to 10. Any vertex not on the left (in case of x_i being True) or right (in case of x_i being False) path, can reach the starting vertex or any of the ending vertices, using a connecting edge at time 1. Similarly it hold for the paths in the opposite direction, where the connecting edges have the label 10. A temporal path among two vertices not on the left (in case of x_i being True) or right (in case of x_i being False) path uses first a connecting edge at time 1, then a portion of the left (in case of x_i being True) or right (in case of x_i being False) path and again the appropriate connecting edge at time 10. This proves that λ_ϕ on X_i admits a temporal path among any two vertices in X_i.

2. If two variable gadgets X_i and X_j do not share a fork, i.e., variables x_i and x_j are not in the same clause in ϕ, and are both set to True by τ, then we label the following variable gadgets:
   - the edge $d_i d_j$, connecting the left path of BX_i with the left path of BX_j, gets the label 5,
   - three edges of the form $e_i f_j^{\ell'}$ ($\ell' \in \{1, 2, 3\}$), that connect the left path of BX_i to left paths of $F^{\ell'} X_j$, with the labels 4 and 6,

APPENDIX

- three edges of the form $e_j f_i^\ell$ ($\ell \in \{1,2,3\}$), that connect the left path of BX_j to left paths of $F^\ell X_i$, with the labels 4 and 6.

The labeling λ_ϕ uses 74 labels for each variable gadget and 13 labels on 7 variable edges that connect both variable gadgets. Note, the three other combinations (x_i, x_j are both FALSE, one of x_i, x_j is TRUE and the other FALSE) give rise to the labeling λ_ϕ that uses the same number of labels on both variable gadgets and variable edges, where the labeled variable edges are chosen appropriately.

Since labeling variable edges does not change the labeling on each variable gadget, we know that there is still a temporal path among any two vertices from the same variable gadget. We need to prove now that there is a temporal path among any two vertices from X_i and X_j. The edge $d_i d_j$, with the label 5, connects all the vertices from the $BX_i \setminus \{e_i, \overline{e_i}\}$ to the vertices from the $BX_j \setminus \{e_j, \overline{e_j}\}$ and vice versa. To go from the starting vertex s_i of X_i to the ending vertex t_j^ℓ of X_j we use the following route. From s_i to e_i we use the left labeled path on X_i with labels from 1 to 5, then the edge $e_i f_j^{\ell'}$ at time 6 to reach the corresponding fork $F^{ell'} X_j$ of X_j and from $f_j^{\ell'}$ to the ending vertex $t_j^{\ell'}$ we use the left labeled path of X_j with labels 7 to 10. This temporal path connects all vertices in the base BX_i to all vertices in the forks $FX_j^{\ell'}$, where $\ell' \in \{1,2,3\}$. Similar we obtain temporal paths from vertices in the base BX_j to vertices in the forks FX_i^ℓ, where $\ell \in \{1,2,3\}$. To go from any vertex in the fork $F^\ell X_i$ to any vertex of the X_j we use the following route. First, we reach the vertex f_i^ℓ, by the time 4, using the left labeled path of X_i. Then we use the edge $f_i^\ell e_j$ at time 5. Now, by the construction of λ_ϕ of X_j, each vertex in X_j can be reached from e_j from time 5 to 10. Therefore all vertices from $F^\ell X_i$ can reach any vertex in X_j. This is true for all $\ell \in \{1,2,3\}$. Similarly it holds for temporal paths from any vertex in the fork $F^{\ell'} X_j$ ($\ell' \in \{1,2,3\}$) to vertices of the X_i. The only thing left to show is that the vertices $\{e_i, \overline{e_i}$ can reach all other vertices in BX_j. This is true as there is a temporal path using the edge $e_i f_j^{\ell'}$ at time 5 and then, from $f_j^{\ell'}$ to any vertex in the base BX_j, the left labeled path of BX_j, that is labeled by λ_ϕ. This is true for all $\ell' \in \{1,2,3\}$. Similarly it holds for the temporal paths from $\{e_j, \overline{e_j}\}$ to the vertices in BX_i. Therefore λ_ϕ admits a temporal path among any two vertices of variable gadgets, that do not share the fork.

3. If two variable gadgets X_i and X_j share a fork, i.e., variables x_i and x_j are in the same clause, are both set to TRUE and $F^r X_i = F^{r'} X_j$, then we label the following variable edges:
- the edge $d_i d_j$, connecting the left path of BX_i and BX_j, gets the label 5,
- two edges of the form $e_i f_j^{\ell'}$ ($\ell' \in \{1,2,3\} \setminus \{r'\}$), that connect the left path of BX_i to left paths of $F^{\ell'} X_j$, with the labels 4 and 6,
- two edges of the form $e_j f_i^\ell$ ($\ell \in \{1,2,3\} \setminus \{r\}$), that connect the left path of BX_j to left paths of $F^\ell X_i$, with the labels 4 and 6.

The labeling λ_ϕ uses 9 labels on 5 variable edges that connect both variable gadgets. Note, the three other combinations (x_i, x_j are both FALSE, one of x_i, x_j is TRUE and the other FALSE) give rise to the labeling λ_ϕ that uses the same number of labels on variable edges, where the labeled edges are chosen accordingly to the truth values of x_i and x_j. The only difference is in the labeling of the shared fork $F^r X_i = F^{r'} X_j$. There are two possibilities, one when the truth value of x_i and x_j is the same and one when it is different, i.e., $x_i = x_j$ or $x_i \neq x_j$.

a) Let us start with the case when $x_i \neq x_j$. Without loss of generality (w.l.o.g.) we can assume that x_i is TRUE and x_j FALSE. In the labeling λ_ϕ we label all left paths in the variable gadget X_i and all right paths in X_j. By the construction of the graph G_ϕ

(and the rules of how to identify vertices of the two forks), the left labeling of $F^r X_i$ coincides with the right labeling of $F^{r'} X_j$. Therefore λ_ϕ uses $2 \cdot 74 - 16 = 132$ labels on both variable gadgets.

b) Let us now observe the case when $x_i = x_j$. W.l.o.g. we can assume that both variables are True. In the labeling λ_ϕ we label all left paths of both variable gadgets. By the construction of the graph G_ϕ (and the rules of how to identify vertices of the two forks), the fork $F^r X_i = F^{r'} X_j$ gets labeled from both sides, i.e., all edges in the fork get 2 labels. Therefore λ_ϕ uses $2 \cdot 74 - 8 = 140$ labels on both variable gadgets.

Identifying two forks $F^r X_i = F^{r'} X_j$ and labeling them using the union of both labelings on each fork, preserves temporal paths among all the vertices from X_i and X_j. This is true as the labeling in each variable is not changed by the labeling in the other variable. Among forks that are not in the intersection there are still the variable edges left, which assure that vertices from different variable gadgets can reach them or can be reached by them. Therefore the labeling λ_ϕ admits a temporal path among any two vertices from the variable gadgets X_i, X_j, that have a fork in the intersection.

Summarizing all of the above we get that the labeling λ_ϕ uses 74 labels on each variable gadget and 13 labels on variable edges among any two variables, from which we have to subtract the following:

- 4 labels for each pairs of variable edges between two variables that appear in the same clause,
- 16 labels for the shared fork between two variables, that appear in a satisfied clause,
- 8 labels for the shared fork between two variables, that appear in a non-satisfied clause.

Altogether sums up to the $74n + 13\frac{n(n-1)}{2} - 4m - 16k - 8(m-k)$ labels. Therefore, if τ satisfies at least k clauses of ϕ, the labeling λ_ϕ consists of at most $\frac{13}{2}n^2 + \frac{99}{2}n - 8k$ labels. ◀

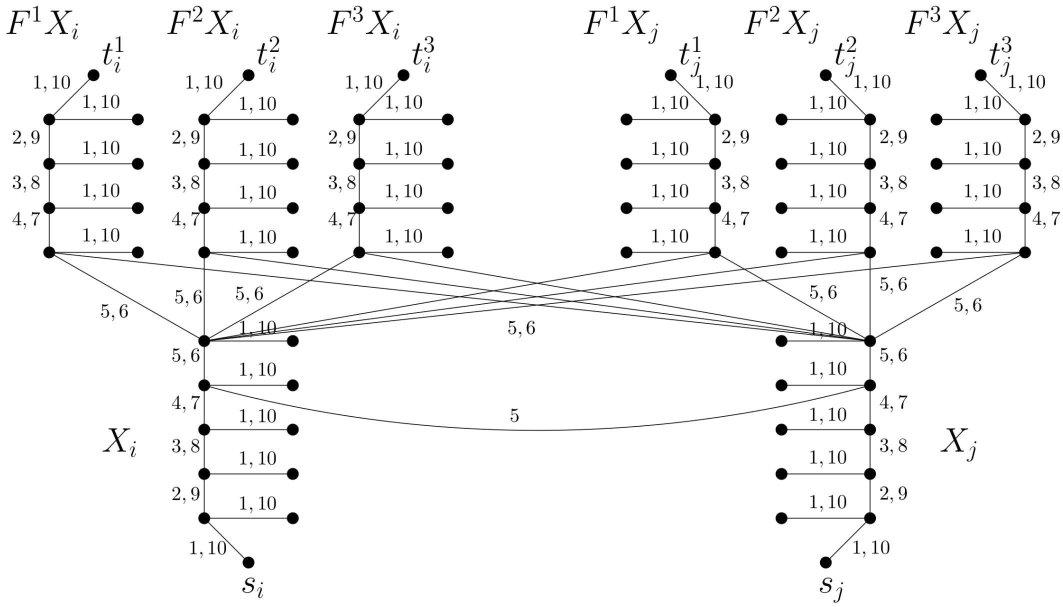
Before proving the statement in the other direction, we have to show some structural properties. Let us fix the following notation. If a labeling λ_ϕ labels all left (resp. right) paths of the variable gadget X_i (i.e., both bottom-up from s_i to t_i^1, t_i^2, t_i^3 and top-down from t_i^1, t_i^2, t_i^3 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 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 17.** *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.*

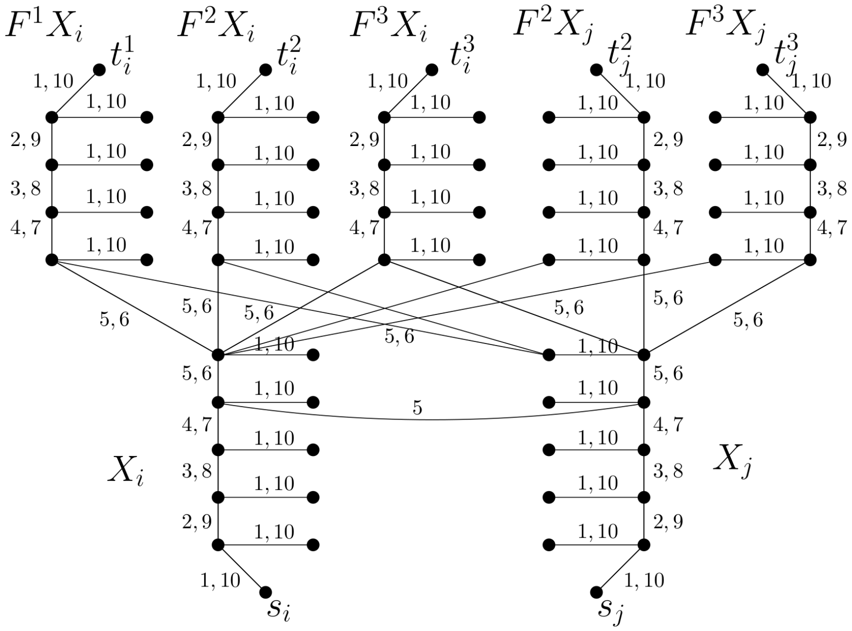
Proof. Let λ_ϕ be a minimum labeling of G_ϕ that admits at least one variable gadget X_i that is neither left-aligned nor right-aligned.

First we will prove that there exists a fork $F^\ell X_i$ of X_i that admits at least three partial upward or downward paths, i.e., it either has two partial upward paths (one on each side of

APPENDIX



(a) x_i and x_j do not appear together in any clause.



(b) x_i and x_j appear together in a clause, where x_i appears with its third and x_j with its first appearance.

Figure 5 Example of the labeling λ on variable gadgets X_i, X_j and variable edges between them, where x_i is TRUE and x_j FALSE in ϕ. Note, edges that are not labeled are omitted, $F^3X_i = F^1X_j$ and $t_i^3 = t_j^1$.

the fork) and at least one partial downward path, or two partial downward paths (one on each side of the fork) and at least one partial upward path. For the sake of contradiction, suppose that each of the forks F^1X_i, F^2X_i, F^3X_i contains at most two partial upward or downward paths. Then, since λ_ϕ must have in X_i at least one upward and at least one downward path between s_i and t_i^ℓ, $\ell \in \{1, 2, 3\}$, it follows that each fork $F^\ell X_i$ has *exactly*

APPENDIX

one partial upward and *exactly* one partial downward path.

Assume that each of the forks F^1X_i, F^2X_i, F^3X_i has both its partial upward and downward paths on the same side of X_i (i.e., either both on the left or both on the right side of X_i). If all of them have their partial upward and downward paths on the left (resp. right) side of X_i, then X_i is left-aligned (resp. right-aligned), which is a contradiction. Therefore, at least one fork (say F^1X_i) has its partial upward and downward paths on the left side of X_i and at least one other fork (say F^2X_i) has its partial upward and downward paths on the right side of X_i. But then there is no temporal path from t_i^1 to t_i^2 of length 10 in λ_ϕ, which is a contradiction. Therefore there exists at least one fork $F^\ell X_i$ (say, F^1X_i w.l.o.g.), in which (w.l.o.g.) the partial upward path is on the right side and the partial downward path is on the left side of X_i.

Since the partial downward path of F^1X_i is on the left side of X_i, it follows that the partial upward path of each of F^2X_i and F^2X_i is on the left side of X_i. Indeed, otherwise there is no temporal path of length 10 from t_i^1 to t_i^2 or t_i^3 in λ_ϕ, a contradiction. Similarly, since the partial upward path of F^1X_i is on the right side of X_i, it follows that the partial downward path of each of F^2X_i and F^2X_i is on the right side of X_i. But then, there is no temporal path of length 10 from t_i^2 to t_i^3, or from t_i^3 to t_i^2 in λ_ϕ, which is also a contradiction. Therefore at least one fork $F^\ell X_i$ (say F^3X_i) of X_i admits at least three partial upward or downward paths.

W.l.o.g. we can assume that the fork F^3X_i has two partial downward paths and at least one partial upward path which is on the left side of X_i. We distinguish now the following cases.

Case A. The fork F^3X_i has no partial upward path on the right side of X_i. Then the base BX_i has a partial upward path on the left side of X_i. Furthermore, each of the forks F^1X_i, F^2X_i has a partial downward path on the left side of X_i.

Case A-1. The base BX_i of X_i has no partial downward path on the left side of X_i; that is, there is no temporal path from vertex e_i to vertex s_i with labels "6,7,8,9,10". Then the base BX_i of X_i has a partial downward path on the right side of X_i, as otherwise there would be no temporal path of length 10 from any of t_i^1, t_i^2, t_i^3 to s_i. For the same reason, each of the forks F^1X_i, F^2X_i has a partial downward path on the right side of X_i.

Case A-1-i. None of the forks F^1X_i, F^2X_i has a partial upward path on the left side of X_i. Then each of the forks F^1X_i, F^2X_i has a partial upward path on the right side of X_i, as otherwise there would be no temporal path of length 10 from s_i to t_i^1 or t_i^2. For the same reason, the base BX_i has a partial upward path on the right side of X_i. Therefore we can remove the label "5" from the left bridge edge $e_i f_i^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case A-1-ii. Exactly one of the forks F^1X_i, F^2X_i (say F^1X_i) has a partial upward path on the left side of X_i. Then the fork F^2X_i has a partial upward path on the right side of X_i. Furthermore the base BX_i has a partial upward path on the right side of X_i, since otherwise there would be no temporal path of length 10 from s_i to t_i^2. In this case we can modify the solution as follows: remove the labels "1,2,3,4,5" from the partial right-upward path of BX_i and add the labels "6,7,8,9,10" to the partial left-upward path of the fork F^2X_i. Finally we can remove the label "5" from the right bridge edge $\overline{e_i}\overline{f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case A-1-iii. Each of the forks F^1X_i, F^2X_i has a partial upward path on the left side of X_i. In this case we can modify the solution as follows: remove the labels "10,9,8,7,6" from the partial right-downward path of BX_i and add the same labels "10,9,8,7,6" to the partial left-downward path of the base BX_i. Finally we can remove the label "5" from the right bridge edge $\overline{e_i f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case A-2. The base BX_i of X_i has a partial downward path on the left side of X_i; that is, there is a temporal path from vertex e_i to vertex s_i with labels "6,7,8,9,10".
Case A-2-i. None of the forks F^1X_i, F^2X_i has a partial upward path on the left side of X_i. Then the base BX_i and each of the forks F^1X_i, F^2X_i have a partial upward path on the right side of X_i, as otherwise there would be no temporal paths of length 10 from s_i to t_i^1, t_i^2. Moreover, as none of F^1X_i, F^2X_i has a partial left-upward path, it follows that each of F^1X_i, F^2X_i has a partial downward path on the right side of X_i. Indeed, otherwise there would be no temporal paths of length 10 between t_i^1 and t_i^2. In this case we can modify the solution as follows: remove the labels "1,2,3,4,5" from the partial left-upward path of BX_i and add the labels "6,7,8,9,10" to the partial right-upward path of the fork F^3X_i. Finally we can remove the label "6" from the left bridge edge $e_i f_i^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.
Case A-2-ii. Exactly one of the forks F^1X_i, F^2X_i (say F^1X_i) has a partial upward path on the right side of X_i. Then the fork F^2X_i has a partial upward path on the left side of X_i. Furthermore the base BX_i must have a partial right-upward path, as otherwise there would be no temporal path from s_i to t_i^2. In this case we can modify the solution as follows: remove the labels "1,2,3,4,5" from the partial right-upward path of BX_i and add the labels "6,7,8,9,10" to the partial left-upward path of the fork F^2X_i. Finally we can remove the label "5" from the right bridge edge $\overline{e_i f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.
Case A-2-iii. Each of the forks F^1X_i, F^2X_i has a partial upward path on the right side of X_i. Then we we can simply remove the label "5" from the right bridge edge $\overline{e_i f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case B. The fork F^3X_i has also a partial upward path on the right side of X_i. That is, F^3X_i has partial upward-left, upward-right, downward-left, and downward-right paths.

Case B-1. The base BX_i of X_i has no partial downward path on the left side of X_i. Then the base BX_i of X_i has a partial downward path on the right side of X_i, as otherwise there would be no temporal path of length 10 from any of t_i^1, t_i^2, t_i^3 to s_i. For the same reason, each of the forks F^1X_i, F^2X_i has a partial downward path on the right side of X_i.

Note that Case B-1 is symmetric to the case where the base BX_i of X_i has no partial right-downward (resp. left-upward, right upward) path.
Case B-1-i. None of the forks F^1X_i, F^2X_i has a partial upward path on the left side of X_i. This case is the same as Case A-1-i.
Case B-1-ii. Exactly one of the forks F^1X_i, F^2X_i (say F^1X_i) has a partial upward path on the left side of X_i. Then both the base BX_i and the fork F^2X_i has a partial right-upward path, as otherwise there would be no temporal path of length 10 from s_i to t_i^2. In this case, we can always remove the label "6" from the left bridge edge $e_i f_i^3$ of the fork F^3X_i (without compromising the temporal connectivity), thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.
Case B-1-iii. Each of the forks F^1X_i, F^2X_i has a partial upward path on the left side of X_i. That is, each of F^1X_i, F^2X_i has a partial left-upward and a partial right-downward path. The following subcases can occur:
Case B-1-iii(a). None of the forks F^1X_i, F^2X_i has a partial right-upward path. Then each of the forks F^1X_i, F^2X_i has a partial left-downward path, since otherwise there would not exist temporal paths of length 10 between t_i^1 and t_i^2. Furthermore, the base BX_i has a

partial left-upward path, since otherwise there would not exist a temporal path of length 10 from s_i to t_i^1 and t_i^2. In this case, we can remove the label "6" from the right bridge edge $\overline{e_i f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case B-1-iii(b). Exactly one of the forks F^1X_i, F^2X_i (say F^1X_i) has a partial right-upward path. Then the base BX_i has a partial left-upward path, since otherwise there would not exist a temporal path of length 10 from s_i to t_i^2. Similarly, the fork F^1X_i has a partial left-downward path, since otherwise there would not exist a temporal path of length 10 from t_i^1 to t_i^2. In this case we can modify the solution as follows: First, remove the labels "10,9,8,7,6" from the partial right-downward path of BX_i and add the labels "10,9,8,7,6" to the partial left-downward path of BX_i. Second, remove the labels "5,6" from each of t two right bridge edges $\overline{e_i f_i}^1$ and $\overline{e_i f_i}^3$ of the forks F^1X_i and F^3X_i, respectively. Third, remove the label "5" from the right bridge edge $\overline{e_i f_i}^1$ of the fork F^2X_i. Finally, add the five labels "5,4,3,2,1" to the partial left-downward path of the fork F^2X_i. The resulting labeling λ_ϕ^* still preserves the temporal reachabilities and has the same number of labels as λ_ϕ, while the variable gadget X_i is aligned.

Case B-1-iii(c). Each of the forks F^1X_i, F^2X_i has a partial right-upward path. In this case, we can always remove the label "5" from the left bridge edge $e_i f_i^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Case B-2. The base BX_i of X_i has partial left-downward, right-downward, left-upward, and right-upward paths. Then, due to symmetry, we may assume w.l.o.g. that the fork F^1X_i has a left-upward path. Suppose that F^1X_i has also a left-downward path. In this case we can modify the solution as follows: remove the labels "1,2,3,4,5" and "10,9,8,7,6" from the partial right-upward and right-downward paths of BX_i and add the labels "6,7,8,9,10" and "5,4,3,2,1" to the partial left-upward and left-downward paths of the fork F^2X_i. Finally we can remove the label "6" from the right bridge edge $\overline{e_i f_i}^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Finally suppose that F^1X_i has no partial left-downward path. Then F^1X_i has a partial right-down path, since otherwise there would not exist any temporal path of length 10 from t_i^1 to s_i. Similarly, the fork F^2X_i has a partial right-upward path, since otherwise there would not exist any temporal path of length 10 from t_i^1 to t_i^2. In this case we can modify the solution as follows: First remove the labels "1,2,3,4,5" and "10,9,8,7,6" from the partial left-upward and left-downward paths of BX_i. Second add the labels "6,7,8,9,10" to the partial right-upward path of the fork F^1X_i and add the labels "5,4,3,2,1" to the partial right-downward path of the fork F^2X_i. Finally remove the label "6" from the left bridge edge $e_i f_i^3$ of the fork F^3X_i, thus obtaining a labeling with fewer labels than λ_ϕ, a contradiction.

Summarizing, starting from an optimum λ_ϕ of G_ϕ, in which at least one variable gadget is neither left-aligned nor right-aligned, we can modify λ_ϕ to another labeling λ_ϕ^*, such that λ_ϕ^* has one more variable-gadget that is aligned and $|\lambda_\phi| = |\lambda_\phi^*|$. Note that this modification can only happen in Case B-1-iii(b); in all other cases our case analysis arrived at a contradiction. Note here that, by making the above modifications of λ_ϕ, we need to also appropriately modify the "connecting edges" (within the variable gadgets) and the "variable edges" (between different variable gadgets), without changing the total number of labels in each of these edges. Finally, it is straightforward that all modifications of λ_ϕ can be done in polynomial time. This concludes the proof. ◀

▶ **Theorem 18.** *If $OPT_{\mathrm{MAL}}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$ then $OPT_{\mathrm{MonMaxXOR(3)}}(\phi) \geq k$, where n is the number of variables in the formula ϕ.*

APPENDIX

Proof. Recall by Lemma 15 that $d_\phi = 10$. Let λ_ϕ be an optimum solution to $\mathrm{MAL}(G_\phi, 10)$, which uses $\mathrm{OPT}_{\mathrm{MAL}}(G_\phi, d_\phi) \leq \frac{13}{2}n^2 + \frac{99}{2}n - 8k$ labels by the assumption of the theorem. We will prove that there exists a truth assignment τ that satisfies at least k clauses of ϕ. Recall that, since ϕ is an instance of MonMaxXOR(3) with n variables, it has $m = \frac{3}{2}n$ clauses.

Let X_i and X_j be two variable gadgets in G_ϕ. First we observe that the temporal path from a starting vertex s_i of X_i, to any of the ending vertices t_i^ℓ, where $\ell \in \{1, 2, 3\}$, must only go through the vertices and edges of the variable gadget X_i. This is true since in any other case the temporal path would use at least one variable edge and in this case the distance of the path would increase by at least one. Therefore, the path would be of length at least 11, but since the diameter of the graph is 10, the largest label that is allowed to be used is 10 and thus the longest temporal path can use at most 10 edges. Similarly it holds for temporal paths from the ending vertices t_i^ℓ ($\ell \in \{1, 2, 3\}$) to the starting vertex s_i and the temporal paths among the ending vertices. Even more, these temporal paths must be either all on the left or all on the right side of X_i, i.e., they have to use vertices and edges that are all on the left or the right side of the base BX_i and each fork $F^\ell X_i$. This holds as paths of any other form (i.e., containing vertices and edges of both sides) are of length at least 11. Consequently, to label a (s_i, t_i^1)-path in both directions any labeling must use at least $2 \cdot 10$ labels. Now, to label (s_i, t_i^2) and (s_i, t_i^3)-paths, the labels on the base BX_i can be reused, which produces additional 10 labels on each fork F^2X_i and F^3X_i. In the case when all these labels were used on the same path of the variable gadget i.e., all labels were placed on the left or on the right side of BX_i and F^iX_i, there are also temporal paths connecting all three ending vertices, without having to introduce any extra labels. The only missing part is to assure that also all the vertices from the opposite side (i.e., if the labeling used left paths, then the opposite vertices are on the right side, or vice versa) are able to reach and be reached by any other vertex. Therefore, we need at least 2 more labels (one for incoming and one for outgoing temporal paths) on the edges connecting them with the path (vertices) on the other side. Altogether, to ensure the existence of a temporal path between any two vertices from X_i, a labeling must use at least 74 labels on a variable gadget X_i.

Now, let X_i and X_j be such variable gadgets that do not share the fork. As observed above, all vertices from each variable gadget can only be reached among each other, without using the variable edges. Therefore, the variable edges must be labeled in such a way, that they ensure a temporal path among vertices from different variable gadgets. W.l.o.g. we can assume that X_i is left-aligned and X_j is right-aligned by λ_ϕ (all the other cases of aligned and non-aligned labelings of X_i and X_j by λ_ϕ, are symmetric). Since the starting vertex s_i is on the distance 10 from the ending vertices of $t_j^{\ell'}$ ($\ell' \in \{1, 2, 3\}$) of X_j, there must be a temporal path using all labels, to connect them. This path must use the edge of the form $e_i\overline{f_j}^{\ell'}$, as any other path is longer than 10. Since the path must be traversed in both direction each edge $e_i\overline{f_j}^{\ell'}$ ($\ell' \in \{1, 2, 3\}$) must have at least 2 labels. Similarly it holds for the (s_j, t_i^ℓ)-paths ($\ell \in \{1, 2, 3\}$) and the edges $\overline{e_j}f_i^\ell$ ($\ell \in \{1, 2, 3\}$). For a vertex s_i to reach s_j we must label the edge $d_i\overline{d_j}$, as any other (s_i, s_j)-path is longer than 10. Therefore, we need at least one extra label for the edge $d_i\overline{d_j}$. Altogether, to ensure the existence of a temporal path among two vertices from two variable gadgets that do not share a fork, a labeling must use at least 13 labels on the variable edges.

Lastly, let X_i and X_j be two variable gadgets that share a fork. W.l.o.g. we can suppose that X_i is left-aligned by the optimum labeling λ_ϕ and that $F^rX_i = F^{r'}X_j$. By the construction of G_ϕ, there exists a temporal path to and from all the vertices in the fork $F^rX_i = F^{r'}X_j$ to all vertices in X_i and X_j, as there is a temporal path among all vertices from X_i and a temporal path among all vertices in X_j. As observed above, these paths do not use the variable edges, but the variable edges must be labeled in such a way, that they ensure a temporal path among vertices from different variable gadgets. Now if we suppose that the variable gadget X_j is right-aligned by the labeling λ_ϕ, then a temporal path between s_i and s_j must use the edge $d_i\overline{d_j}$ and therefore at least one extra label is used for this edge. A temporal path between s_i and $t_j^{\ell'}$, where $\ell' \in \{1,2,3\} \setminus \{r'\}$, must use the edge $e_i\overline{f_j}^{\ell'}$. Since the edge of this form is traversed in both directions it must have at least two labels. Similarly it holds for the temporal paths between t_i^{ℓ} ($\ell \in \{1,2,3\} \setminus \{r\}$) and s_j. Altogether, to ensure the existence of a temporal path among any two vertices from two variable gadgets that share a fork, a minimum labeling must use at least 9 labels on the variable edges. Similarly we can see that also all other combinations of aligned and non-aligned labelings of X_i and X_j by λ_ϕ, require at least 9 labels on the variable edges.

The only thing left to study, in the case of two variable gadgets that share a fork, is what happens in the intersecting fork. By Lemma 17 we know that the variable gadgets X_i and X_j are aligned by the labeling λ_ϕ. Suppose that $F^r X_i = F^{r'} X_j$. W.l.o.g. we can assume that X_i is left-aligned. We distinguish the following two cases.

- The variable gadget X_j is right-aligned. Then, by the construction of G_ϕ, the fork $F^1 X_i = F^1 X_j$ is labeled using the same labeling, i.e., the left labeling of the variable gadget X_i coincides with the right labeling of the variable gadget X_j. This "saves" 16 labels from the total number of labels used on variable gadgets X_i and X_j.
- The variable gadget X_j is left-aligned. In this case all edges in the fork $F^1 X_i = F^1 X_j$ admit two labels. This "saves" 8 labels from the total number of labels used on variable gadgets X_i and X_j, since both labelings coincide on the connecting edges.

From the labeling λ_ϕ of G_ϕ we construct a truth assignment τ of ϕ in the following way. If a variable gadget X_i is left-aligned, we set x_i to TRUE and if it is right-aligned, we set x_i to FALSE. Using the results from above we deduce that the truth assignment τ satisfies at most k clauses. ◀

Since MAL is clearly in NP, the next theorem follows directly by Theorems 14, 16, and 18.

▶ **Theorem 19.** 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 20.** MSL *is NP-complete.*

Proof. 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 [25].

APPENDIX

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., $n_0 n_1$,
- 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 6. 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.

We claim 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.

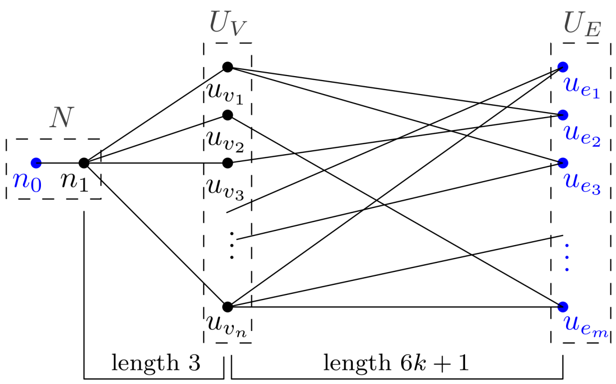


Figure 6 Example of a canonical layering of a directed acyclic graph (DAG).

($\Rightarrow$): Assume (G, k) is a YES instance of the Vertex Cover and let $S \subseteq V(G)$ be a vertex cover for G of size k. We construct a labeling λ for G^* that uses k^* labels and admits a temporal path between all vertices from R^* as follows.

For the sake of easier explanation we use the following terminology. A temporal path starting at n_0 and finishing at some u_e is called a *returning path*. Contrarily, a temporal path from some u_e to n_0 is called a *forwarding path*.

Let U_S be the set of corresponding vertices to S in G^*. From each edge vertex u_e there exists a path of length $6k + 1$ to at least one vertex $u_v \in U_S$, since S is a vertex cover in G. We label exactly one of these paths, using labels $1, 2, \ldots, 6k + 1$. Since S is of size k,

APPENDIX

this part uses $k(6k+1)$ labels. Now we label a path from each $v \in U_S$ to n_1 using labels $6k+2, 6k+3, 6k+4$. Each path uses 3 labels, and since S is of size k we used $3k$ labels for all of them. At the end we label the edge n_0n_1 with the label $\ell^* = 6k+5$. Using this procedure we have created a forwarding path from each edge vertex u_e to the start vertex n_0 and we used $3k + m(6k+1) + 1$ labels.

To create the returning paths, we label paths from n_1 to each vertex in U_S with labels $\ell^*+1, \ell^*+2, \ell^*+3$. Now again, we label exactly one path from vertices in U_S to each edge-vertex u_e, using labels $\ell^*+4, \ell^*+5, \dots, \ell^*+3+6k$. We used extra $3k + m(6k+1)$ labels and created a returning path from n_0 to each vertex in U_E.

All together, the constructed labeling uses $k^* = 6k + 2m(6k+1) + 1$ labels, the only thing left to show is that there exists a temporal path between any pair of edge-vertices $u_e, u_f \in U_E$. It is not hard to see that this holds, as we can construct a temporal path between two edge-vertices as a union of a (sub)path of a temporal path from the first edge-vertex to the starting vertex n_0 and a (sub)path of a temporal path from the starting vertex to the other edge-vertex.

($\Leftarrow$): Assume that (G^*, R^*, k^*) is a YES instance of the MSL. We construct a vertex cover of size at most k for G as follows.

Let us first observe the following, a forwarding and returning path between the starting vertex n_0 and the same edge-vertex u_e, can intersect in at most one time edge. Even more, two temporal paths between the same pair of vertices, going in the opposite directions, intersect in at most one time edge.

By the construction of G^* each (temporal) path between n_0 and a vertex in U_E passes through the set U_V. Since there are m vertices in U_E and each path between a vertex $u_e \in U_E$ and some $u_v \in U_V$ is of length $6k+1$, we need at least $m(6k+1)$ labels to connect U_E to U_V in "one direction". Using the observation from above, we get that there can be at most 1 time edge in common between any two temporal paths among any pair of edge-vertices, therefore we need at least $2m(6k+1) - 1$ labels for paths in both directions. We call these the forwarding path F_e (from u_e to some u_v) and the returning path R_e (from some $u_{v'}$ to u_e) for u_e. It is straightforward to check that every u_e can have at most one forwarding path and one returning path, since every additional path would require at least an additional $6k$ labels and then no connection between n_0 and U_V would be possible.

All labeled temporal paths between N and U_V can be split into two sets, one containing all temporal paths that are a part of (or can be extended to) some returning path, denote them P_N^+ and the others which are a part of (or can be extended to) some forwarding path, denote them P_N^-. It is not hard to see that each temporal path from P_N^+ or P_N^- starts and ends in $N \cup U_V$, i.e., no temporal path starts/ends in one of the dummy vertices. Therefore each temporal path in P_N^+ or P_N^- uses 3 labels. Again, using the above observation we get that temporal paths from P_N^+ and P_N^- share at most one label. Since this part uses at most $6k+1$ labels, there are at most $2k$ temporal paths in P_N^+ and P_N^-. Suppose that $|P_N^+| \leq |P_N^-|$ (the case where $|P_N^+| > |P_N^-|$ is analogous). Let $U_S \subseteq U_V$ be the set of vertices in U_V such that $P_N^+ \cap U_S \neq \emptyset$, i.e., U_S consists of vertices that are endpoints of temporal paths in P_N^+. We claim that $S = \{v \mid u_v \in U_S\}$ is a vertex cover of G and $|S| \leq k$. It is not hard to see that $|P_N^+| \leq k$ and therefore $|S| \leq k$.

We first make the following observation. We define a partial order on the set $\mathcal{P} = \{F_e, R_e \mid e \in E\}$ of forwarding and returning paths as follows. For two paths $P, Q \in \mathcal{P}$, we say that $P < Q$ if all labels used in P are strictly smaller than the smallest label used in Q. We can assume w.l.o.g. that the defined ordering is a total ordering on $\mathcal{P}$ since we can order incomparable path pairs arbitrarily by modifying the labels in a way that does not change the size and the connectivity properties of the labeling. Furthermore, we can observe that for any two $e, e' \in E$ with $e \neq e'$ we have that $F_e < R_{e'}$ since in order for u_e to reach $u_{e'}$, the path F_e needs to be used before the path $R_{e'}$. It follows that there is at most one edge $e \in E$ such that $R_e < F_e$, otherwise we would reach a contradiction to the above observation.

Now assume for contradiction that S is not a vertex cover of G. Then there is an edge $e = \{v, w\} \in E$ such that $\{v, w\} \cap S = \emptyset$. To reach u_e from n_0 there needs to be an edge $e' = \{v, w'\}$ (or symmetrically $\{w, w'\}$) such that we can reach $u_{w'}$ from n_1 via some path P, then continue to $u_{e'}$ using $R_{e'}$, then continue to u_v using $F_{e'}$, finally reach u_e using R_e. Notice that this requires $P < R_{e'} < F_{e'} < R_e$. This implies that the path from n_0 to u_e cannot be longer since otherwise there would be two edges e', e'' with $R_{e'} < F_{e'}$ and $R_{e''} < F_{e''}$, a contradiction. It also implies that edge e is the only edge in E with $e \cap S = \emptyset$.

Now consider an edge $e'' = \{w', v''\} \neq e'$ such that there is no direct path from n_0 to $u_{v''}$. If such an edge does not exist then w' and all of its neighbors , different than v, are in S. Hence we can remove w' from S and add v to S to obtain a vertex cover for G of size at most k. Assume that edge e'' with the described properties exists and consider the temporal path from $u_{e'}$ to $u_{e''}$. This path must start with $F_{e'}$ thus reaching u_v. From there the path cannot continue to some $u_{e'''}$ since for all $e''' \neq e'$ we have that $F_{e'''} < R_{e'''}$ hence the path cannot continue from $u_{e'''}$. It follows that the path has to eventually reach n_1 continue to $u_{w'}$ from there. However, recall that $P < F_{e'}$ which means that we cannot use P to reach $u_{w'}$ from n_1. Hence, there is a second temporal path P' (using the same edges as P with later labels) from n_1 to $u_{w'}$ with $F_{e'} < P'$. This implies that $|S| < k$ and we can add v to S to obtain a vertex cover of size at most k for G. ◀

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 7. 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 21.** *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 21 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 7 and potentially adding an extra edge to form a C_4; we then argue that the

APPENDIX

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.

Proof. Assume there is a labeling λ for G that labels all edges in the subgraph H of G. We describe a procedure to transform H into a tree T by removing edges from H such that T can be labeled with k labels such that all vertices in R are pairwise temporally connected.

Consider a terminal vertex $r^* \in R$. Let $H^+_{r^*}$ be a minimum subgraph of H and $\lambda^+_{r^*}$ a minimum sublabeling of λ for $H^+_{r^*}$ such that r^* can temporally reach all vertices in $R \setminus \{r^*\}$ in $(H^+_{r^*}, \lambda^+_{r^*})$. Let us first observe that $H^+_{r^*}$ is a tree where all leafs are vertices from R and $\lambda^+_{r^*}$ assigns exactly one label to every edge in $H^+_{r^*}$.

First note that all vertices in $(H^+_{r^*}, \lambda^+_{r^*})$ are temporally reachable from r^*. If a vertex is not reachable, we can remove it, a contradiction to the minimality of $H^+_{r^*}$. Now assume that $H^+_{r^*}$ is not a tree. Then there is a vertex $v \in V(H^+_{r^*})$ such that v is temporally reachable from r^* in $(H^+_{r^*}, \lambda^+_{r^*})$ via two temporal paths P, P' that visit different vertex sets, i.e. $V(P) \neq V(P')$. Assume w.l.o.g. that both P and P' are foremost among all temporal paths that visit the vertices in $V(P)$ and $V(P')$, respectively, in the same order. Let the arrival time of P be at most the arrival time of P'. Then we can remove the last edge traversed by P' with all its labels from $(H^+_{r^*}, \lambda^+_{r^*})$ such that afterwards r^* can still temporally reach all vertices in $R \setminus \{r^*\}$, a contradiction to the minimality of $H^+_{r^*}$. From now on, assume that $H^+_{r^*}$ is a tree. Assume that $H^+_{r^*}$ contains a leaf vertex v that is not contained in R. Then we can remove v from $(H^+_{r^*}, \lambda^+_{r^*})$ such that afterwards r^* can still temporally reach all vertices in $R \setminus \{r^*\}$, a contradiction to the minimality of $H^+_{r^*}$. Lastly, assume that there is an edge $e = uv$ in $H^+_{r^*}$ such that $\lambda^+_{r^*}$ assigns more than one label to e. Let v be further away from r^* than u in $H^+_{r^*}$ and let P be a foremost temporal path from r^* to v in $(H^+_{r^*}, \lambda^+_{r^*})$ with arrival time t. Then we can remove all labels except for t from e and afterwards r^* can still temporally reach all vertices in $R \setminus \{r^*\}$, a contradiction to the minimality of $\lambda^+_{r^*}$.

Let $H^-_{r^*}$ be a minimum subgraph of H and $\lambda^-_{r^*}$ a minimum sublabelling of λ for $H^-_{r^*}$ such that each vertex in $R \setminus \{r^*\}$ can temporally reach r^* in $(H^-_{r^*}, \lambda^-_{r^*})$. We can observe by analogous arguments as above that $H^-_{r^*}$ is a tree where all leafs are vertices from R and $\lambda^-_{r^*}$ assigns exactly one label to every edge in $H^-_{r^*}$.

We define the following sets of edges:

- The set of edges only appearing in $H^+_{r^*}$: $E^+_{r^*} = E(H^+_{r^*}) \setminus E(H^-_{r^*})$.
- The set of edges only appearing in $H^-_{r^*}$: $E^-_{r^*} = E(H^-_{r^*}) \setminus E(H^+_{r^*})$.
- The set of edges appearing in both $H^+_{r^*}$ and $H^-_{r^*}$: $E^{+-}_{r^*} = E(H^+_{r^*}) \cap E(H^-_{r^*})$.
- The set of edges appearing in both $H^+_{r^*}$ and $H^-_{r^*}$ that receive the same label from $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$: $E^*_{r^*} = \{e \in E^{+-}_{r^*} \mid \lambda^+_{r^*}(e) = \lambda^-_{r^*}(e)\}$.

We claim that there exists a labelling λ' of size k for G such that there are two trees $H^+_{r^*}, H^-_{r^*}$ with the above described properties and $|E(H^+_{r^*})| + |E(H^-_{r^*})| - |E^*_{r^*}| = k - x$ for some $x \geq 0$ and

- $|E^*_{r^*}| \leq x + 1$ if k is odd, and
- if k is even, then $|E^*_{r^*}| \leq x + 2$ and there exist two edges e^+, e^- in H that each of them, when added to $H^+_{r^*}, H^-_{r^*}$, respectively, creates a C_4 in $H^+_{r^*}, H^-_{r^*}$, respectively.

We first argue that the statement of the lemma follows from this claim. Afterwards we prove the claim. Assume that $|E^+_{r^*}| \leq |E^-_{r^*}|$ (the case where $|E^+_{r^*}| > |E^-_{r^*}|$ is analogous).

Assume that $|E^*_{r^*}| \leq x + 1$. Then we clearly have

$$2|E(H^+_{r^*})| - 1 = 2|E^+_{r^*}| + 2|E^{+-}_{r^*}| - 1 \leq |E(H^+_{r^*})| + |E(H^-_{r^*})| - 1 = k - x + |E^*_{r^*}| - 1 \leq k.$$

APPENDIX

It follows that we can temporally label $H^+_{r^*}$ with at most k labels such that all vertices in $H^+_{r^*}$ can pairwise temporally reach each other, using the result that trees with m edges can be temporally labeled with $2m-1$ labels (see Theorem 7). Since we assume $(G, R, k-1)$ is a NO instance of MSL it follows that $k = 2m-1$ and hence this can only happen if k is odd.

Assume that $|E^*_{r^*}| \leq x+2$ and there exist two edges e^+, e^- in H that each of them, when added to $H^+_{r^*}, H^-_{r^*}$, respectively, creates a C_4 in $H^+_{r^*}, H^-_{r^*}$, respectively.. Then we clearly have

$$2|E(H^+_{r^*}) \cup \{e^+\}| - 4 = 2|E^+_{r^*}| + 2|E^{+-}_{r^*}| - 2 \leq |E(H^+_{r^*})| + |E(H^-_{r^*})| - 2 = k - x + |E^*_{r^*}| - 2 \leq k.$$

It follows that we can temporally label $H^+_{r^*}$ together with edge e^+ with at most k labels such that all vertices in $H^+_{r^*}$ with edge e^+ can pairwise temporally reach each other, using the result that graphs containing a C_4 with n vertices can be temporally labeled with $2n-4$ labels (see Theorem 7). Since we assume $(G, R, k-1)$ is a NO instance of MSL it follows that $k = 2n-4$ and hence this can only happen if k is even.

Now we prove that there exists a labeling λ' of size k for G such that there are two trees $H^+_{r^*}, H^-_{r^*}$ with the above described properties and $|E(H^+_{r^*})| + |E(H^-_{r^*})| - |E^*_{r^*}| = k - x$ for some $x \geq 0$ and $|E^*_{r^*}| \leq x+1$.

Let $H^+_{r^*}, H^-_{r^*}$ be two trees with the above described properties and $|E(H^+_{r^*})| + |E(H^-_{r^*})| - |E^*_{r^*}| = k - x$ for some $x \geq 0$. We will argue that by slightly modifying the labeling λ (and with that $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$, that way ultimately obtaining λ') and $H^+_{r^*}, H^-_{r^*}$, we achieve that $|E(H^+_{r^*})| + |E(H^-_{r^*})| - |E^*_{r^*}| = k - x'$ for some $x' \geq 0$ and either $|E^*_{r^*}| \leq x'+1$ or $|E^*_{r^*}| \leq x'+2$. We will argue that in the former case we must have that k is odd, and in the latter case we must have that k is even. Note that if $|E^*_{r^*}| = 1$ we are done, hence assume from now on that $|E^*_{r^*}| \geq 2$.

We consider several cases. For the sake of presentation of the next cases, define the *head* of a temporal path as the last vertex visited by the path and the *extended head* of a temporal path as the last two vertices visited by the path. Furthermore, define the *tail* of a temporal path as the first vertex visited by the path and the *extended tail* of a temporal path as the first two vertices visited by the path.

Case A. Assume there is a temporal path P from r^* to some $r \in R \setminus \{r^*\}$ in $H^+_{r^*}$ that traverses two edges in $E^*_{r^*}$. Let $e, e' \in E^*_{r^*}$ with $e \neq e'$ such that there is a temporal path P from r^* to some $r \in R \setminus \{r^*\}$ in $H^+_{r^*}$ that traverses w.l.o.g. first e and then e' and a *maximum number* α of edges lies between them in P and the distance β between r^* and e is minimum. Note that this implies that $\lambda^+_{r^*}(e) < \lambda^+_{r^*}(e')$.

In the following we analyse several cases. In some of them we can deduce that the labeling λ must use labels that are not present in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$ that are unique to that case. This implies that for each of these cases we can attribute one label outside of $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ to edge e or e'.

In some other cases we describe modifications that do not increase $|E(H^+_{r^*}) \cup E(H^-_{r^*})|$ and either

- strictly decrease β, or
- strictly decrease α and not increase β, or
- strictly decrease $|E^*_{r^*}|$ and not increase α and β,

while preserving that

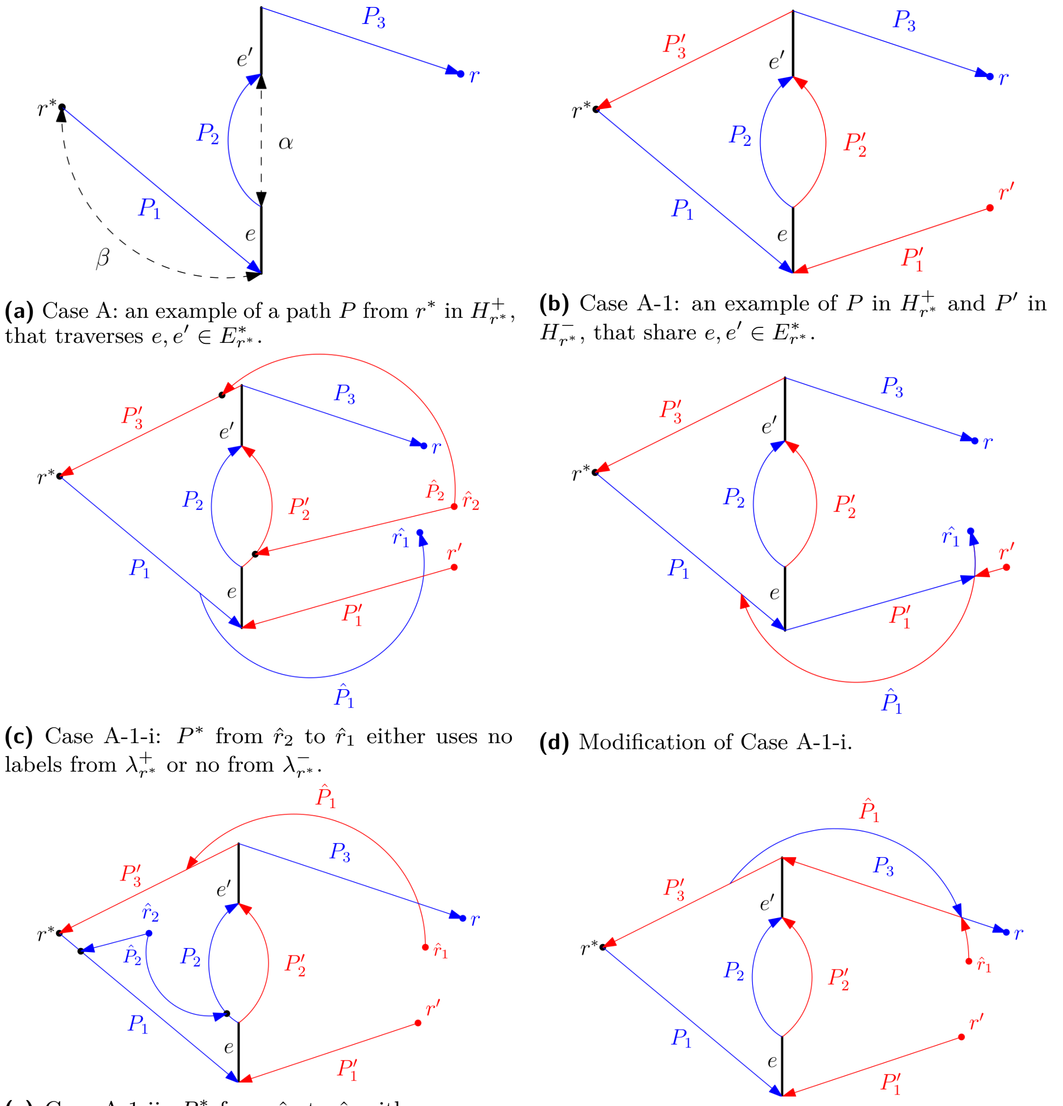
- $H^+_{r^*}$ and $H^+_{r^*}$ are trees with leafs in R, and
- $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ assign at most one label per edge.

Whenever a modification satisfies the above requirements it is clear that it can only be applied a finite number of times. Whenever we describe a case that requires modifications that do not satisfy the above requirements, we explicitly show that these modifications can

APPENDIX

only be applied a finite number of times as well. Overall this then shows that after a finite number of modifications, none of the described cases will apply.

We partition the temporal path P into the part P_1 from r^* to e, the part consisting of e itself, the part P_2 between e and e', the part consisting of e' itself, and the part P_3 from e' to r. Now in $H^-_{r^*}$ we can have two different scenarios. For illustrations of all variations of Case A see Figures 7–9.



(a) Case A: an example of a path P from r^* in $H^+_{r^*}$, that traverses $e, e' \in E^*_{r^*}$.

(b) Case A-1: an example of P in $H^+_{r^*}$ and P' in $H^-_{r^*}$, that share $e, e' \in E^*_{r^*}$.

(c) Case A-1-i: P^* from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

(d) Modification of Case A-1-i.

(e) Case A-1-ii: P^* from $\hat{r}_1$ to $\hat{r}_2$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

(f) Modification of Case A-1-ii.

Figure 7 Cases A-1 – A-1-ii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

Case A-1. There is a temporal path P' from some $r' \in R \setminus \{r^*\}$ to r^* in $H^-_{r^*}$ that traverses both e and e'. Note that this implies that e is traversed before e'.

We partition the temporal path P' into the part P'_1 from r' to e, the part consisting of e itself, the part P'_2 between e and e', the part consisting of e' itself, and the part P'_3 from e' to r^*.

APPENDIX

The analysis of each one follows from the observation that the labels in P_3' are larger than the ones in P_1.

Case A-1-i. Assume there is a path $\hat{P}_1$ in $H_{r^*}^+$ starting at a vertex that is visited by P_1 and ending at $\hat{r}_1 \in R \setminus \{r^*\}$ such that $\hat{r}_1 = r'$ or $\hat{P}_1$ and P_1' intersect in a vertex. For our analysis, we treat these two cases the same since in both cases we can assume that r' can reach $\hat{r}_1$, in the latter through the intersection point. If there is a path $\hat{P}_2$ in $H_{r^*}^-$ starting at some $\hat{r}_2 \in R \setminus \{r^*, r'\}$ and ending at the extended tail of P_2' or P_3', then the temporal path P^* in (G, λ) from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda_{r^*}^+$ or no from $\lambda_{r^*}^-$.
Case A-1-ii. Assume there is a path $\hat{P}_1$ in $H_{r^*}^-$ starting at $\hat{r}_1 \in R \setminus \{r^*\}$ and ending at a vertex that is visited by P_3', such that $\hat{r}_1 = r$ or $\hat{P}_1$ and P_3 intersect in a vertex. Again for our analysis, we treat these two cases the same since in both cases we can assume that $\hat{r}_1$ can reach r, in the latter through the intersection point. If there is a path $\hat{P}_2$ in $H_{r^*}^+$ starting at the extended tail of P_1 or P_2 and ending at some $\hat{r}_2 \in R \setminus \{r^*, r\}$, then the temporal path P^* in (G, λ) from $\hat{r}_1$ to $\hat{r}_2$ either uses no labels from $\lambda_{r^*}^+$ or no from $\lambda_{r^*}^-$.

Assume that one of the above two applies. We assume that there is no path $\hat{P}_2$ in $H_{r^*}^-$ starting at some $\hat{r}_2 \in R \setminus \{r^*, r'\}$ and ending at the extended tail of P_2' or P_3' in Case A-1-i and that there is no path $\hat{P}_2$ in $H_{r^*}^+$ starting at the extended tail of P_1 or P_2 and ending at some $\hat{r}_2 \in R \setminus \{r^*, r\}$, since in both cases we can directly deduce that we need labels outside of $\lambda_{r^*}^+$ and $\lambda_{r^*}^-$. Then we modify λ in the following way without changing its connectivity properties. First, we scale all labels in λ by a factor of $|V|$.

The idea is first to essentially switch the roles of P_1' and $\hat{P}_1$ in Case A-1-i and switch the roles of P_3 and $\hat{P}_1$ in Case A-1-ii. Assume Case A-1-i applies.

- We remove $\hat{P}_1$'s edges and labels from $H_{r^*}^+$ and $\lambda_{r^*}^+$, respectively, add $\hat{P}_1$'s edges to $H_{r^*}^-$. Add the edges between the (original) tail of $\hat{P}_1$ to e to $H_{r^*}^-$ and add the respective labels for those edges from $\lambda_{r^*}^+$ also to $\lambda_{r^*}^-$. Add new labels for the edges of $\hat{P}_1$ to $\lambda_{r^*}^-$ such that there is temporal paths from r' to r^* that does use edges from P_1'.
- We remove P_1''s edges and labels from $H_{r^*}^-$ and $\lambda_{r^*}^-$, respectively, add P_1''s edges to $H_{r^*}^+$, and add new labels for the edges of P_1' to $\lambda_{r^*}^+$ such that there is a temporal path from r^* to r'.

Now assume Case A-1-ii applies. We make analogous modifications.

- We remove $\hat{P}_1$'s edges and labels from $H_{r^*}^-$ and $\lambda_{r^*}^-$, respectively, add $\hat{P}_1$'s edges to $H_{r^*}^+$. Add the edges from the head of $\hat{P}_1$ to e' to $H_{r^*}^+$ and add the respective labels for those edges from $\lambda_{r^*}^-$ also to $\lambda_{r^*}^+$. Add new labels for the edges of $\hat{P}_1$ to $\lambda_{r^*}^+$ such that there is temporal paths from r^* to r that does use edges from P_3.
- We remove P_3's edges and labels from $H_{r^*}^+$ and $\lambda_{r^*}^+$, respectively, add P_3's edges to $H_{r^*}^-$, and add new labels for the edges of P_3 to $\lambda_{r^*}^-$ such that there are temporal paths from r to r^*.

Note that after the modifications $H_{r^*}^+$ and $H_{r^*}^-$ are still trees, and $\lambda_{r^*}^+$ and $\lambda_{r^*}^-$ still assign at most one label per edge. Furthermore, we have that the modification do not increase the sum of edges in both trees $|E(H_{r^*}^+) \cup E(H_{r^*}^-)|$. Note that these modifications potentially increase $|E_{r^*}^*|$ and α. However, note that in both cases we strictly decrease β. From now on assume that Cases A-1-i and A-1-ii do not apply.

We start with three further subcases. The analysis of each one follows from the observation that the labels in P_3' are larger than the ones in P_1.
Case A-1-iii. Assume there is a path $\hat{P}$ in $H_{r^*}^+$ starting at a vertex that is visited by P_1 but is different from its tail and extended head and ending at some $\hat{r} \in R \setminus \{r^*, r\}$. Then the temporal path P^* in (G, λ) from r' to $\hat{r}$ needs at least one label that is not contained in $\lambda_{r^*}^+$ or $\lambda_{r^*}^-$. More specifically, P^* either uses no labels from $\lambda_{r^*}^+$ or no from $\lambda_{r^*}^-$.

APPENDIX

Case A-1-iv. Assume there is a path $\hat{P}$ in $H^-_{r^*}$ starting at some $\hat{r} \in R \setminus \{r^*, r'\}$ and ending at a vertex that is visited by P'_3 but is different from its extended tail and head. Then the temporal path P^* in (G, λ) from $\hat{r}$ to r needs at least one label that is not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$. More specifically, P^* either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.
Case A-1-v. Assume there is a path $\hat{P}_1$ in $H^+_{r^*}$ starting at a vertex that is visited by P_2 but is different from its tail and extended head and ending at some $\hat{r}_1 \in R \setminus \{r^*, r\}$. Furthermore, assume there is a path $\hat{P}_2$ in $H^-_{r^*}$ starting at some $\hat{r}_2 \in R \setminus \{r^*, r'\}$ and ending at a vertex that is visited by P'_2 but is different from its extended tail and head. Then, if $\hat{r}_2 \neq \hat{r}_1$ and $P_2 \neq P'_2$, or the starting vertex of $\hat{P}_1$ is by at least two edges closer to e than the starting vertex of $\hat{P}_2$, the temporal path P^* in (G, λ) from $\hat{r}_2$ to $\hat{r}_1$ needs at least one label that is not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$. More specifically, P^* either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

In the above three Cases A-1-iii to A-1-v we do not make any modifications, since we can directly deduce that we need labels outside of $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$. For the remainder of this case distinction, we assume that Cases A-1-iii to A-1-v do not apply.

We can further observe the following using analogous arguments as above.
Case A-1-vi. Assume there is a path $\hat{P}_1$ in $H^+_{r^*}$ starting at the extended head of P_1 and ending at some $\hat{r}_1 \in R \setminus \{r^*, r, r'\}$. If there is a path $\hat{P}_2$ in $H^-_{r^*}$ starting at some $\hat{r}_2 \in R \setminus \{r^*, r'\}$ and ending at a vertex from P'_2 that is not its tail or a vertex from P'_3, then, if $\hat{r}_2 \neq \hat{r}_1$, the temporal path P^* in (G, λ) from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.
Case A-1-vii. Assume there is a path $\hat{P}_1$ in $H^-_{r^*}$ starting at some $\hat{r}_1 \in R \setminus \{r^*, r, r'\}$ and ending at the extended tail of P'_3. If there is a $\hat{P}_2$ in $H^+_{r^*}$ starting at a vertex from P_1 or a vertex from P_2 that is not its head and ending at some $\hat{r}_2 \in R \setminus \{r^*, r\}$, then, if $\hat{r}_1 \neq \hat{r}_2$, the temporal path P^* in (G, λ) from $\hat{r}_1$ to $\hat{r}_2$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

First, assume that Case A-1-vi or Case A-1-vii or none of them apply. Then we modify λ in the following way without changing its connectivity properties. First, we scale all labels in λ by a factor of $|V|$.

The idea is first to essentially switch the roles of P_1 and P'_3.

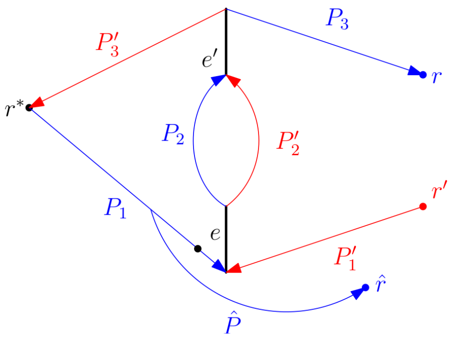
- We remove P_1's edges and labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively, add P_1's edges to $H^-_{r^*}$, and add new labels for the edges of P_1 to $\lambda^-_{r^*}$ such that there are temporal paths from both endpoints of e to r^* that only use the new labels.
- We remove P'_3's edges and labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively, add P'_3's edges to $H^+_{r^*}$, and add new labels for the edges of P'_3 to $\lambda^+_{r^*}$ such that there are temporal paths from r^* to both endpoints of e that only use the new labels.

In both modification above, we assume w.l.o.g. that the smallest and the largest label assigned to an edge of P_1 by $\lambda^+_{r^*}$ before the modification are equal the smallest and the largest label, respectively, assigned to an edge of P'_3 by $\lambda^+_{r^*}$ after the modification. Symmetrically, we assume w.l.o.g. that the smallest and the largest label assigned to an edge of P'_3 by $\lambda^-_{r^*}$ before the modification are equal the smallest and the largest label, respectively, assigned to an edge of P_1 by $\lambda^-_{r^*}$ after the modification. Note that now there is a path from r^* to r in $(H^+_{r^*}, \lambda^+_{r^*})$ that does not use edges e and e'. Furthermore, there is a path from r' to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$ that does not use edges e and e'.

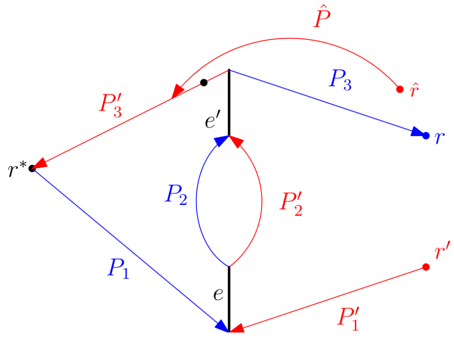
Now we have to adjust labels on e, e', P_2, and P'_2, depending on whether Case A-1-vi, Case A-1-vii or none of them apply.

- If Case A-1-vi applies, then we remove e, e', and the edges of P'_2 and their labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively. Furthermore, we exchange the labels of e and e' and the edges of P_2 assigned by $\lambda^+_{r^*}$ in a way that there is a temporal path from r^* to $\hat{r}_1$ (see Case A-1-vi) in $(H^+_{r^*}, \lambda^+_{r^*})$.

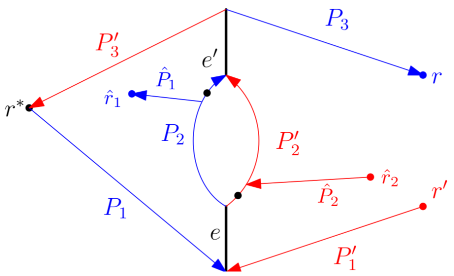
APPENDIX



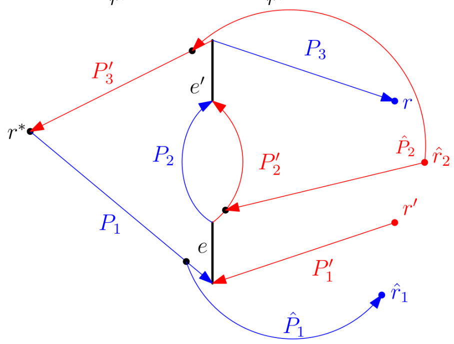
(a) Case A-1-iii: P^* from r' to $\hat{r}$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



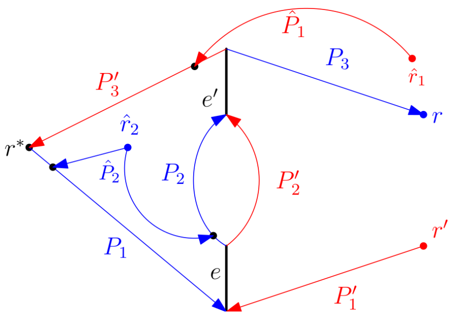
(b) Case A-1-iv: P^* from $\hat{r}$ to r either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



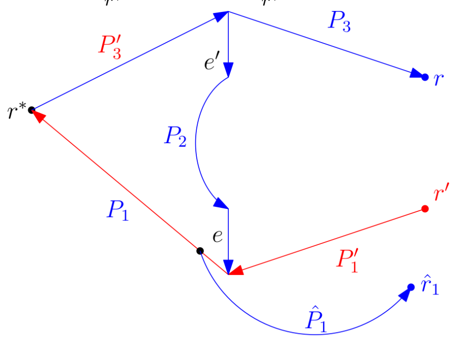
(c) Case A-1-v: P^* from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



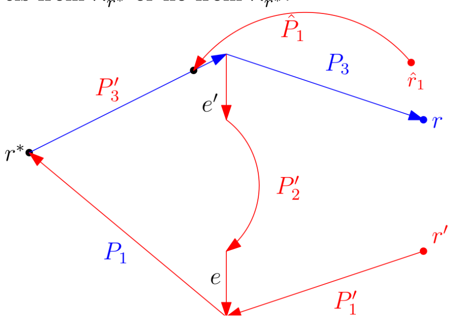
(d) Case A-1-vi: P^* from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



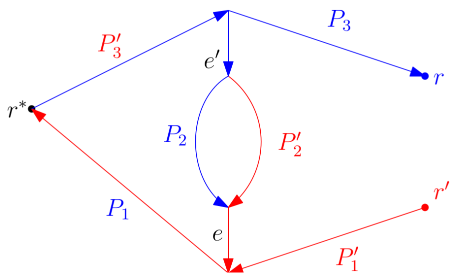
(e) Case A-1-vii: P^* from $\hat{r}_1$ to $\hat{r}_2$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



(f) Modification of Case A-1-vi.



(g) Modification of Case A-1-vii.



(h) Modification when none of the cases A-1-vi nor A-1-vii apply.

Figure 8 Cases A-1-iii – A-1-vii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

APPENDIX

- If Case A-1-vii applies, then we remove e, e', and the edges of P_2 and their labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively. Furthermore, we exchange the labels of e and e' and the edges of P'_2 assigned by $\lambda^-_{r^*}$ in a way that there is a temporal path from $\hat{r}_1$ (see Case A-1-vii) to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$.
- If none of the Cases A-1-vi and A-1-vii apply, then we remove e its labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively, and we remove e' its labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively. We modify the labels of P_2 assigned by $\lambda^+_{r^*}$ is a way that all terminals that were reachable from r^* before the modifications can now be reached via e'. We modify the labels of P'_2 assigned by $\lambda^-_{r^*}$ is a way that all terminals that could reach r^* before the modifications can now reach r^* via e.

Note that after the modifications $H^+_{r^*}$ and $H^-_{r^*}$ are still trees, and $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ still assign at most one label per edge. Furthermore, we have that the modification do not increase the sum of edges in both trees $|E(H^+_{r^*}) \cup E(H^-_{r^*})|$. Lastly, and most importantly, we have that at least one of $H^+_{r^*}$ and $H^-_{r^*}$ does contain both edges e and e'. It follows that we strictly decrease $|E^*_{r^*}|$ without increasing α.

It follows that after exhaustively performing the above modifications we have that if Case A-1 applies, then one of the Cases A-1-iii to A-1-v has to apply.

Case A-2. There are two temporal paths P', P'' from some $r', r'' \in R \setminus \{r^*\}$, respectively, to r^* in $H^-_{r^*}$ such that P' traverses e and P'' traverses e'. We consider several different subcases. Let $e = uv$ and let u be the vertex that is closer to r^* in $H^+_{r^*}$. Partition P' into P'_1 from r' to e, then e, and then P'_2 from e to r^*.

Case A-2-i. Assume the head of P'_1 is v.

We remove e and its labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively. To obtain a new path in $(H^-_{r^*}, \lambda^-_{r^*})$, we traverse P'_1, then traverse P_2 (by modifying $\lambda^-_{r^*}$ on P'_1 accordingly) which lets us reach P'' and then we traverse P'' to reach r^*.

Note that after the modifications $H^-_{r^*}$ is still a tree and $\lambda^-_{r^*}$ still assign at most one label per edge. However, the size of $E^*_{r^*}$ changes, in particular it can increase, but the maximal number α of edges between two edges from $E^*_{r^*}$ in P decreases by one.

Case A-2-ii. Assume the head of P'_1 is u. Assume there is a path $\hat{P}$ in $H^+_{r^*}$ starting at a vertex that is visited by P_1 but is different from its tail and extended head and ending at some $\hat{r} \in R \setminus \{r^*, r\}$, such that $\hat{r} = r'$ or $\hat{P}$ and P'_1 intersect in a vertex. For our analysis, we treat these two cases the same since in both cases we can assume that r' can reach $\hat{r}$, in the latter through the intersection point.

Case A-2-ii(a). Furthermore, assume there is a path $\hat{P}'$ in $H^-_{r^*}$ starting at some $\hat{r}' \in R \setminus \{r^*, r'\}$ and ending at a vertex that is visited by P'_2. Then the temporal path P^* in (G, λ) from $\hat{r}'$ to r' either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

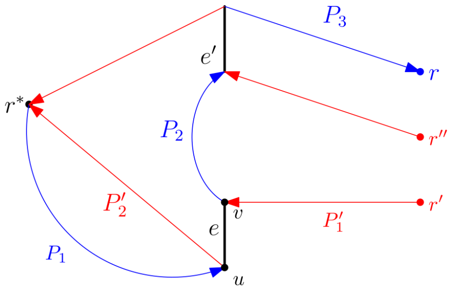
Case A-2-ii(b). Furthermore, assume there is a path $\hat{P}''$ in $H^-_{r^*}$ starting at some $\hat{r}'' \in R \setminus \{r^*, r'\}$ and ending at a vertex that is visited by P''_2. Then the temporal path P^* in (G, λ) from $\hat{r}''$ to r' either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

Assume that Cases A-2-ii(a) and (b) do not apply. Then we modify λ in the following way without changing its connectivity properties. First, we scale all labels in λ by a factor of $|V|$.

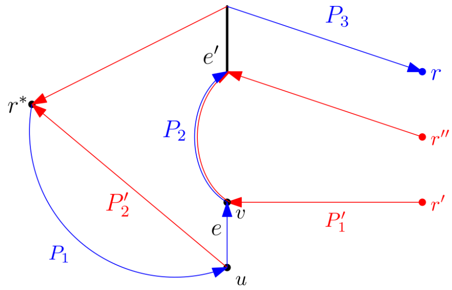
The idea is to essentially switch the roles of $\hat{P}$ and P'_2.

- We remove P_1's and $\hat{P}$'s edges and labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively, add $\hat{P}$'s edges to $H^-_{r^*}$. Add the edges from the tail of $\hat{P}$ to r^* to $H^-_{r^*}$ and add labels for those edges to $\lambda^-_{r^*}$ such that there is a path from r' to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$ that uses the newly added labels.
- We remove P'_1's and P'_2's edges and labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively, add P'_1's and P'_2's edges to $H^+_{r^*}$, and add new labels for the edges of P'_1 and P'_2 to $\lambda^+_{r^*}$ such that there

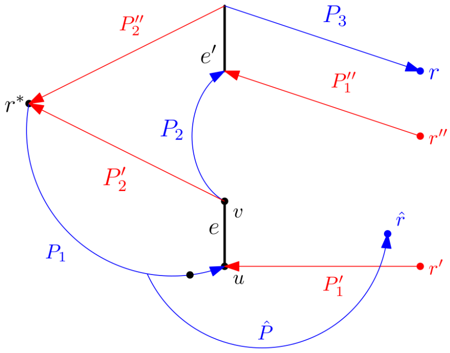
APPENDIX



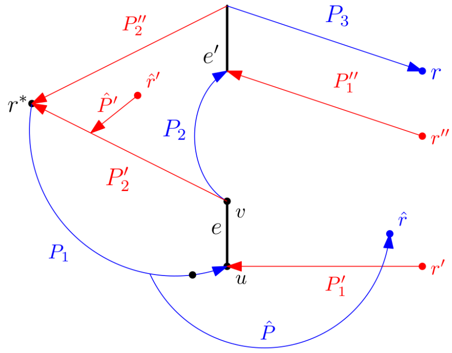
(a) Example of Case A-2-i.



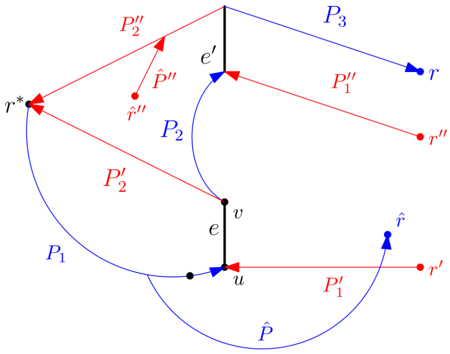
(b) Modification of Case A-2-i.



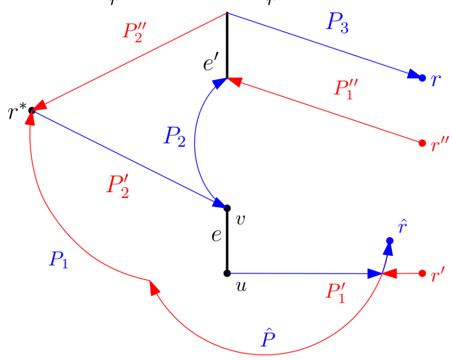
(c) Example of Case A-2-ii.



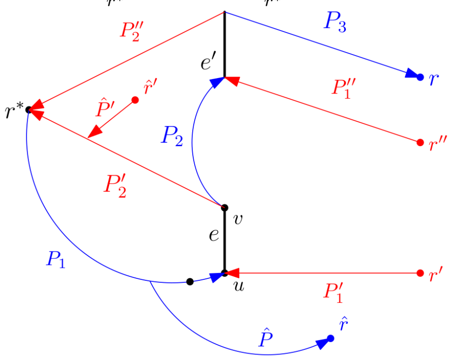
(d) Case A-2-ii(a): P^* from $\hat{r}'$ to r' either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



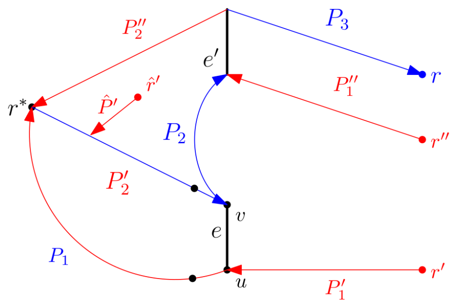
(e) Case A-2-ii(b): P^* from $\hat{r}''$ to r' either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



(f) Modification of Case A-2-ii when A-2-ii(a) and A-2-ii(b) do not apply.



(g) Case A-2-iii: P^* from $\hat{r}'$ to r either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



(h) Modification of Case A-2-iii.

Figure 9 Cases A-2-i – A-2-iii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

APPENDIX

is temporal path from r^* to $\hat{r}$ in $(H^+_{r^*}, \lambda^+_{r^*})$.

Note that after the modifications $H^+_{r^*}$ and $H^-_{r^*}$ are still trees, and $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ still assign at most one label per edge. Furthermore, we have that the modification do not increase the sum of edges in both trees $|E(H^+_{r^*}) \cup E(H^-_{r^*})|$. Lastly, and most importantly, we have that the path from r^* to r in $H^+_{r^*}$ does not contain both edges e and e'. It follows that we decreased α.

Case A-2-iii. Assume the head of P'_1 is u. Assume there is a path $\hat{P}$ in $H^+_{r^*}$ starting at a vertex that is visited by P_1 but is different from its tail and extended head and ending at some $\hat{r} \in R \setminus \{r^*, r, r'\}$. Then the temporal path P^* in (G, λ) from r' to $\hat{r}$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$. Furthermore, assume there is a path $\hat{P}'$ in $H^-_{r^*}$ starting at some $\hat{r}' \in R \setminus \{r^*, r'\}$ and ending at a vertex that is visited by P'_2 but is different from its extended tail and head. Then the temporal path P^* in (G, λ) from $\hat{r}'$ to r either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

We again modify λ in a way that does not change its connectivity properties. First, we scale all labels in λ by a factor of $|V|$. We essentially switch the roles of P_1 and P'_2.

We remove P_1's edges and labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively, add P_1's edges to $H^-_{r^*}$, and add new labels for the edges of P_1 to $\lambda^-_{r^*}$ such that there are temporal paths from both endpoints of e to r^* that only use the new labels. We remove P'_2's edges and labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively, add P'_2's edges to $H^+_{r^*}$, and add new labels for the edges of P'_2 to $\lambda^+_{r^*}$ such that there are temporal paths from r^* to both endpoints of e that only use the new labels.

Note that now there is a path from $\hat{r}'$ to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$ that does not use edge e. Further note that after the modifications $H^+_{r^*}$ and $H^-_{r^*}$ are still trees, and $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ still assign at most one label per edge. Furthermore, we have that the modification do not increase the sum of edges in both trees $|E(H^+_{r^*}) \cup E(H^-_{r^*})|$. It follows that we strictly decrease $|E^*_{r^*}|$ without increasing α.

Now consider the case where we have a temporal path P from some $r \in R \setminus \{r^*\}$ to r^* in $H^-_{r^*}$ that traverses both e and e' and two temporal paths P_1, P_2 from r^* to some $r_1, r_2 \in R \setminus \{r^*\}$, respectively, in $H^+_{r^*}$ such that P_1 traverses e and P_2 traverses e'. This case is analogous to the previously discussed case.

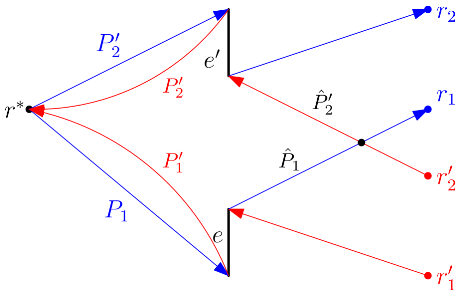
From now on we assume that Case A-2 does not apply.

Case B. From now on we assume that none of the above described cases apply. This means that there is no path from r^* to some $r \in R \setminus \{r^*\}$ in $H^+_{r^*}$ that traverses both e and e' and there is no path from some $r' \in R \setminus \{r^*\}$ to r^* in $H^-_{r^*}$ that traverses both e and e'. It follows that for every $e \in E^*_{r^*}$ we have a path in $H^+_{r^*}$ from r^* to some $r \in R \setminus \{r^*\}$ that only traverses e from the edges in $E^*_{r^*}$ and we have a path in $H^-_{r^*}$ from some $r' \in R \setminus \{r^*\}$ to r^* that only traverses e from the edges in $E^*_{r^*}$. All the following cases are illustrated in Figure 10.

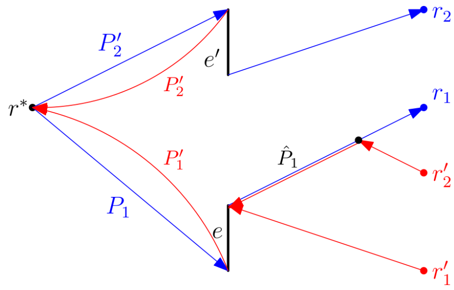
Case B-1. Let $e, e' \in E^*_{r^*}$ and let P_1 be a path in $H^+_{r^*}$ from r^* to some $r_1 \in R \setminus \{r^*\}$ that only traverses e from the edges in $E^*_{r^*}$ and let P_2 be a path in $H^-_{r^*}$ from some $r_2 \in R \setminus \{r^*\}$ to r^* that only traverses e from the edges in $E^*_{r^*}$. Let P'_1 be a path in $H^+_{r^*}$ from r^* to some $r'_1 \in R \setminus \{r^*\}$ that only traverses e' from the edges in $E^*_{r^*}$ and let P'_2 be a path in $H^-_{r^*}$ from some $r'_2 \in R \setminus \{r^*\}$ to r^* that only traverses e' from the edges in $E^*_{r^*}$.

Consider the case where all choices of P_1, P_2, P'_1, P'_2 with the above properties we have $r_1 = r'_2$ or P_1 and P'_2 intersect in a vertex after they traversed e and e', respectively. Again for our analysis, we treat these two cases the same since in both cases we can assume that r'_2 can reach r_1, in the latter through the intersection point. The case where all choices of P_1, P_2, P'_1, P'_2 with the above properties we have $r'_1 = r_2$ or P'_1 and P_2 intersect in a vertex

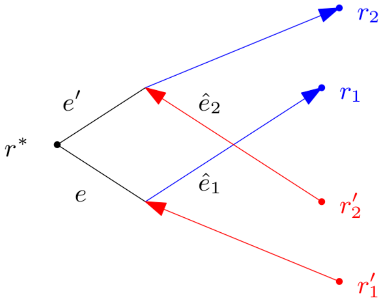
APPENDIX



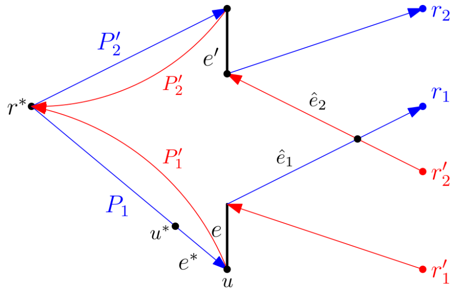
(a) Case B-1-i: $|\hat{P}_1| \leq |\hat{P}_2'|$ and $|\hat{P}_1| + |\hat{P}_2'| \geq 3$.



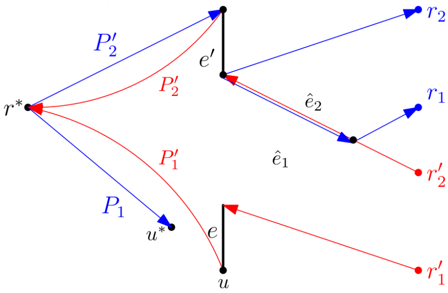
(b) Modification of Case B-1-i.



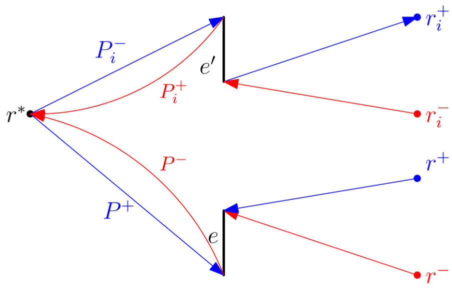
(c) Case B-1-ii(a): The edges e, e', $\hat{e}_1, \hat{e}_2$ form a C_4.



(d) Case B-1-ii(b): e, e', $\hat{e}_1, \hat{e}_2$ do not form a C_4.



(e) Modification of Case B-1-ii-b.



(f) Case B-2: $r^+ \neq r_i^+$ and $r^- \neq r_i^-$.

Figure 10 Cases B-1 – B-2, where *blue* color corresponds to the labeling $\lambda_{r^*}^+$ and *red* to $\lambda_{r^*}^-$.

after they traversed e' and e, respectively, is symmetric.

Fix temporal paths P_1, P_2, P_1', P_2' with the above properties and $r_1 = r_2'$ or P_1 and P_2' intersect in a vertex after they traversed e and e', respectively. Let $\hat{P}_1$ be the path segment from e to the first vertex included in P_2' (excluding e) and let $\hat{P}_2'$ be the path segment from the last vertex included in P_1 to e' (excluding e').

Case B-1-i. Assume $|\hat{P}_1| \leq |\hat{P}_2'|$ (the opposite case is symmetric) and $|\hat{P}_1| + |\hat{P}_2'| \geq 3$ (not both paths are only a single edge). We remove $\hat{P}_2'$'s edges and e and the corresponding labels from $H_{r^*}^-$ and $\lambda_{r^*}^-$, respectively, such that there is a temporal path from r_2' to e that uses the new labels.

Note that after the modifications $H_{r^*}^+$ and $H_{r^*}^-$ are still trees, and $\lambda_{r^*}^+$ and $\lambda_{r^*}^-$ still assign at most one label per edge. Furthermore, we have that the modification do not increase the sum of edges in both trees $|E(H_{r^*}^+) \cup E(H_{r^*}^-)|$. Lastly, and most importantly, we have that at least one of $H_{r^*}^+$ and $H_{r^*}^-$ does contain both edges e and e'.

Case B-1-ii. Assume $|\hat{P}_1| = |\hat{P}_2'| = 1$, that is, both paths are only a single edge $\hat{e}_1$ and $\hat{e}_2$, respectively.

Case B-1-ii(a). The edges e, e', $\hat{e}_1$, and $\hat{e}_2$ form a C_4. Then we are in the case that k is even. In this case we set $\hat{e}_1$ to be e^+ and we set $\hat{e}_2$ to be e^-. One of these two edges will be used to close the C_4, depending on whether which of $H^+_{r^*}$ and $H^-_{r^*}$ has fewer edges. The edges e and e' stay in $E^*_{r^*}$ and will be the only two edges for which we cannot account a label in λ that is not present in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$. In this case we have that $|E^*_{r^*}| \leq x' + 2$ is fulfilled.

If Case B-1-ii(a) never applies, then we are in the case that k is odd and we have to be able to account a label in λ that is not present in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$ for all but one edge in $E^*_{r^*}$.

Case B-1-ii(b). The edges e, e', $\hat{e}_1$, and $\hat{e}_2$ do not form a C_4. Let $e = uv$ and $e' = u'v'$ and let u and u' be the vertices closer to r^* in P_1 and P'_2, respectively. Then this means there is either at least one edge e^* between r^* and u or between r^* and u'. Consider the case where e^* is between r^* and u and let $e^* = uu^*$ for some vertex u^*. In this case e^* is contained in $H^+_{r^*}$. The other case is symmetric. Let e^{**} be the edge between r^* and u in $H^-_{r^*}$, that is incident with u.

Note that $\lambda^+_{r^*}(e^*) < \lambda^+_{r^*}(e) < \lambda^-_{r^*}(e^{**})$. We now make the following modification. We remove label $\lambda^+_{r^*}(e^*)$ and add a new label to $\hat{e}_2$ in $\lambda^+_{r^*}$ that is chosen in a way that allows for a temporal path from r^* to r_1 via e' and then $\hat{e}_2$.

Case B-2. Fix some $e \in E^*_{r^*}$ and let P^+ be a path in $H^+_{r^*}$ from r^* to some $r^+ \in R\setminus\{r^*\}$ that only traverses e from the edges in $E^*_{r^*}$ and let P^- be a path in $H^-_{r^*}$ from some $r^- \in R \setminus \{r^*\}$ to r^* that only traverses e from the edges in $E^*_{r^*}$. For all $e_i \in E^*_{r^*} \setminus \{e\}$ let P^+_i be a path in $H^+_{r^*}$ from r^* to some $r^+_i \in R \setminus \{r^*\}$ that only traverses e_i from the edges in $E^*_{r^*}$ and let P^-_i be a path in $H^-_{r^*}$ from some $r^-_i \in R \setminus \{r^*\}$ to r^* that only traverses e_i from the edges in $E^*_{r^*}$. Note that for all $i \neq i'$ we have that $r^+_i \neq r^+_{i'}$ and $r^-_i \neq r^-_{i'}$. Now consider edge e_i. If $\lambda^+_{r^*}(e) \leq \lambda^+_{r^*}(e_i)$, then the temporal path in (G, λ) from r^-_i to r^+ needs at least one label that is not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$. If $\lambda^+_{r^*}(e) > \lambda^+_{r^*}(e_i)$, then the temporal path in (G, λ) from r^- to r^+_i needs at least one label that is not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$. This implies, if Case B-1-ii(a) does not apply, that λ contains at least $|E^*_{r^*}| - 1$ labels that are not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$ and hence $|E^*_{r^*}| \leq x' + 1$. If Case B-1-ii(a) applies, then λ contains at least $|E^*_{r^*}| - 2$ labels that are not contained in $\lambda^+_{r^*}$ or $\lambda^-_{r^*}$ and hence $|E^*_{r^*}| \leq x' + 2$.

This finishes the proof. ◀

Having Lemma 21, 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. Recall the definition of Steiner Tree.

Steiner Tree
Input: A static graph $G = (V, E)$, a subset of vertices $R \subseteq V$ and a positive integer k.
Question: Is there a subtree of G that includes all the vertices of R and that contains at most k edges.

Let (G, R, k) be an instance of MSL. Note that if G is C_4-free, then Lemma 21 immediately implies that we can use an algorithm for Steiner Tree on the same input graph G with the same terminal vertices R and check whether the resulting solution subtree has at most $k^* = \lceil (k+1)/2 \rceil$ edges. In the case where G contains C_4s, we have to determine first whether there is a C_4 in G that can be labeled in an optimal labeling. Formally, we show the following.

▶ **Theorem 22.** MSL *is in FPT when parameterized by the number of terminals.*

Proof. Assume we have access to an algorithm $\mathcal{A}$ for Steiner Tree that on input (G, R) outputs the size of a minimum solution, that is, an integer k such that (G, R, k) is a YES instance of Steiner Tree and $(G, R, k - 1)$ is a NO instance of Steiner Tree.

APPENDIX

Let (G, R, k) be an instance of MSL and let $k^* = \mathcal{A}(G, R)$. For all C_4's in G let $k_{C_4} = \mathcal{A}(G, R \cup V(C_4))$. If there exist an C_4 in G such that $k_{C_4} = k^*$, then (G, R, k) is a YES instance of MSL if and only if $k \geq 2k^* - 2$. Otherwise (G, R, k) is a YES instance of MSL if and only if $k \geq 2k^* - 1$.

We first show correctness, then we analyse the running time.

($\Leftarrow$): Assume there exist a C_4 in G such that $k_{C_4} = k^*$. Then there exist a subtree of G connecting all terminal vertices and containing three edges of the C_4. We add the missing edge of the C_4 and label the subgraph using Theorem 7. This requires $2k^* - 2$ labels and clearly afterwards all terminals can pairwise reach each other. Hence, we have that if $k \geq 2k^* - 2$, then (G, R, k) is a YES instance of MSL. Assume there is no C_4 in G such that $k_{C_4} = k^*$. Then there exist a subtree of G connecting all terminal vertices and containing k^* edges. We label this tree using Theorem 7. This requires $2k^* - 1$ labels and clearly afterwards all terminals can pairwise reach each other. Hence, we have that if $k \geq 2k^* - 1$, then (G, R, k) is a YES instance of MSL.

($\Rightarrow$): Assume that (G, R, k) is a YES instance of MSL and let $k_{\text{opt}} \leq k$ such that (G, R, k_{opt}) is a YES instance of MSL and $(G, R, k_{\text{opt}} - 1)$ is a NO instance of MSL. By Lemma 21, we have that if k_{opt} is odd, then there is a labeling λ of size k_{opt} for G such that the edges labeled by λ form a tree H, and every leaf of H is a vertex in R. It is easy to see that H is a solution for the Steiner Tree instance (G, R). Hence, $\mathcal{A}(G, R)$ outputs a lower bound k^* for the number of edges in H. Furthermore, since all leafs of H are terminals, we have that every vertex in (H, λ) can temporally reach every other vertex. By Theorem 7 we know that then λ needs $2k^* - 1$ labels. This implies that $k \geq k_{\text{opt}} \geq 2k^* - 1$.

Now assume that k_{opt} is even. Then by Lemma 21 we have that there is a labeling λ of size k^* for G such that the edges labeled by λ form a graph H that is a tree H' with one additional edge that forms a C_4, and every leaf of H' is a vertex in R. For the C_4 that is formed we have that $\mathcal{A}(G, R \cup V(C_4))$ outputs a lower bound k^* for the number of edges in H'. Note that we have $k^* \leq \mathcal{A}(G, R)$, since otherwise $2k^* - 2 > 2\mathcal{A}(G, R) - 1$, which means by Theorem 7 that $k_{\text{opt}} < 2k^* - 2$. However, since all leafs of H' are terminals, we have that every vertex in (H, λ) can temporally reach every other vertex. Hence, Theorem 7 implies that $k_{\text{opt}} \geq 2k^* - 2$. It follows that $k_{\text{opt}} < 2k^* - 2$ leads to a contradiction and we have $k \geq k_{\text{opt}} \geq 2k^* - 2$.

Running time: We can use the FPT-algorithm for Steiner Tree parameterized by the number of terminals by Dreyfus and Wagner [14] for algorithm $\mathcal{A}$. Note that we need to iterate over all C_4s in G (there are at most n^4 of them). Each time we invoke $\mathcal{A}(G, R \cup V(C_4))$, we increase the number of terminals by at most four. It follows that overall we obtain an FPT running time for the number of terminals as a parameter. ◀

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 19 and 20. 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 22, implies that MASL is strictly harder than MSL (parameterized by the number $|R|$ of terminals), unless FPT=W[1].

▶ **Theorem 23.** 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.*

Proof. To prove that the MASL is W[1]-hard when parameterized by the combination of the

APPENDIX

number $|R|$ of the terminals and the number k of labels, even if the restriction a on the age is a constant, we provide a parameterized polynomial-time reduction from Multicolored Clique parameterized by the number of colors, which is W[1]-hard [19].

Multicolored Clique

Input: A static graph $G = (V, E)$, a positive integer k, a vertex-coloring $c : V(G) \to \{1, 2, \ldots, k\}$.

Question: Does G have a clique of size k including vertices of all k colors?

Let (G, k, c) be an input of the Multicolored Clique problem and denote $|V(G)| = n, |E(G)| = m$. We construct (G^*, R^*, a^*, k^*), the input of MASL using the following procedure. The vertex set $V(G^*)$ consists of the following vertices:

- a "color-vertex" corresponding to every color of $V(G)$: $C = \{c_i | i \in \{1, 2, \ldots, k\}$ a color of $V(G)\}$,
- 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: $U_E = \{u_e | e \in E(G)\}$,
- a "color-combination-vertex" corresponding to a pair of two colors of $V(G)$: $W = \{c_{i,j} | i, j \in \{1, 2, \ldots, k\}, i < j,$ colors of $V(G)\}$, and
- $2n + 4m + 5m + \frac{11}{8}(k^4 - 2k^3 - k^2 + 2k) + \frac{11}{2}(k^3 - 3k^2 + 2k)$ "dummy" vertices.

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

- a path of length 3 (using 2 dummy vertices) between a color-vertex c_i, corresponding to the color i, and every vertex-vertex $u_v \in U_V$, where v is of color i in $V(G)$, i.e., $c(v) = i$,
- for every edge $e = vw \in E(G)$, where $c(v) = i$ and $c(w) = j$, we connect the corresponding edge-vertex u_e with
  - the vertex-vertices u_v and u_w, each with a path of length 3 (using 2 dummy vertices),
  - the color-combination-vertex $c_{i,j}$, with a path of length 6 (using 5 dummy vertices),
- a path of length 12 (using 11 dummy vertices), between each pair of color-combination-vertices, and
- a path of length 12 (using 11 dummy vertices), between all pairs of color-vertices c_i and color-combination-vertices c_{jk}, where $i \notin \{j, k\}$, i.e., we connect the color-vertex of color i with all color-combination vertices of pairs of color that do not include i.

We set $R^* = C \cup W$ (note that $|R^*| \in O(k^2)$), $a^* = 12$ and $k^* = 6k + 6(k^2 - k) + 6(k^2 - k) + 3(k^4 - 2k^3 - k^2 + 2k) + 12(k^3 - 3k^2 + 2k)$. This finishes the construction. It is not hard to see that this construction can be performed in polynomial time. For an illustration see Figure 11. At the end G^* is a graph with $3n + 10m + \frac{1}{2}(k^2 + k) + \frac{11}{8}(k^4 - 2k^3 - k^2 + 2k) + \frac{11}{2}(k^3 - 3k^2 + 2k)$ vertices and $3n + 12m + \frac{3}{2}(k^4 - 2k^3 - k^2 + 2k) + 6(k^3 - 3k^2 + 2k)$ edges.

We claim that (G, k, c) is a YES instance of the Multicolored Clique if and only if (G^*, R^*, a^*, k^*) is a YES instance of the MASL.

($\Rightarrow$): Assume (G, k, c) is a YES instance of the Multicolored Clique. Let $S \subseteq V(G)$ be the set of vertices that form a multicolored clique in G. We construct a labeling λ for G^* that uses k^* labels, which are not larger than $a^* = 12$, and admits a temporal path between all vertices from R^* as follows.

Let U_S be the set of corresponding vertices to S in G^*. For each $v \in S$ of color i we label the three edges connecting c_i to u_v with labels $1, 2, 3$, one per each edge, in order to create temporal paths starting in c_i and with labels $12, 11, 10$, one per each edge, in order to create temporal paths that finish in c_i. For every edge $vw = e \in E$ with endpoints in S we label the path from both of its endpoint vertex-vertices u_v, u_w to the edge-vertex u_e with labels $4, 5, 6$, one per each edge, and with labels $9, 8, 7$, one per each edge. This ensures the existence of both temporal paths between c_i and c_j. More precisely, (c_i, c_j)-temporal

APPENDIX

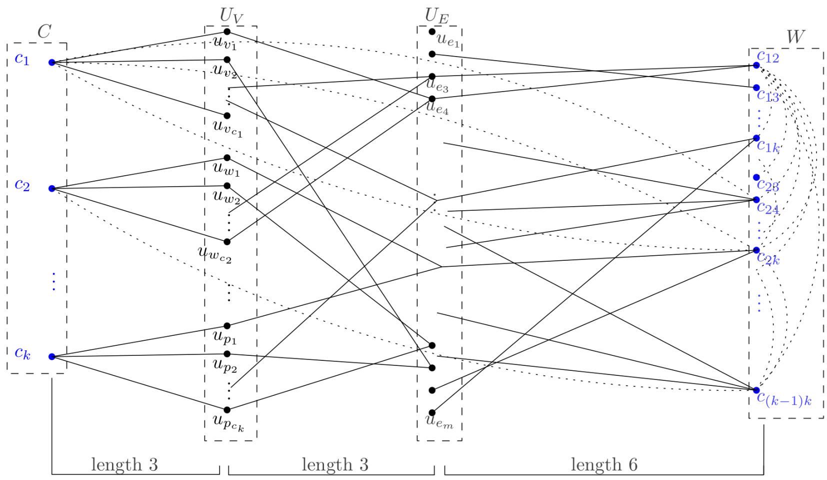


Figure 11 An example of the construction of the input graph for MASL. For better readability, some paths among the vertices in W and paths among $c_i \in C$ and $c_{jk} \in W$ ($i \neq j \neq k$), are not depicted.

path (resp. (c_j, c_i)-temporal path) uses labels $1, 2, 3$ to reach u_v (resp. u_w), from where it continues with $4, 5, 6$ to u_e, then with $7, 8, 9$ reaches u_w (resp. u_v) and finally with $10, 11, 12$ it finishes in c_j (resp. c_i). Note, since S is a multicolored clique then each vertex $v' \in S$ is of a unique color i' and all vertices in S are connected. Therefore, using the above construction for all vertices in S, vertex c_i reaches and is reached by every other color vertex c_j through the vertex-vertex u_v. Even more, since there is an edge e connecting any two vertices $v, w \in S$, there is a unique edge vertex u_e (and consequently a unique path), that is used for both temporal paths between vertex-vertices u_v, u_w and their corresponding color-vertices. The above construction clearly produces a temporal path (of length 12) between any two color-vertices. This construction uses $2 \cdot 3$ labels between every color-vertex c_i and its unique vertex-vertex u_v, where $v \in S$ and $c(c) = i$, and $2 \cdot 6$ labels from each edge vertex u_e to both of its endpoint vertex-vertices, where e is an edge of the multicolored clique formed by the vertices in S. All in total we used $6k + 12\binom{k}{2} = 6k + 6(k^2 - k)$ labels, to connect all edge-vertices corresponding to edges formed by S with their endpoints vertex-vertices.

Now, let c_{ij} and $c_{i'j'}$ be two arbitrary color-combination-vertices. By the construction of G^* there is a unique path of length 12 connecting them, which we label with labels $1, 2, \ldots, 12$ in both directions. This labeling uses $2 \cdot 12$ labels for each pair of color-combination-vertices, hence all together we use $24\frac{|W|(|W|-1)}{2}$ labels, since $|W| = \binom{k}{2}$ this equals to $3(k^4 - 2k^3 - k^2 + 2k)$.

Finally, let $c_{i'}$ and c_{ij} be two arbitrary color and color-combination-vertices, respectively. In the case when $i' \notin \{i, j\}$ there is a unique path of length 12 in G^* between them (that uses only the dummy vertices). We label this path with labels $1, 2, \ldots, 12$ in both directions. This procedure uses $2 \cdot 12$ labels for each pair of such vertices, hence all together we use $24k\binom{k-1}{2}$ labels, which equals to $12(k^3 - 3k^2 + 2k)$. In the case when $i' \in \{i, j\}$ (w.l.o.g. $i' = i$) we connect the vertices using the following path. In S exists a unique vertex of color i, denote it v. By the definition of S there is also vertex w of color j, which is connected to v with some edge, denote it e. Therefore, to obtain a (c_i, c_{ij})-temporal path, we first reach u_v from c_i with labels $1, 2, 3$, then continue to u_e, using labels $4, 5, 6$, from where we continue to $c_{i,j}$ using the labels $7, 8, \ldots, 12$. The (c_{ij}, c_i)-temporal path uses the same edges,

APPENDIX

with labels in reversed order. This construction introduced $2 \cdot 6$ new labels on the path of length 6 between the edge-vertex u_e and the color-class-vertex c_{ij} and reused all labels on the (c_i, u_e)-temporal paths. Repeating this for every color-class-vertex we use $2 \cdot 6|W|$ new labels, since $|W| = \binom{k}{2}$ this equals to $6(k^2 - k)$.

All together λ uses $6k + 6(k^2 - k) + 6(k^2 - k) + 3(k^4 - 2k^3 - k^2 + 2k) + 12(k^3 - 3k^2 + 2k)$ labels.

($\Leftarrow$): Assume that (G^*, R^*, a^*, k^*) is a YES instance of the MASL and let λ be the corresponding labeling of G^*. Before we construct a multicolored clique for G, we prove that the distance between any two terminal vertices from R^* in G^* is 12.

Case A. Let $c_i, c_j \in C$ be two arbitrary color-vertices and let e be an edge in G with endpoints of color i and j, i.e., $e = vw \in E(G)$ and $c(v) = i, c(w) = j$. There are two options how to reach c_j from c_i. One when the path connecting them passes through the set E and the other, when it passes through the set W.
Case A-1. If the path passes through the set E, we must first go through a vertex-vertex u_v, then we go to the edge-vertex u_e, continue to the vertex-vertex u_w and finish in c_j. Since all these vertices are connected with a path of length 3, we get that the distance of the whole (c_i, c_j)-path is 12.
Case A-2. If the path passes through the set W, then we must go through the color-class-vertex c_{ij}. Since the path between any color-vertex and color-class-vertex is of length 12 (we prove this in the following paragraph), the whole (c_i, c_j)-path is of length 24.

Therefore, the shortest path connecting two color-vertices is of length 12 and must go through the appropriate edge-vertex.

Case B. Let c_{ij} and $c_{i'}$ be two arbitrary vertices from the color-class-vertices and color-vertices. We distinguish two cases.
Case B-1. First, when $i' \notin \{i, j\}$. Then, by the construction of G^*, there exists a direct path of length 12, connecting them. Any other $(c_{i'}, c_{ij})$-path must either go from $c_{i'}$ to some color-class-vertex $c_{i'j'}$, which is then connected with a path of length 12 to the c_{ij}, or go to one of the color-vertices and then continue to the c_{ij}. In both cases the constructed path is strictly longer than 12.
Case B-2. Second, when $i' \in \{i, j\}$. Let $c(v) = i$ and $vw = e \in E(G)$. Then there is a path from c_i to c_{ij} that goes through the vertex-vertex u_v (using a path of length 3), continues to the edge-vertex u_e (using a path of length 3), which is connected to the color-class-vertex c_{ij} (using a path of length 6). Hence the constructed (c_i, c_{ij})-path is of length 12. There exists also another (c_i, c_{ij})-path, that goes through some other $c_{ij'}$ color-class-vertex, but it is longer than 12.

Case C. Let c_{ij} and $c_{i'j'}$ be two arbitrary color-class-vertices. By construction of G^*, there is a path of length 12 connecting them. Any other $(c_{ij}, c_{i'j'})$-path, must use at least one vertex-vertex, which is on the distance 9 from the color-class-vertices (therefore the path through it would be of length at least 18), or a color-vertex, which is on the distance 12 from the color-class-vertices. In both cases the constructed path is strictly longer than 12.

It follows that the distance between any two terminal vertices in R^* is 12, hence a temporal path connecting them must use all labels from 1 to 12. Using this property we know that any labeling that admits a temporal path among all terminal vertices must definitely use all labels $1, 2, \ldots, 12$ on the temporal paths among any two color-combination-vertices c_{ij} and $c_{i'j'}$, and among a color-vertex $c_{i'}$ and a color-combination-vertex c_{ij}, where $i' \notin \{i, j\}$. This is true as there are unique paths of length 12 among them. For these temporal paths

APPENDIX

we must use $2 \cdot 12\frac{|W|(|W|-1)}{2}$ labels (since $|W| = \binom{k}{2}$ this equals to $3(k^4 - 2k^3 - k^2 + 2k)$) and $2 \cdot 12k\binom{k-1}{2}$ labels (which equals to $12(k^3 - 3k^2 + 2k)$). Therefore, the labeling λ can use only $6k + 6(k^2 - k) + 6(k^2 - k)$ labels to connect all other terminals.

Let us now observe what happens with the temporal paths connecting remaining temporal vertices. To create a temporal path starting in a color-vertex c_i and ending in some other color-vertex (or color-combination-vertex), λ must label at least 3 edges, to allow c_i to reach one of its corresponding vertex-vertices u_v. Similarly it holds for a temporal path ending in c_i. Since the path connecting c_i to some other terminal is of length 12, the labels used on the temporal paths starting and ending in c_i cannot be the same. In fact the labels must be $1, 2, 3$ for one direction and $12, 11, 10$ for the other. Therefore, λ uses at least $6k$ labels on edges between vertices of C and U_V. Extending the arguing from above, for c_i to reach some (suitable) edge vertex u_e the path needs to continue from u_v to u_e and must use the labels $4, 5, 6$ (or $9, 8, 7$ in case of the path in the opposite direction). From u_e the path can continue to the corresponding color-combination-vertex $c_{i,j}$ where it must use the labels $7, 8, \dots, 12$, or to the vertex-vertex corresponding to the other endpoint of e. This finishes the construction of the temporal path from a color-vertex to the color-class-vertex and the temporal paths among color-vertices. The remaining thing is to connect a color-class-vertex with its corresponding color-vertices. The temporal path must go through some edge vertex u_e, that is on the distance 6 from it, therefore the labeling must use the labels $1, 2, \dots, 6$. From u_e the path continues to the suitable vertex-vertex and then to the color-vertex. Using the above labeling we see that λ must use at least $2 \cdot 6|W|$ labels (which equals to $6(k^2 - k)$ labels) on the edges between the color-class-vertices in W and the edge vertices in U_E and at least $2 \cdot 6\binom{k}{2}$ labels (which equals to $6(k^2 - k)$ labels) on the edges between the edge-vertices in U_E and vertex-vertices in U_V. Since all this together equals to k^*, all of the bounds are tight, i.e., labeling cannot use more labels.

We still need to show that for every color-vertex c_i there exists a unique vertex-vertex u_v connected to it such that all temporal paths to and from c_i travel only through u_v. By the arguing on the number of labels used, we know that there can be at most two vertex-vertices that lie on temporal paths to or from c_i. More precisely, one that lies on every temporal path starting in c_i and the other that lies on every temporal path that finishes in c_i. Let now $u_v, u_{v'}$ be two such vertex-vertices. Suppose that u_v lies on all temporal paths that start in c_i and $u_{v'}$ on all temporal paths that end in c_i. Now let u_e be the edge-vertex on a temporal path from c_i to c_j, and let u_w be the vertex-vertex connected to c_j and u_e. Therefore the (c_i, c_j)-temporal path has the following form: it starts in c_i, uses the labels $1, 2, 3$ to reach u_v, then continues to u_e with $4, 5, 6$, then with $7, 8, 9$ reaches u_w and with $10, 11, 12$ ends in the c_j. To obtain the (c_i, c_{ij})-teporal path we must label the edges from u_e to c_{ij} with the labels $6, 7, \dots, 12$, since the edge-vertex u_e is the only edge-vertex connected to the color-class-vertex c_{ij} that can be reached from c_i (if there would be another such edge-vertex, then the labeling λ would use too many labels on the edges between U_V and U_E). Now, for the color-vertex c_j to be able to reach the color-class-vertex c_{ij}, it must use the same labels between the u_e and c_{ij} (using the same reasoning as before). Therefore the path from c_j to u_e (through) u_w uses also the labels $1, 2, \dots, 6$. But then for c_j to reach c_i the temporal path must use the vertex-vertex u_w, even more it must use the edge vertex u_e and consequently the vertex-vertex u_v, from where it would reach c_i. But this is in the contradiction with the assumption that the path from c_i to u_v uses only labels $1, 2, 3$. Therefore, every color-vertex c_i admits a unique vertex-vertex u_v that lies on all (c_i, c_j) and (c_j, c_i)-temporal paths. For the conclusion of the proof we claim that all vertices v corresponding to these unique vertex-vertices u_v of color-vertices c_i, form a multicolored

APPENDIX

clique in G. This is true as, by construction, a temporal path between two vertex-vertices u_v, u_w corresponds to the edge $vw = e \in E(G)$. Since every vertex-vertex is connected to exactly one color-vertex, this corresponds to the vertex coloring of $V(G)$. In G^* there is a temporal path among any two color vertices, therefore the vertex-vertices used in these temporal paths can be reached among each other, which means that they really do form a multicolored clique. ◀

Note here that, in the constructed instance of MASL in the proof of Theorem 23, 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 23 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 24.** 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 19, 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.

APPENDIX

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

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

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

22 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.

23 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.

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

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

26 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.

APPENDIX

27 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.

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. URL: `https://arxiv.org/abs/2010.08423`.

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