The complexity of computing optimum labelings for temporal connectivity ⋆

Nina Klobas[a], George B. Mertzios[a], Hendrik Molter[b], Paul G. Spirakis[c]

[a]*Department of Computer Science, Durham University, United Kingdom*
[b]*Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[c]*Department of Computer Science, University of Liverpool, United Kingdom*

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 total number $|\lambda|$ of time-labels that we need to assign to the edges of G such that the resulting temporal graph (G, λ) is temporally connected? Here $|\lambda|$ denotes the sum, over all edges, of the size of the set of labels associated to an edge. 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*. 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 of time-labels of the solution. 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).

⋆An extended abstract of this work was published in the proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022) [29]. This version contains full proof details. George B. Mertzios was supported by the EPSRC grant EP/P020372/1. Hendrik Molter was supported by the ISF, grants No. 1070/20 and No. 1456/18, and the ERC, grant number 949707. Paul G. Spirakis was supported by the EPSRC grant EP/P02002X/1.

Email addresses: `nina.klobas@durham.ac.uk` (Nina Klobas), `george.mertzios@durham.ac.uk` (George B. Mertzios), `molterh@post.bgu.ac.il` (Hendrik Molter), `p.spirakis@liverpool.ac.uk` (Paul G. Spirakis)

Keywords:
Temporal graph, Graph Labeling, Foremost Temporal Path, Temporal Connectivity, Steiner Tree.

---

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 [25, 38, 40]. Inspired by the foundational work of Kempe et al. [27], 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 [27])**.** *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 finite set of discrete time-labels.*

In our work we require the underlying graph G to be a simple graph, although in general this does not have to be the case. 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. [27] 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. [33] to temporal graphs with multiple labels per edge. Furthermore, depending on the particular application, both variations of temporal paths with non-decreasing [6, 27, 28] and with strictly increasing [15, 31, 33] 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, 2, 3, 8, 10, 17, 18, 19, 22, 30, 37, 41].

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 by Deligkas et al. [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, 39]. Furthermore, some non-path temporal graph problems have been recently introduced too, including for example temporal variations of maximal cliques [7, 42], vertex cover [4, 23], vertex coloring [32, 36], matching [34], and transitive orientation [35].

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 directed version has already been investigated, as detailed shortly. 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 assign 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 by Akrida et al. [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 the 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. [33] 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. In this paper we overcome these difficulties by developing a hardness reduction from a variation of the problem Max XOR SAT (see Theorem 13 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 size of the labeling $|\lambda|$ and number $|R|$ of terminal vertices, by providing a parameterized Turing reduction to Steiner Tree parameterized by the number of terminals. Steiner Tree is known to be fixed-parameter tractable with respect to the number of terminal vertices [14]. 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 in Lemma 15 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 forms a C_4.

In Section 4.3 we prove that MASL is W[1]-hard even with respect to the number of time-labels of the solution. This also implies that MASL is W[1]-hard with respect to the number $|R|$ of terminals, since the number of time-labels in the solution 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.

For an easier overview of the area, we also outline all of the known and new results in Table 1.

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). A *tree* is a connected graph that does not contain any cycles. A *subtree* T of a graph G is a subgraph of G that is also a tree. 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

Graph restrictions	Age non-restricted	Age restricted
Directed graphs	open	APX-hard [33]
Directed acyclic graphs	poly. time solvable (see Theorem 7)	poly. time solvable (see Theorem 7)
Undirected cycles	poly. time solvable (see Theorem 5)	poly. time solvable (see Lemma 2)
Undirected graphs	poly. time solvable (see Theorem 5)	NP-hard (see Theorem 13)
Steiner labeling, undirected graphs	NP-complete, FPT w.r.t. $\lvert R\rvert$ (see Theorems 14 and 16)	W[1]-hard w.r.t. $\lvert\lambda\rvert, \lvert R\rvert$ (see Theorem 17)

Table 1: An overview of previously known results, our results, and open problems. Rows 2–5 concern ML (second column) and MAL (third column), while row 6 concerns MSL (second column) and MASL (third column).

path 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 the formal definitions of our four decision problems.

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

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

Note that if the temporal paths are allowed to admit non-decreasing time-labels all problems are solved by simply assigning label 1 to all (necessary) edges. Therefore, ML and MAL can be solved in polynomial time, while MSL and MASL reduce directly to Steiner Tree. Observe also that for MAL, whenever G is not connected or the input age bound a is strictly smaller than the diameter d of G, the answer is NO. Thus, we always assume in the analysis of MAL that G is a connected graph and $a \geq d$, where d is the diameter of the input graph G. For simplicity of the presentation, we denote by $\kappa(G, d)$ the smallest number

k for which (G, d, k) is a YES instance for MAL.

Observation 1. *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 1 is asymptotically tight as, for cycle graphs C_n with diameter d, we have that $\kappa(C_n, d) = \Theta(n^2)$.

Lemma 2. *Let C_n be a cycle on n vertices, where $n \neq 4$, and let d be its diameter. Then*

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

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.

Case 1: n is odd. Let $n = 2d+1$. Then, for each vertex $v_i \in V(C_n)$, there are exactly two distinct vertices v_{i+d} and v_{i-d} at distance d from v_i. In particular, there exists a unique path of length d from v_i to v_{i+d}, and thus the only way that a temporal path (with labels at most d) can exist from v_i to v_{i+d} is that the jth edge (for every $j = 1, \ldots, d$) of the unique path of length d from v_i to v_{i+d} contains the label j. Due to symmetry, by just considering every vertex v_i of C_n, it follows that every edge of C_n must contain each of the labels $1, 2, \ldots, d$. Therefore $\kappa(C_n, d) \geq nd = 2d^2 + d$.

Conversely, in the labeling of C_n, where every edge contains every label in $\{1, 2, \ldots, d\}$, clearly the age of the temporal graph is d and there exists a temporal path from every vertex to every other vertex. Therefore $\kappa(C_n, d) \leq nd = 2d^2 + d$, and thus $\kappa(C_n, d) = 2d^2 + d$ when $n = 2d + 1$.

Case 2: n is even. Let $n = 2d$. Then, for each vertex $v_i \in V(C_n)$, there is exactly one vertex v_{i+d} at distance d from v_i, and exactly two distinct vertices v_{i+d-1} and v_{i-d+1} at distance $d-1$ from v_i. In particular, there exists a unique path of length $d-1$ from v_i to v_{i+d-1}, and thus the only way that a temporal path (with labels at most d) can exist from v_i to v_{i+d-1} is that the jth edge (for every $j = 1, \ldots, d-1$) of the unique path of length $d-1$ from v_i to v_{i+d-1} contains the label j or the label $j+1$.

We will now prove that, without loss of generality, for every two consecutive edges $v_{i-1}v_i$ and v_iv_{i+1}, the total number of labels of these two edges is at least d, i.e., $|\lambda(v_{i-1}v_i)| + |\lambda(v_iv_{i+1})| \geq d$. Suppose otherwise that $|\lambda(v_{i-1}v_i)| + |\lambda(v_iv_{i+1})| \leq d-1$. Then there exists some $a \in \{1, 2, \ldots, d\}$ such that neither of the edges $v_{i-1}v_i$ and v_iv_{i+1} contains label a. First, let $a = 1$. Then any temporal path from v_i to v_{i+d} will have to start with label at least 2, and thus cannot arrive at v_{i+d} by time d, a contradiction. Second, let $a = d$. Similarly, any temporal path from v_{i+d} to v_i will have to arrive at v_i by time $d-1$. However, this is not possible, as the distance between v_{i+d} and v_i in C_n is d, a contradiction.

Now let $2 \leq a \leq d-1$. Then, the only way that a temporal path (with labels at most d) can exist from vertex v_{i-a} to vertex $v_{i-a+d-1}$ is that the edges $v_{i-1}v_i$ and v_iv_{i+1} contain the label $a+1$ and the label $a+2$, respectively, as both these edges cannot contain label a by assumption. Similarly, the only way that a temporal path (with labels at most d) can exist from vertex v_{i-a+1} to vertex v_{i-a+d} is that the edge edges $v_{i-1}v_i$ and v_iv_{i+1} contain the label $a-1$ and the label $a+1$, respectively. By symmetry it follows that edge v_iv_{i+1} also contains label $a-1$ (by just considering vertices v_{i+a} to vertex $v_{i+a-d+1}$). That is, for the two consecutive edges $v_{i-1}v_i$ and v_iv_{i+1} we have that

$$a-1, a+1 \in \lambda(v_{i-1}v_i) \cap \lambda(v_iv_{i+1}) \tag{1}$$

Summarizing, consider two consecutive edges $v_{i-1}v_i$ and v_iv_{i+1}. The union $\lambda(v_{i-1}v_i) \cup \lambda(v_iv_{i+1})$ of their labels always contains labels 1 and d. The only possibility that this union is missing some label a is that both $\lambda(v_{i-1}v_i)$ and $\lambda(v_iv_{i+1})$ contain both labels $a-1$ and $a+1$. Furthermore, it follows that it is not possible that two consecutive labels $a, a+1$ miss from the union $\lambda(v_{i-1}v_i) \cup \lambda(v_iv_{i+1})$. Therefore $|\lambda(v_{i-1}v_i)| + |\lambda(v_iv_{i+1})| \geq d$. Thus, as there are $\frac{n}{2}$ disjoint pairs of consecutive edges, it follows that $\kappa(C_n, d) \geq \frac{n}{2}d = d^2$.

Conversely, consider the labeling where, for every $i = 1, \ldots, d$, the edge $v_{2i-1}v_{2i}$ (resp. the edge $v_{2i}v_{2i+1}$) contains all the odd (resp. all the even) labels within the set $\{1, 2, \ldots, d\}$. It is straightforward to check that, with this labeling, there exists a temporal path (with maximum label at most d) from every vertex to every other vertex. Therefore $\kappa(C_n, d) \leq \frac{n}{2}d = d^2$, and thus $\kappa(C_n, d) = d^2$. □

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 [24]).

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. [21], where the relationship with the gossip problem has also been drawn. Contrary to ML, this problem is NP-hard [21]. 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 rooted 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.

The gossip problem considers a set A of n agents, each possessing a unique secret. Two agents $x, y \in A$ can communicate by making a phone call, denoted as an unordered pair (x, y). During their conversation, they share all the information they currently know. The objective is to determine a minimum sequence of phone calls that results in all agents knowing all secrets. We focus on a specific variation of the gossip problem where each agent can call only a specific subset of agents from A. This problem can be modelled using a graph $G = (V, E)$, where each agent $x \in A$ is represented by a vertex $v_x \in V$ and for every allowed phone call between agents x and y we add an edge $v_x v_y$ to the set of edges E of G. The goal is to find a minimum sequence of edges in G such that, by following this sequence, all agents end up knowing all the secrets.

The above gossip problem is naturally connected to ML. The only difference between the two problems is that, in gossip protocols, 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 4).

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 $\mathcal{G}_{\mathcal{C}} = (G, \lambda)$ using the following procedure. For every agent $x \in A$ we create a vertex $v_x \in V(G)$ and for every allowed phone call between agents x and y we add an edge $v_x v_y$ to $E(G)$. We now label edges of G using the following procedure: for every phone call $c(i) \in \mathcal{C}$ between its two corresponding agents x_i, y_i we add the label i to the edge $(v_{x_i} v_{y_i})$ of $\mathcal{G}_{\mathcal{C}}$. In the end, the labeling λ is completely determined by the sequence of phone calls.

Observation 3. *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 3.

Lemma 4. *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 $t \in \{1, 2, \dots, \alpha(G, \lambda)\}$ be an arbitrary time step within the lifetime of (G, λ), and let E_t be the set of edges of G that appear at time t in (G, λ). Let us denote with $k_t = |E_t|$. We now order edges from E_t in an arbitrary order, so we get $e_1^t, e_2^t, \dots, e_{k_t}^t$, where $e_i^t \in E_t$. We repeat this for all t in the lifetime of (G, λ) and get an ordering $O_E = E_1, E_2, \dots, E_{\alpha(G,\lambda)} = e_1^1, e_2^1, \dots, e_{k_1}^1, e_1^2, \dots, e_{k_{\alpha(G,\lambda)}}^{\alpha(G,\lambda)}$ of all edges e in G, that receive at least one label in the temporal graph (G, λ). Note that some edges repeat in this ordering (as one edge can have multiple labels and therefore appears in multiple sets E_i). We now create a new labeling λ' of G, where the i-th edge in the ordering O_E receives the label i. This results in a temporal graph (G, λ'), where each label occurs in exactly one edge. 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 $i = 1, 2, \dots, m$, if (G, λ') contains the edge e with time-label i, we add a phone call $c(i)$ between the two endpoints of the edge e. Since both (G, λ) and (G, λ') are temporally connected, it follows that the sequence $c(1), c(2), \dots, c(m)$ of calls results in every agent knowing every secret. This completes the proof. $\square$

Now denote with $f(n)$ the minimum number of calls needed to complete gossiping among a set A of n agents, where only a specific set of pairs of agents $B \subseteq \binom{A}{2}$ 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 as edges. Then it is known by Bumby [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 3 and Lemma 4 and by the results of Bumby [9].

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:*

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

We now present a procedure to obtain labelings that achieve the bounds from Theorem 5, for an example see Figure 1. Let G be a connected graph, we distinguish two cases, one where G contains no C_4 and the second one where there is at least one C_4 as a subgraph in G.

Labelings for graphs G with no C_4. We start by finding a spanning tree T of G (for example, by using a BFS algorithm). We can now in linear time determine the diameter d_T of the tree T, and two vertices v_s and v_t that are exactly d_T apart. Let us denote with $P_d = (v_s = v_0, v_1, v_2, \dots, v_{d-1}, v_d = v_t)$ the path between vertices v_s and v_t. We now label the path P_d as follows: for all $i \in \{1, 2, \dots, d-1\}$ the edge $e_i = v_{i-1}v_i$ receives the labels i and $2d - i$, and the edge $e_d = v_{d-1}v_t$ receives only the label d. We have now created two temporal paths between v_s and v_t, one starting at v_s at time 1 and finishing at v_t at time d, and the other starting at v_t at time d and finishing at v_s at time $2d - 1$. Clearly all the vertices of P_d can reach each other.

Let v be a leaf of T that is not in P_d. We now denote with P_v the path connecting v to P_d, more precisely, let $v_i \in P_d$ be the first vertex of P_d that is on a unique path from v to v_t in T. Then $P_v = (v = u_0, u_1, u_2, \ldots, u_k = v_i)$ for some $k \geq 1$. We now label each edge $f_j = u_{j-1}u_j$ of P_v, where $j \in \{1, 2, \ldots, k\}$, with the labels j and $2d - j$. Note that since the diameter of T is d it follows that the distance between v and v_t is at most d, therefore $k + i \leq d$ (resp. $k + (d - i) \leq d$). Thus, there exists a temporal path from v to v_t starting at time 1, reaching vertex $v_i \in P_v \cap P_d$ at time k, continuing to v_t with the edge v_iv_{i+1} at the time $i + 1$ and finishing at time d. Similarly, there is a temporal path from v_t to v starting at time d, reaching vertex v_i via the edge v_iv_{i+1} at time $2d - (i + 1)$, continuing towards v at time $2d - k$ and finishing at time $2d - 1$.

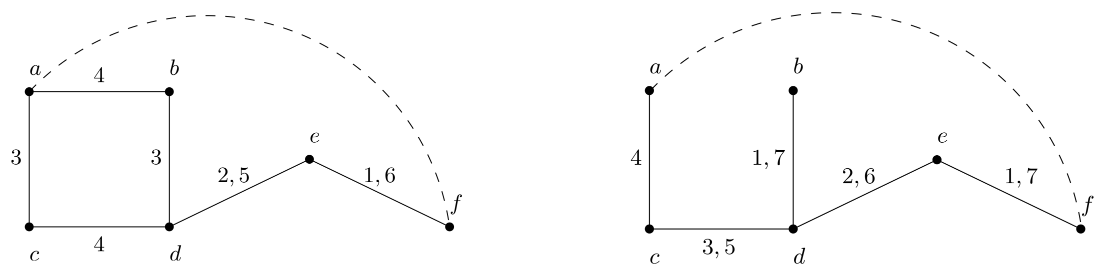
We repeat the above procedure to label all of the remaining edges of T. Afterwards, there is a temporal path from every vertex v to v_t (arriving at time d) and there is a temporal path from v_t to all other vertices v (starting at time d). Recall that v_t is a leaf of T and let v_{d-1} denote its neighbor. We can construct a temporal walk from a vertex v to another vertex v' as follows. Start at v and move along the temporal path from v to v_t until v_{d-1}. The arrival time at v_{d-1} is strictly smaller than d. Now consider the temporal path from v_t to v'. This temporal path visits v_{d-1} and continues from this vertex at a time strictly larger than d. It follows that we can obtain a temporal walk from v to v' through v_{d-1}.

This procedure assigns exactly one label to edge v_dv_{d-1}, and exactly two labels to every other edge of T. Therefore, we end up with a labeling λ of G, that uses $2n - 3$ labels.

Labelings for graphs G that contain a C_4. We first find a C_4 in G, $C = (a, b, c, d)$. Then, we contract C to a single vertex p, i.e., we remove the C_4 from G and add a vertex p that is connected to all of the vertices in G that have at least one neighbor in the removed C_4, and create a graph G_p. In G_p we find a spanning tree T_p rooted at p. Having determined T_p, we can now construct the spanning subgraph H of G, that is a tree with a C_4. We do this as follows; we iterate over edges $e = (v_iv_j) \in E(T_p)$ and distinguish two cases. First, if $p \notin e$ then we add e to H, second, if $p \in e$ (w.l.o.g. $p = v_i$) then we add to H the edge from v_j to one of the vertices a, b, c, d (we know that at least one such edge exists in G, if there is more than one such edge we arbitrarily chose one).

We have now created a spanning subgraph H of G which is a C_4, $C = (a, b, c, d)$ together with 4 subtrees, each rooted in a distinct vertex of C. Denote these trees as T_a, T_b, T_c, T_d. We note that these subtrees are allowed to be formed just by the root. For a vertex $u \in \{a, b, c, d\}$ we label edges of a tree T_u by starting at a leaf vertex v in T_u and travel towards the root u, using the unique path $P^{T_u}_{v,u}$ between v and u in T_u, where the first edge of $P^{T_u}_{v,u}$ (incident to v) receives the label 1, second edge the label 2, etc., up to the last edge (incident to u) that receives the label d_v, where d_v is the length of the path $P^{T_u}_{v,u}$. We now repeat this procedure for all the leaf vertices in T_u. At this point, we have assigned at least one label to each edge of T_u. On edges with more than one label we keep the highest label and discard all others. We end up with a labeling λ of the tree T_u, where each edge has exactly one label and for each vertex in T_u there exists a temporal path to the root vertex u. Let us denote with d_u the value of the highest label we have assigned to any of the edges in T_u. We repeat this procedure for every $u \in \{a, b, c, d\}$. Now, let $r_H = \max_{u \in \{a,b,c,d\}}\{d_u\}$ be the maximum label we have used on any of the edges of all T_u (for $u \in \{a, b, c, d\}$) so far. We label the edges of the C_4 as follows: $\lambda(ab) = \lambda(cd) = r_H + 1$ and $\lambda(ad) = \lambda(bc) = r_H + 2$. All of the above results in the existence of a temporal path from each vertex $v \in V(H)$ to all of the four vertices $\{a, b, c, d\}$ of the C_4. Moreover, note that each such temporal path reaches the vertices of the C_4 by the time $r_h + 2$. If we now ensure the existence of a temporal path from each vertex $u \in \{a, b, c, d\}$ of the C_4 to all of the vertices in its corresponding tree T_u, where the starting time of the temporal path is at least $r_H + 3$, then we have successfully constructed the labeling λ that temporally connects all pairs of vertices. To achieve this we do the following: every edge with a label i, that is not a part of a C_4, gets the second label $2r_H + 3 - i$. It is not hard to see that these new labels now ensure the existence of a temporal path from each $u \in \{a, b, c, d\}$ to every vertex in its corresponding T_u using a temporal path that starts at time $r_H + 3$ or later.

Our procedure results in a labeling λ of H that admits a temporal path among all pairs of vertices. The labeling assigns just 1 label to all four edges of the C_4 and exactly 2 labels to all other edges, which achieves the bound of $2n - 4$ labels in total.



(a) An example of a labeling that temporally connects a graph that contains a C_4, where the C_4 edges receive one label and all other edges of a spanning tree receive two labels.

(b) An example of a labeling that temporally connects a graph that does not contain a C_4, where one edge receives one label and all other edges of a spanning tree receive two labels.

Figure 1: An example of labeling meeting bounds from Theorem 5 for a graph containing a C_4 (Figure 1a) and a graph without a C_4 (Figure 1b). We mark the edges of a spanning tree or spanning tree with a C_4 with a solid line and all other edges with a dashed line.

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. We want to point out that these results contrast the APX-hardness for the minimization analogue of MAL on general directed graphs, proven in [33], while the more relaxed version of ML remains still open. For notation uniformity, we call these minimization problems $\mathrm{ML}_{directed}$ and $\mathrm{MAL}_{directed}$, respectively. First we define a *canonical layering* of a DAG, which is useful for our algorithm.

Definition 3. *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 $0 \leq i \leq d$, 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 2.

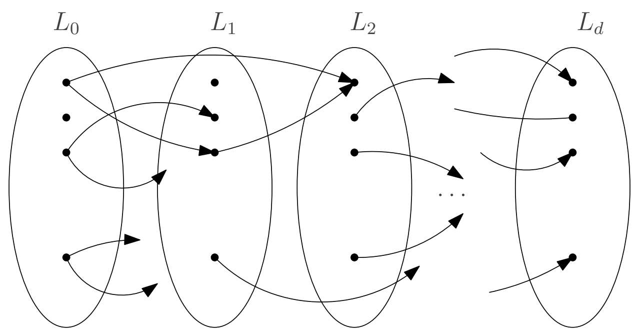


Figure 2: Example of a canonical layering.

Lemma 6. *Let $G = (V, E)$ be a DAG with n vertices and m arcs. We can produce the canonical layering of G in $O(n + m)$ time.*

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. 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. $\square$

We use the canonical layering to prove the following result.

Theorem 7. *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 6. 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 6, 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 w from S_u. At the end of this procedure, the set S_u contains exactly those vertices $v \in N^+(u)$, 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" 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). $\square$

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 equal to the diameter d of the input graph. It is worth noting here that, for any $x \geq 1$, the complexity of MAL remains open in the case where the age is allowed to be at most $d + x$.

To prove the NP-hardness we provide a reduction from the 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 compute a truth assignment τ of ϕ which satisfies the greatest possible number of clauses.

Max-Cut on cubic graphs reduces to MonMaxXOR(3) using the following reduction. Given a cubic graph G for each vertex $v \in V(G)$ create a variable x_v in the MonMaxXOR(3) formula ϕ_G. For every edge $uv \in E(G)$, add the clause $(x_v \oplus x_u)$ to ϕ_G. It is easy to see that computing a maximum cut in G (i.e., a partition of $V(G)$ into two sets A and $\overline{A}$ such that the number $|\{uv \in E(G) : u \in A, v \in \overline{A}\}|$ of edges between A and $\overline{A}$ is maximized), is equivalent with computing a maximum number of satisfied clauses in ϕ_G. Since Max-Cut is known to be NP-hard even in cubic graphs [5], we conclude the following.

Theorem 8. MonMaxXOR(3) *is NP-hard.*

We now describe 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 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 $|\lambda_\phi| \leq 7n^2 + 49n - 8k$ labels, where the maximum used label is 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 λ_ϕ, in each variable gadget there are exactly two labelings that temporally connect starting and ending vertices, which corresponds to the True or False truth assignment of the variable in the input formula ϕ. For every clause $(x_i \oplus x_j)$ we identify 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 (for an illustration see Figure 3), 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 easier analysis we fix the following notation. Vertex $s_i \in BX_i$ is called *starting vertex* of X_i, vertices t_i^ℓ ($\ell \in \{1, 2, 3\}$) are called *ending vertices* of X_i. Vertices $a_i, b_i, c_i, d_i, e_i, f_i^\ell, g_i^\ell, h_i^\ell, m_i^\ell$ (resp. $\overline{a_i}, \overline{b_i}, \ldots \overline{m_i}^\ell$) are called the left (resp. the right) vertices of X_i. A path connecting s_i, t_i^ℓ that passes only through the left (resp. the right) vertices 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.*

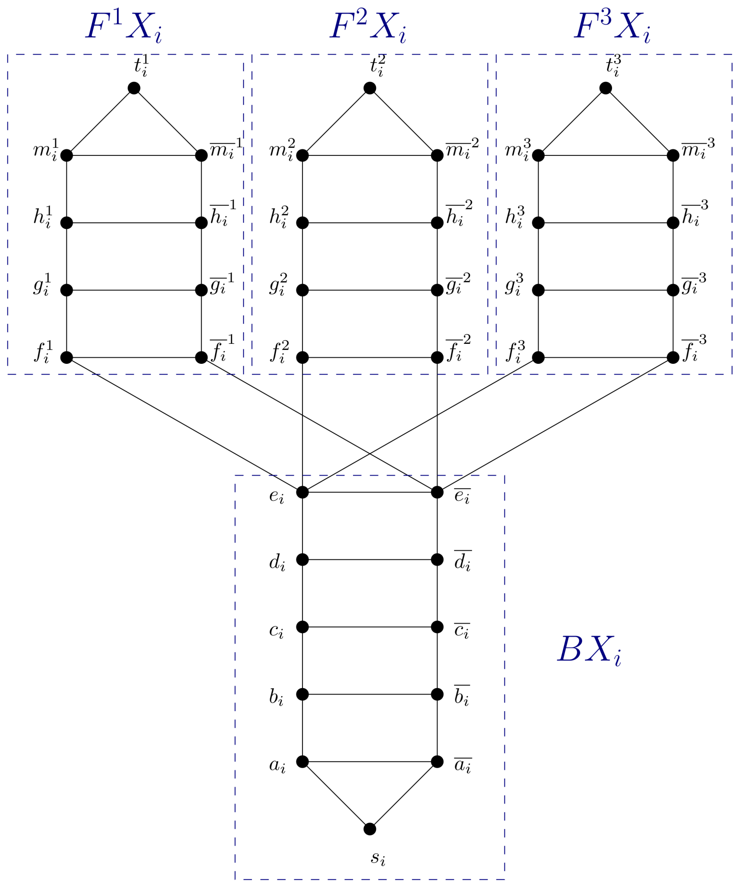


Figure 3: An example of a variable gadget X_i in G_ϕ, corresponding to the variable x_i from ϕ.

Connecting variable gadgets. There are two ways in which we connect two variable gadgets, depending on 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 (for an illustration see Figure 4). In this case we add the following edges between the variable gadgets X_i and X_j:

   - from e_i (resp. $\overline{e_i}$) to $f_j^{\ell'}$ and $\overline{f_j}^{\ell'}$, where $\ell' \in \{1, 2, 3\}$,
   - from e_j (resp. $\overline{e_j}$) to f_i^ℓ and $\overline{f_i}^\ell$, where $\ell \in \{1, 2, 3\}$,
   - from d_i (resp. $\overline{d_i}$) to d_j and $\overline{d_j}$.

We call these edges the *inter-variable edges*.

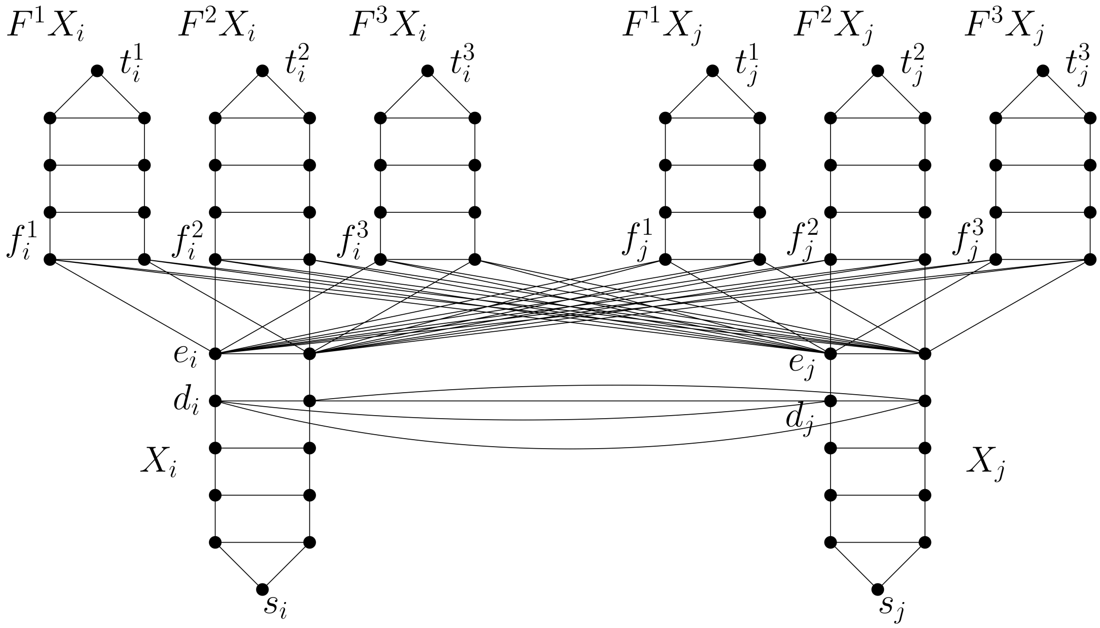


Figure 4: An example of two non-intersecting variable gadgets and inter-variable edges among them.

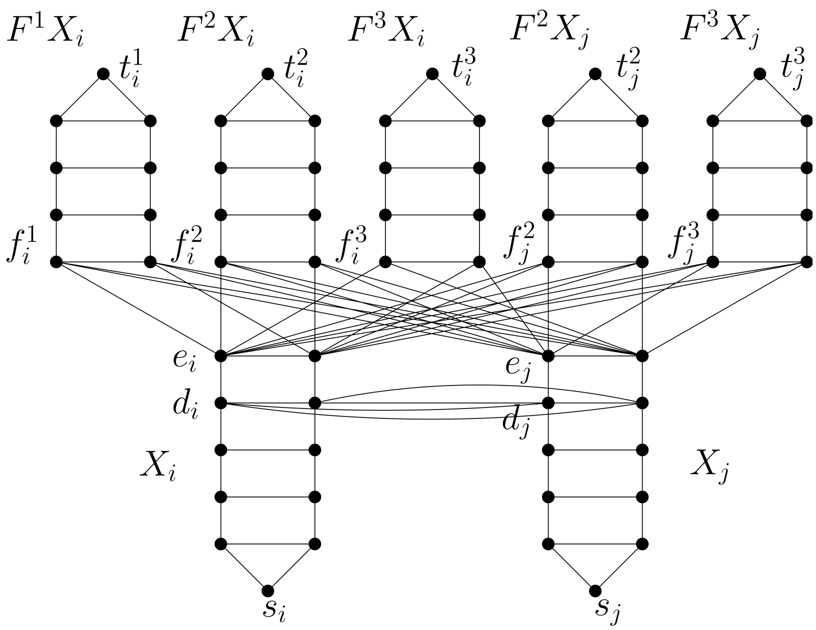


Figure 5: An example of two intersecting variable gadgets X_i, X_j corresponding to variables x_i, x_j, that appear together in some clause in ϕ, where it is the third appearance of x_i and the first appearance of x_j.

2. Two variables appear in a clause together (for an illustration see Figure 5). 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 $\overline{d_j}$.

This finishes the construction of G_ϕ.

Remark 1. *In our construction of G_ϕ, the three "forks" F^1X_i, F^2X_i, F^3X_i of a variable gadget X_i (see Figures 3, 4, and 5) can be also interpreted as "clause gadgets", in the following sense. Each fork in the construction corresponds to exactly two variables, say x_i and x_j, where the formula ϕ contains the clause $(x_i \oplus x_j)$. In the construction, this fork appears both as F_i^h and as F_j^ℓ, for some $1 \le h, \ell \le 3$, which is essentially the intersection of the two variable gadgets X_i and X_j.*

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

Lemma 9. *The diameter 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 there exists a path of length at most 10 between any two vertices from two different variable gadgets, which proves the desired bound.

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 each of the following sets $\{a_i, \overline{a_i}\}, \{b_i, \overline{b_i}\}, \{c_i, \overline{c_i}\}, \{d_i, \overline{d_i}\}, \{e_i, \overline{e_i}\}, \{f_i^\ell, \overline{f_i}^\ell\}, \{g_i^\ell, \overline{g_i}^\ell\}, \{h_i^\ell, \overline{h_i}^\ell\}, \{m_i^\ell, \overline{m_i}^\ell\}$, 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. A path between any two ending vertices $t_i^{\ell_1}, t_i^{\ell_2}$ (where $\ell_1, \ell_2 \in \{1,2,3\}$ and $\ell_1 \ne \ell_2$), has to go through at least one of the vertices from each of the following sets $\{m_i^{\ell_1}, \overline{m_i}^{\ell_1}\}$,$\{m_i^{\ell_2}, \overline{m_i}^{\ell_2}\}$, $\{h_i^{\ell_1}, \overline{h_i}^{\ell_1}\}$,$\{h_i^{\ell_2}, \overline{h_i}^{\ell_2}\}$, $\{g_i^{\ell_1}, \overline{g_i}^{\ell_1}\}$,$\{g_i^{\ell_2}, \overline{g_i}^{\ell_2}\}$, $\{f_i^{\ell_1}, \overline{f_i}^{\ell_1}\}$,$\{f_i^{\ell_2}, \overline{f_i}^{\ell_2}\}$, $\{e_i, \overline{e_i}\}$. Similarly as before, the shortest path uses exactly one vertex from each set and is of size 10. It is not hard to see that the distance between any other vertex $v \in X_i \setminus \{s_i, t_i^\ell\}$ (where $\ell \in \{1,2,3\}$) and the starting vertex or one of the ending vertices is at most 9, as vertex v lies on one of the shortest (s_i, t_i^ℓ) or $(t_i^{\ell_1}, t_i^{\ell_2})$ paths (where $\ell_1, \ell_2 \in \{1,2,3\}$ and $\ell_1 \ne \ell_2$), but it is not an endpoint of it. By the similar reasoning there exists a path between any two vertices $u, v \in X_i \setminus \{s_i, t_i^\ell\}$ (where $\ell \in \{1,2,3\}$), of distance at most 9. Therefore, the diameter of X_i is 10.

Now we want to show that the distance between any two vertices from different variable gadgets is at most 10. Let us start with the case where two variable gadgets X_i and X_j share no fork (i.e., x_i and x_j do not appear in the same clause of ϕ). A path between s_i and t_j^ℓ (for $\ell \in \{1,2,3\}$) travels through at least one of the vertices from the following sets $\{a_i, \overline{a_i}\}, \{b_i, \overline{b_i}\}, \ldots, \{e_i, \overline{e_i}\}, \{f_j^\ell, \overline{f_j}^\ell\}, \{g_j^\ell, \overline{g_j}^\ell\}, \ldots, \{m_j^\ell, \overline{m_j}^\ell\}$. The shortest path goes through exactly one vertex in each of the sets, therefore it is of length 10. From this it also follows that there exists a path between any base vertex $v \in BX_i$ and fork vertex $u \in FX_j$ of length at most 10. Next, observe a path between s_i and s_j that goes through at least one of the vertices from each of the following sets $\{a_i, \overline{a_i}\}, \{b_i, \overline{b_i}\}, \{c_i, \overline{c_i}\}, \{d_i, \overline{d_i}\}, \{d_j, \overline{d_j}\}, \{c_j, \overline{c_j}\}, \{b_j, \overline{b_j}\}, \{a_j, \overline{a_j}\}$. Again, the shortest path will use exactly one vertex in each set and is of distance 9. Therefore, all of the a, b, c, d vertices from X_i and X_j are at distance at most 9 from each other. Since the path (e_i, f_j^1, e_j) is of length 2 and $(e_i, d_i, d_j, c_j, b_j, a_j, s_j)$ is of length 6 it follows that e_i is at distance at most 3 from e_j and 6 from s_j. Therefore, all of the vertices from BX_i and BX_j are at distance at most 9 from each other. Lastly, a path between $t_i^{\ell_1}$ and $t_j^{\ell_2}$ (where $\ell_1, \ell_2 \in \{1,2,3\}$) travels through at least one of the vertices from the following sets $\{m_i^{\ell_1}, \overline{m_i}^{\ell_1}\}, \{h_i^{\ell_1}, \overline{h_i}^{\ell_1}\}, \{g_i^{\ell_1}, \overline{g_i}^{\ell_1}\}, \{f_i^{\ell_1}, \overline{f_i}^{\ell_1}\}, \{e_i, \overline{e_i}\}, \{f_j^{\ell_2}, \overline{f_j}^{\ell_2}\}, \{g_j^{\ell_2}, \overline{g_j}^{\ell_2}\}, \{h_j^{\ell_2}, \overline{h_j}^{\ell_2}\}, \{m_j^{\ell_2}, \overline{m_j}^{\ell_2}\}$. Since the shortest path visits exactly one vertex from each set, it is of length 10. Similarly as before, it follows that there is a path between any two vertices $u \in F^{\ell_1}X_i$ and $v \in F^{\ell_2}X_j$ (where $\ell_1, \ell_2 \in \{1,2,3\}$) of distance at most 10. Therefore, we get that the diameter of a subgraph of G_ϕ that contains any two variable gadgets that do not share a fork is 10. In the case when two variable gadgets X_i and X_j share a fork, it is not hard to see that the shortest path among any two vertices $u \in X_i$ and $v \in X_j$ does not become grater than in the case when two variable gadgets do not share a fork.

All together it follows that the distance among any two vertices in G_ϕ is at most 10. $\square$

In the following, let $\mathrm{OPT}_{\textsc{MonMaxXOR}(3)}(\phi)$ denote the size of an optimal solution for $\textsc{MonMaxXOR}(3)$ on instance ϕ, and let $\mathrm{OPT}_{\mathrm{MAL}}(G_\phi, 10)$ denote the size of an optimum solution for MAL on instance $(G_\phi, 10)$.

Lemma 10. *If $OPT_{\textsc{MonMaxXOR}(3)}(\phi) \geq k$ then $OPT_{\mathrm{MAL}}(G_\phi, 10) \leq 7n^2 + 49n - 8k$, 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 $|\lambda_\phi| \leq 7n^2 + 49n - 8k$ labels, such that (G_ϕ, λ_ϕ) is temporally connected and $\alpha(G_\phi, \lambda_\phi) = 10$. Recall that since ϕ is an instance of $\textsc{MonMaxXOR}(3)$ 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 6.

1. If a variable x_i from ϕ is set to $\textsc{True}$ 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 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 now need to prove that there exists a temporal path among any two vertices in X_i. Suppose x_i is set to TRUE in the truth assignment τ of ϕ (the case of x_i being FALSE is analogous). By the construction of λ_ϕ, there are temporal paths from s_i to any of the t_i^ℓ, where $\ell \in \{1, 2, 3\}$, and vice versa. Labeling λ_ϕ of G_ϕ gives rise to the following temporal paths. There is a temporal path from the starting vertex s_i to the ending vertex t_i^ℓ, where $\ell \in \{1, 2, 3\}$, which uses the left path of X_i, and labels $1, 2, \ldots, 10$. Similarly, it holds for the temporal (t_i^ℓ, s_i)-path. The temporal path connecting two ending vertices $t_i^{\ell_1}, t_i^{\ell_2}$ (where $\ell_1, \ell_2 \in \{1, 2, 3\}$ and $\ell_1 \neq \ell_2$), uses first the left path of the fork $F^{\ell_1} X_i$, with labels 1 to 5, to reach e_i, and then continues on the left path of the fork $F^{\ell_2} X_i$ using labels 6 to 10. Since the temporal paths among starting and ending vertices use the left path of the gadget X_i it follows that all vertices on the left path reach all starting and ending vertices, and consequently, they also reach each other. Any remaining vertex, i.e., a vertex on the right path of the gadget X_i, can reach the starting vertex using first their corresponding connecting edge at time 1, and then the remaining part of the temporal path from t_i^ℓ (for $\ell \in \{1, 2, 3\}$) to s_i. Similarly, it holds for the temporal paths towards all of the ending vertices. In the case of temporal paths from s_i (or t_i^ℓ) to the vertices on the right side of X_i, the temporal paths start with the edges of the left path of X_i at time 1 and finish using the corresponding connecting edge at time 10. Lastly, temporal paths among two vertices from the right path of X_i use as a first and last edge the corresponding connecting edge at time 1 and 10 respectively, and a part of the (s_i, t_i^ℓ) or (t_i^ℓ, s_i)-temporal path. This proves that the labeling λ_ϕ of 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 inter-variable edges as follows:

   - the edge $d_i d_j$, connecting the left path of BX_i with the left path of BX_j, gets labels 5 and 6,

- three edges of the form $e_i f_j^{\ell'}$ ($\ell' \in \{1,2,3\}$) that connect the left path of BX_i to the left paths of $F^{\ell'}X_j$ get labels 5 and 6,
- three edges of the form $e_j f_i^{\ell}$ ($\ell \in \{1,2,3\}$) that connect the left path of BX_j to the left paths of $F^{\ell}X_i$ get labels 5 and 6.

We have assigned 14 labels to 7 inter-variable edges that connect both variable gadgets, while the number of labels assigned to each variable gadget remains the same. Note that the other three 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 inter-variable edges, where the labeled inter-variable edges are chosen appropriately. For an example see Figure 6a.

Since labeling inter-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. First observe that there is a unique temporal path from s_i to t_j^ℓ (for $\ell \in \{1,2,3\}$), that first uses the left path of the base of X_i with labels $1,2,3,4,5$, the inter-variable edge $e_i f_j^\ell$ with label 6 and continues to t_j^ℓ using the left path of the fork F_j^ℓ with labels $7,8,9,10$. The reverse (t_j^ℓ, s_i)-temporal path uses the same edges with labels $1,2,\ldots,10$, as defined by λ_ϕ. From the above it follows that there exists a temporal path from any vertex in the base of BX_i to any vertex in a fork F_j^ℓ and vice versa (note, if any of the starting/ending vertices is not on a left path of X_i or X_j we use corresponding connecting edges at time 1 or 10). Next, we show that there is a temporal path between two ending vertices $t_i^{\ell_1} \in X_i$ and $t_j^{\ell_2} \in X_j$ (where $\ell_1, \ell_2 \in \{1,2,3\}$). More specifically, the $(t_i^{\ell_1}, t_j^{\ell_2})$-temporal path first uses the left side of the fork $F_i^{\ell_1}$ with labels $1,2,3,4,5$ to reach the vertex $e_i \in X_i$ and then uses the inter-variable edge $e_i f_j^{\ell_2}$ at time 6 and continues on the left side of the fork $F_j^{\ell_2}$ with labels $7,8,9,10$ to reach $t_j^{\ell_2}$. Thus it holds that any vertex in any of the forks of X_i can reach any vertex in any of the forks of X_j. Note that the last temporal path proves also that $e_i \in X_i$ reaches all of the vertices in all of the forks of X_j (and vice versa). Let us now show that e_i reaches also all the vertices in the base of X_j (and vice versa). First, the (e_i, e_j)-temporal path is of length 2, starts with the inter-variable edge $e_i f_j^{\ell_2}$ at time 5 and finishes with the edge $f_j^{\ell_2} e_j$ at time 6. Second, e_i reaches vertex s_j using the temporal path that travels through vertices $e_i, d_i, d_j, c_j b_j, a_j, s_j$ with labels $5,6,7,8,9,10$ on the respective edges. Conversely, the (s_j, e_i)-temporal path travels through the same vertices $s_j, a_j, b_j, c_j, d_j, d_i, e_i$ with labels $1,2,3,4,5,6$ on the respective edges. From the above three temporal paths it follows that e_i in fact does temporally reach all of the vertices in the base of X_j and vice versa. Lastly, we want to prove that all of the remaining base vertices of X_i (i.e. vertices of form a, b, c, d) reach all of the remaining base vertices in X_j. To do so we just have to provide a temporal path from s_i to s_j. This temporal path travels through the vertices $s_i, a_i, b_i, c_i, d_i, d_j, c_j, b_j, a_j, s_j$ using labels $1,2,3,4,5,7,8,9,10$ on the respective edges. All of the above proves that there exists a temporal path among any two vertices in

X_i and X_j, when X_i and X_j share no 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 inter-variable edges:

   - the edge $d_i d_j$ connecting the left path of BX_i and BX_j gets labels 5 and 6,
   - 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 the left paths of $F^{\ell'} X_j$ get labels 5 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 the left paths of $F^{\ell} X_i$ get labels 5 and 6.

   We have assigned 10 labels to 5 inter-variable edges that connect both variable gadgets. Note that 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 inter-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 $x_i \neq x_j$. Then the labeling λ_ϕ of $F^r X_i$ coincides with the labeling of $F^{r'} X_j$. Therefore λ_ϕ uses 16 less labels on the shared fork.
   b) In the case when $x_i = x_j$. 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 8 less labels on the shared fork.

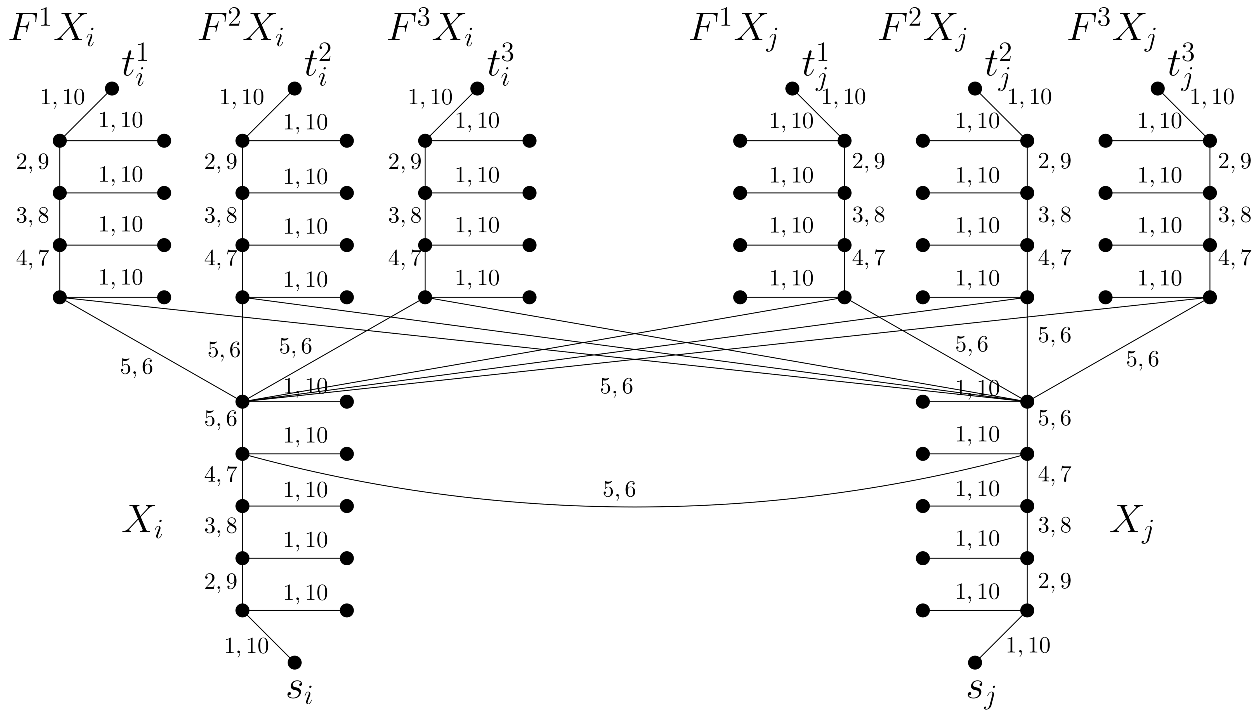
   Identifying two forks $F^r X_i = F^{r'} X_j$ and labeling them using the union of both labelings on each fork, clearly preserves temporal paths among all the vertices from X_i (resp. X_j). What remains to check is that all vertices in X_i reach all the vertices in X_j. This follows from the same proof as in the previous case, where the paths between the two variable gadgets use the appropriate inter-variable edges. Note, since the fork $F^r X_i = F^{r'} X_j$ is in the intersection, the inter-variable edges from X_i (resp. X_j) to $F^r X_i = F^{r'} X_j$ do not exist. 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 14 labels on inter-variable edges among any two variables, from which we have to subtract the following:

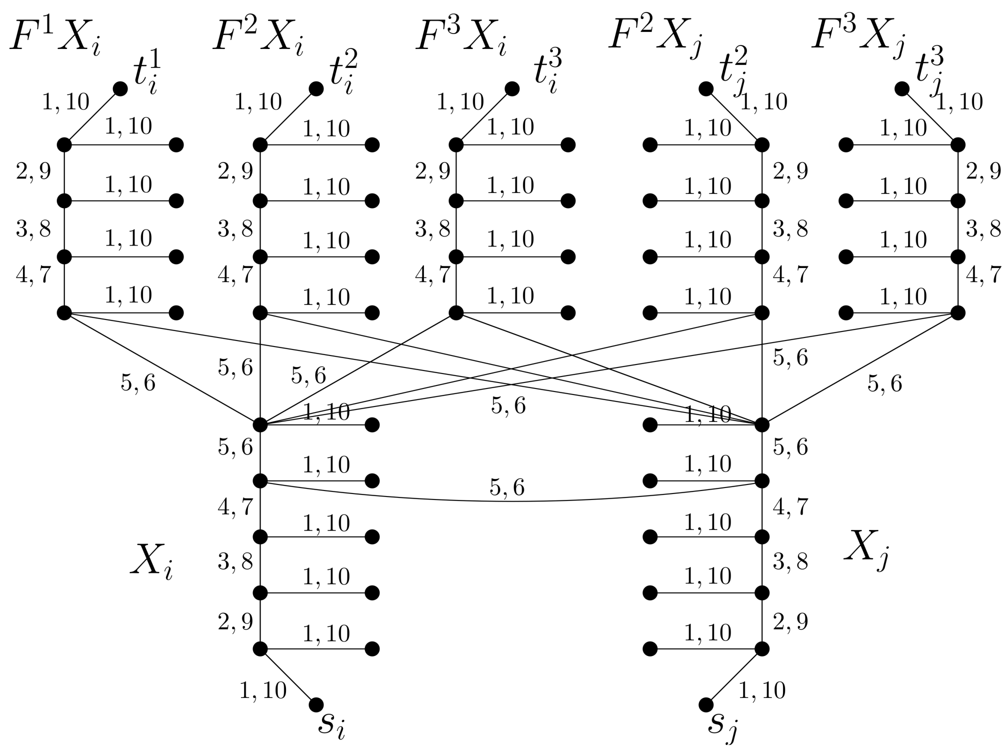
- 4 labels for each pair of inter-variable edges between two variables that appear in the same clause,

- 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.



(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. Here $F^3X_i = F^1X_j$ and $t_i^3 = t_j^1$.

Figure 6: Example of the labeling λ on variable gadgets X_i, X_j and inter-variable edges between them, where x_i is TRUE and x_j FALSE in ϕ. Note that edges that are not labeled are omitted.

Altogether this sums up to $74n + 14\frac{n(n-1)}{2} - 4m - 16k - 8(m-k)$. As a result, given that τ satisfies a minimum of k clauses of ϕ, the labeling λ_ϕ admits at most $7n^2 + 49n - 8k$ labels. $\square$

Before proving the statement in the other direction, we have to show some structural properties. Let us fix the following notation. 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 $F_i^{\ell'}$. 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 that 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). The following technical lemma will allow us to prove the correctness of our reduction.

Lemma 11. *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 (i.e. X_i does not admit all left upward and downward paths, or all right upward and downward paths).

First, we 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 the fork) and at least one partial downward path, or vice versa. 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 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* 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^3X_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. Indeed, if otherwise F^1X_i (resp. F^2X_i) has no partial downward path on the left side of X_i, then there is no path with at most 10 edges from t_i^1 (resp. t_i^2) to t_i^3, a contradiction.

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 $\overline{e_i}\overline{f_i}^3$ 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 $\overline{e_i}\overline{f_i}^3$ 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 can simply 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 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 $\overline{e_i}\overline{f_i}^3$ 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 $\overline{e_i}\overline{f_i}^1$ and $\overline{e_i}\overline{f_i}^3$ of the forks F^1X_i and F^3X_i, respectively. Third, remove the label "5" from the right bridge edge $\overline{e_i}\overline{f_i}^1$ 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 $e_if_i^3$ 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 $\overline{e_i}\overline{f_i}^3$ 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 $e_if_i^3$ 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 bridge edges (within the variable gadgets) and the inter-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. $\square$

Lemma 12. *If $OPT_{\mathrm{MAL}}(G_\phi, 10) \leq 7n^2 + 49n - 8k$ then $OPT_{\mathrm{MonMaxXOR(3)}}(\phi) \geq k$, where n is the number of variables in the formula ϕ.*

Proof. Let λ_ϕ be an optimum solution to $\mathrm{MAL}(G_\phi, 10)$, which uses at most $7n^2 + 49n - 8k$ labels. We will prove that there exists a truth assignment τ that satisfies at least k clauses of ϕ. Lemma 11 implies that every variable gadget of G_ϕ is either left-aligned or right-aligned in λ_ϕ. Throughout the proof, we consider an arbitrary variable gadget X_i, and we assume w.l.o.g. that X_i is left-aligned.

First, we count the minimum number of labels needed in λ_ϕ, so that all temporal paths among vertices of X_i exist. Recall that s_i is at distance 10 from any t_i^ℓ, where $\ell \in \{1, 2, 3\}$, and t_i^ℓ is at distance 10 from any $t_i^{\ell'}$, where $\ell' \in \{1, 2, 3\} \setminus \{\ell\}$. Therefore, any temporal path connecting any two of the vertices in $\{s_i, t_i^1, t_i^2, t_i^3\}$ is of length 10 and must use labels $1, 2, 3, \dots, 10$ along its edges (in this order). Which implies that the k-th edge on the path $(s_i, a_i, b_i, c_i, \dots, m_i^\ell, t_i^\ell)$ admits at least the labels $k, 11-k$. In total, there must exist at least $5 \cdot 2 = 10$ labels on the base BX_i, at least $4 \cdot 2 = 8$ labels on each fork $F^\ell X_i$, and at least 2 labels on each left-bridge edge $e_i f_i^\ell$. That is, we have at least $10 + 3 \cdot (8 + 2) = 40$ labels to temporally connect the vertices $\{s_i, t_i^1, t_i^2, t_i^3\}$ (and also all vertices within the paths among them at the left side of the variable gadget X_i). Furthermore, we need at least two labels for each vertex $\overline{y}$ at the right side of X_i such that there is a temporal path to and from $\overline{y}$ in X_i, i.e., we need at least $17 \cdot 2 = 34$ more labels. That is, within the whole variable gadget X_i we need in total at least $40 + 34 = 74$ labels in λ_ϕ.

Let X_j be a variable gadget in G_ϕ that does not share a fork with X_i. W.l.o.g. we can assume that X_j is right-aligned (all other cases are symmetric). Using the arguing from above we know that the k-th edge of the path $(s_j, \overline{a_j}, \overline{b_j}, \dots, \overline{m_j}^{\ell'}, \overline{t_j}^{\ell'})$, where $\ell' \in \{1, 2, 3\}$, admits at least the labels $k, 11-k$. Observe that the shortest path between the starting vertex s_i of X_i and any of the ending vertices $t_j^{\ell'}$ ($\ell' \in \{1, 2, 3\}$) of X_j is of length 10 and is of the form $(s_i, a_i, \dots, e_i, \overline{f_j}^{\ell'}, \dots, \overline{t_j}^{\ell'})$. Therefore, the inter-variable edge $e_i \overline{f_j}^{\ell'}$ must admit at least the labels $5, 6$. This proves that all of the fork vertices in $F^{\ell'} X_j$ are reachable by the vertices of the base BX_i, and vice versa. Next, we have to make sure that there are temporal paths among vertices from BX_i and BX_j. The temporal path from s_i to s_j through vertices $s_i, a_i, \dots, d_i, \overline{d_j}, \dots, s_j$ is of length 9. Except for the edge $d_i \overline{d_j}$, all other edges are labeled, therefore $d_i \overline{d_j}$ has to admit either the label 5 or 6. Since these temporal paths do not pass through any of the e vertices of BX_i and BX_j, we still need to ensure that there is a temporal path from e_i to $\overline{e_j}$ and s_j, and vice versa (because of the symmetry this is enough to argue that all of the base vertices of BX_i and BX_j reach each other). First, to temporally connect e_i to $\overline{e_j}$ we do not require any new labels as e_i reaches $\overline{e_j}$ using edges $e_i \overline{f_j}^{\ell'}$ and $\overline{f_j}^{\ell'} \overline{e_j}$ at times 5 and 6, respectively. The $(\overline{e_j}, e_i)$-temporal path uses the same edges, but in the reverse order, first $\overline{e_j} \overline{f_j}^{\ell'}$ at time 5 and then $\overline{f_j}^{\ell'} e_i$ at time 6. The (e_i, s_j) temporal path that requires the least amount of extra added labels to X_i and X_j is the path on vertices $e_i, d_i, \overline{d_j}, \ldots, s_j$. This implies that the edge $d_i\overline{d_j}$ has to admit the label 6. Using the same path in the reverse direction, we get the temporal path from s_j to e_i that requires the edge $d_i\overline{d_j}$ to admit the label 5. Therefore, the edges $d_i\overline{d_j}, e_i\overline{f_j}^{\ell'}$ must all admit at least the labels $5, 6$. To sum up, 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 $2 \cdot (3+3) + 2 = 14$ extra labels on the inter-variable edges.

Lastly, let X_j be a variable gadget in G_ϕ that shares a fork with X_i. W.l.o.g. we can suppose that $F^1X_i = F^1X_j$. By the construction of G_ϕ, there exists a temporal path between all vertices in the fork $F^1X_i = F^1X_j$ and all vertices in X_i and X_j. As observed, these paths do not use the inter-variable edges. Using the same arguments as in the case when X_i and X_j do not share a fork, we get that a minimum labeling must use at least $2 \cdot (2+2)+2 = 10$ labels on the inter-variable edges.

The only thing left to inspect is the labeling in the intersecting fork. We distinguish the following two cases.

- The variable gadget X_j is right-aligned. Then, by the construction of G_ϕ, the fork $F^1X_i = F^1X_j$ is labeled using the same labeling as in the variable gadget X_i. 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, each edge in the fork $F^1X_i = F^1X_j$ admits two labels. This "saves" only 8 labels from the total number of labels used on variable gadgets X_i and X_j.

From the labeling λ_ϕ of G_ϕ we construct a truth assignment τ of ϕ as follows. 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.

Suppose that the labeling λ_ϕ satisfies exactly k^* clauses. As previously noted, λ_ϕ uses at least 74 labels on each variable gadget. Whenever two variable gadgets X_i, X_j do not appear in the same clause we need at least 14 extra labels on the inter-variable edges, and whenever X_i, X_j appear in the same clause we need at least 10 labels on the inter-variable gadgets. In the case where X_i, X_j appear in the same clause and both X_i and X_j are left-aligned (i.e. clause of ϕ is not satisfied) the common fork results in 8 less labels, while in the case where X_i is left-aligned and X_j is right-aligned (i.e. clause of ϕ is satisfied), the common fork results in 16 less labels. Consequently,

$$\begin{aligned} |\lambda_\phi| &\geq 74n + 14\binom{n}{2} - 14m + 10m - 8(m-k^*) - 16k^* \\ &= 67n + 7n^2 - 12m - 8k^* \\ &= 7n^2 + 49n - 8k^*. \end{aligned}$$

In the above derived equation, we used the fact that ϕ has $m = \frac{3}{2}n$ clauses. Since $|\lambda_\phi| = \mathrm{OPT}_{\mathrm{MAL}}(G_\phi, 10) \leq 7n^2 + 49n - 8k$ by the statement of the lemma, it follows that $k^* \geq k$, i.e., λ_ϕ satisfies at least k clauses of ϕ. $\square$

MAL is clearly in NP, since temporal connectivity can be checked in polynomial time [27]. Hence, the next theorem follows directly from Theorem 8 and Lemmas 10 and 12.

Theorem 13. MAL *is NP-complete on undirected graphs, 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 the number k of labels (which is a larger parameter than $|R|$), even if the maximum allowed age is a constant.

4.1. Computational Hardness of MSL

In this section, we prove that MSL is NP-complete.

Theorem 14. MSL *is NP-complete.*

Proof. MSL is contained in NP, since temporal connectivity can be checked in polynomial time [27]. To prove that the MSL is NP-hard we provide a polynomial-time reduction from the NP-complete Vertex Cover problem [26].

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 $\forall e \in E$ there exists a vertex $u \in S$ such that u is an endpoint of e?

In the paper [26], the NP-hardness reductions for Vertex Cover produce an instance (G, k) such that $(G, k-1)$ is a NO-instance of Vertex Cover independently of whether the original instance of the problem that was reduced from is a YES- or a NO-instance. Hence, we can make the following assumption. Let (G, k) be an input of the Vertex Cover problem, then G does not admit a vertex cover of size $k-1$. We denote $|V(G)| = n, |E(G)| = m$ and construct (G^*, R^*, k^*), the input of MSL using the following procedure (for an illustration see Figure 7). 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: $U_V = \{u_v \mid v \in V(G)\}$,
- an "edge-vertex" corresponding to every edge of G: $U_E = \{u_e \mid e \in E(G)\}$,
- $2n + 2m(6k + m)$ "dummy" vertices.

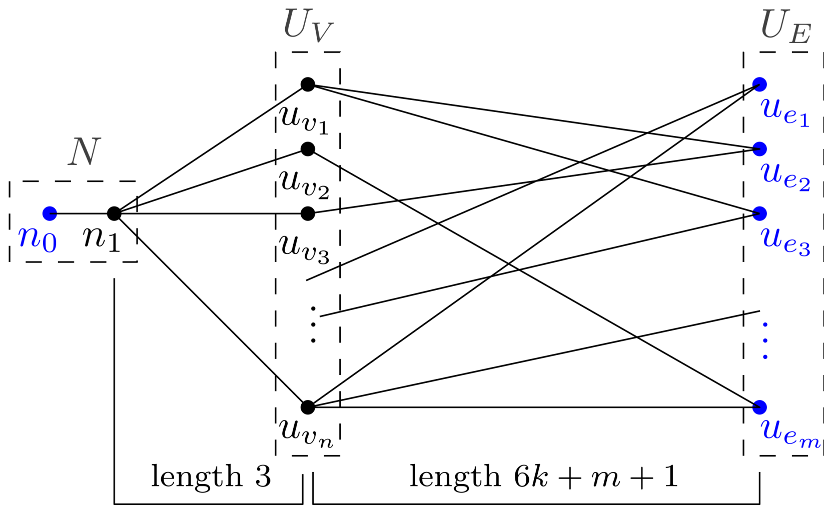


Figure 7: Illustration of the MSL instance produced by the reduction presented in the proof of Theorem 14, where the blue vertices represent set R^*.

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

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

We set $R^* = \{n_0\} \cup U_E$ and $k^* = 6k + 2m(6k + m + 1) + 1$. This finishes the construction. Note that G^* is a graph with $3n+m+2m(6k+m)+2$ vertices and $1+3n+2m(6k+m+1)$ edges. It is not hard to see that the described construction can be performed in polynomial time.

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.

($\Rightarrow$): 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 $6k + m + 1$ 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 $1, 2, \ldots, 6k + m + 1$. Doing this for all vertices $u_e \in U_E$ we use $m(6k + m + 1)$ labels. Now we label a path from each $v \in U_S$

to n_1 using labels $6k + m + 2, 6k + m + 3, 6k + m + 4$. 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 $\ell^* = 6k + m + 5$. Using this procedure we have created a forwarding path from each edge-vertex u_e to the start vertex n_0 and we used $3k + m(6k + m + 1) + 1$ 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 $\ell^* + 4, \ell^* + 5, \ldots, \ell^* + 4 + 6k + m$. We used extra $3k + m(6k + m + 1)$ labels and created a returning path from n_0 to each vertex in U_E.

Altogether, the constructed labeling uses $k^* = 6k + 2m(6k + m + 1) + 1$ labels. What remains to show is that there exists a temporal path between any pair of edge-vertices $u_e, u_f \in U_E$. We can construct a temporal walk W (possibly visiting the same vertex multiple times) from u_e to u_f as follows. Starting at u_e, we go along the forwarding path from u_e to n_0 until we reach n_1. By construction, we arrive at n_1 at time $\ell^* - 1$. Now consider the returning path from n_0 to u_f. This path goes through n_1 and, by construction, arrives at n_1 at time ℓ^*. Hence, we can extend the temporal walk W from n_1 to u_f by following the returning path from u_1 onward.

($\Leftarrow$): 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.

Consider the temporal paths connecting n_0 to the vertices in U_E. By the construction of G^* each temporal path from n_0 to a vertex in U_E passes through the set U_V. Hence, for each vertex $u_e \in U_E$ there is some vertex $u_v \in U_V$ such that u_v is temporally connected to u_e. Now consider the temporal paths connecting the vertices in U_E to n_0. Similarly to the argument above, by the construction of G^* each temporal path from a vertex in U_E to n_0 passes through the set U_V. Hence, each vertex $u_e \in U_E$ needs to be temporally connected to some vertex in $u_v \in U_V$. Fix some $u_e \in U_E$. We can conclude that there is a $u_v \in U_V$ such that u_e is temporally connected to u_v by a temporal path of length $6k+m+1$. Furthermore, there is an $u_{v'} \in U_V$ such that $u_{v'}$ is temporally connected to u_e by a temporal path of length $6k + m + 1$. If $u_v \neq u_{v'}$, then we attribute $12k + m + 2$ labels to vertex u_e. However, if $u_v = u_{v'}$, then the temporal path of length $6k + m + 1$ from u_e to u_v and the temporal path of length $6k + m + 1$ from u_v to u_e may share one time edge: Let P_{ve} be the unique path in G^* of length $6k + m + 1$ that connects u_v and u_e. Then the (u_v, u_e)-temporal path (resp. (u_e, u_v)-temporal path) traverses the edges of P_{ve} from u_v (resp. u_e) to u_e (resp. u_v), where the edges of P_{ve} are labeled strictly increasingly. Hence the two temporal paths may share at most one time edge. Therefore, in this case, we attribute at least $12k + 2m + 1$ labels to u_e. Overall, we attribute at least $m(12k + 2m + 1)$ labels to the vertices in U_E.

For a vertex $u_e \in U_E$, we call a temporal path from u_e to some $u_v \in U_V$ of length $6k + m + 1$ a forwarding path F_e for u_e. Similarly, we call a temporal path from some $u_{v'}$ to u_e of length $6k + m + 1$ a returning path R_e for u_e. For every u_e we have exactly one forwarding path and one returning path. This is true since every additional path would require at least an additional $6k + m$ labels on the edges between U_V and U_E, and then at most 1 label could be placed on the remaining edges, which would result in no temporal paths between $\{n_0, n_1\}$ and U_V.

This allows us to make the following observation. We define a partial order $<_{\text{label}}$ 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 $P <_{\text{label}} Q$ if all labels used in P are strictly smaller than the smallest label used in Q. We can observe that for any two $e, e' \in E$ with $e \neq e'$ we have that $F_e <_{\text{label}} R_{e'}$ 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 $R_e <_{\text{label}} F_e$ or R_e and F_e are incomparable with respect to $<_{\text{label}}$, otherwise we would reach a contradiction to the above observation. From this we can deduce that we attribute $12k + 2m + 2$ labels to each of the edges $e \in E$ with $F_e <_{\text{label}} R_e$, since F_e and R_e cannot share any label. Furthermore, there is at most one edge to which we can attribute $12k + 2m + 1$ labels, since, as argued earlier, if F_e and R_e are incomparable with respect to $<_{\text{label}}$, they can share at most one time edge. It follows now that we attribute at least $m(12k + 2m + 2) - 1$ labels to the vertices in U_E.

We now conclude that there are at most $6k + 2$ labels used on the edges connecting the sets N and U_V. Next, we identify two (potentially intersecting) subsets of U_V. We specify $U_V^+ \subseteq U_V$ such that $u_v \in U_V^+$ if and only if there exists a returning path R_e for some $e \in E$ that starts in u_v. Similarly, we specify $U_V^- \subseteq U_V$ such that $u_v \in U_V^-$ if and only if there exists a forwarding path F_e for some $e \in E$ that ends in u_v. First, observe that it cannot happen that $|U_V^+| > k$ and $|U_V^-| > k$. If this was true then we would use at least $2(k+1)3$ labels on the edges connecting the sets N and U_V, which would imply we have to attribute strictly less than $m(12k + 2m + 2) - 1$ labels to the vertices in U_E, which is not possible. Therefore, at least one of the sets U_V^+ or U_V^- is of size at most k. Assume that $|U_V^+| \leq |U_V^-|$ (the case where $|U_V^+| > |U_V^-|$ is symmetric). We claim that $S = \{v \mid u_v \in U_V^+\}$ is a vertex cover of size at most k for G. It is straightforward to see that S is a vertex cover for G: By the definition of U_V^+, for every $e \in E$ there is a returning path R_e starting in a vertex u_v such that v is one of the two endpoints of e. Hence, for every edge $e \in E$, one of its endpoints is contained in S. In the remainder, we show that $|S| \leq k$.

To this end, consider the temporal connections from n_1 to the vertices in U_E. Every edge-vertex $u_e \in U_E$ is temporally reachable from exactly one vertex-vertex $u_v \in U_V$ through the returning path R_e. Hence, there needs to be a temporal path from n_1 to u_v that arrives in u_v sufficiently early, such that it can be extended to u_e via the returning path R_e. We call this path from n_1 to u_v the *short returning path* R'_e of e. Similarly, consider the temporal connections from the vertices in U_E to n_1. Every edge-vertex $u_e \in U_E$ can reach exactly one vertex $u_v \in U_V$ via a forwarding path F_e. In order to reach n_0, there needs to be a temporal path from u_v to n_1 that starts sufficiently late, such that it can extend the forwarding path from u_e. We call this path from u_v to n_1 the *short forwarding path* F'_e of e.

Analogous as before, we define a partial order $<_{\text{label}}$ on the set $\mathcal{P}' = \{F'_e, R'_e \mid e \in E\} \cup \mathcal{P}$ of (short) forwarding and (short) returning paths. For two paths $P, Q \in \mathcal{P}'$, we say that $P <_{\text{label}} Q$ if all labels used in P are strictly smaller than the smallest label used in Q. Now consider two edges $e, f \in E$ such that the forwarding path F_e ends in u_v and the returning path R_f starts in $u_{v'}$ with $v \neq v'$. Then, by the construction of G^*, there must be a temporal path P from u_v to n_1 and a temporal path P' from n_1 to $u_{v'}$, such that F_e, P, P', and R_f can be concatenated to a temporal path from u_e to u_f. We can assume w.l.o.g. that $P = F'_e$

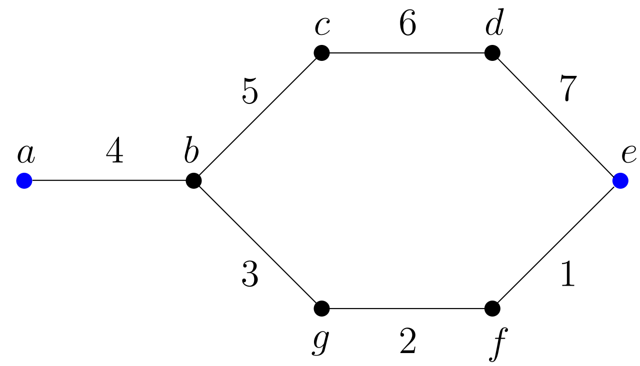


Figure 8: An example of an optimal labeling of an MSL instance, where the temporal (sub)-graph connecting terminal vertices $R = \{a, e\}$ is neither a tree nor a tree with a C_4. Note that this is not a solution to ML, as for example there is no temporal path from c to a or to g. Now, we can remove the labels from the edges bg, gf, fe and add them (in the same order) to the edges bc, cd, de, respectively. This way, we obtain an optimal solution, where the subgraph which has labeled edges is a tree (in this case even a path).

and $P' = R'_f$. It follows that we must have $F_e <_{\text{label}} F'_e <_{\text{label}} R'_f <_{\text{label}} R_f$ whenever the end vertex u_v of F_e is different from the start vertex $u_{v'}$ of R_f. Next, we categorize the short forwarding and short returning paths by their start and end vertices, respectively. Define $\mathcal{F}'_v = \{F'_e \mid F'_e \text{ starts at } u_v\}$ and $\mathcal{R}'_v = \{R'_e \mid R'_e \text{ ends at } u_v\}$. From what we proved above it follows that for any $P \in \mathcal{F}'_v$ and $Q \in \mathcal{R}'_{v'}$, where $v \neq v'$, we must have $P <_{\text{label}} Q$.

Assume now for contradiction that $|S| > k$ which means that $|U_V^+| > k$ and $|U_V^-| > k$. We analyze the case where for all $u_v \in U_V^-$ we have $|\mathcal{F}'_v| = 1$ and for all $u_v \in U_V^+$ we have $|\mathcal{R}'_v| = 1$ and show that already this case yields a contradiction. From here on we denote $\mathcal{F}'_v = \{F'_v\}$ and $\mathcal{R}'_v = \{R'_v\}$. Similarly, as in arguments we made before, we have that for at most one $v \in V$ we can have that $R'_v <_{\text{label}} F'_v$ or R'_v and F'_v are incomparable with respect to $<_{\text{label}}$. Hence, at most one pair of paths, R'_v and F'_v can share a time edge. Since $|U_V^+| > k$ and $|U_V^-| > k$ implies that there are at least $2k + 2$ paths, we have that $2k$ paths need three labels each and at most one pair of paths needs five labels in total. However, this yields a number of at least $6k + 5$ labels, which is more than the $6k + 2$ labels available for these paths. Hence, the assumption that $|S| > k$ leads to a contradiction, which proves that S really is a vertex cover of size at most k. □

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 T of the input graph G (where every leaf is a terminal vertex), and potentially one further edge that forms a C_4 with three edges of the subtree T. Here, recall a subtree T of G is a subgraph of G that is also a tree.

Remark 2. *Recall that in the case of* ML *(Theorem 5 and Bumby [9]), we also have that there exist optimal solutions that label a spanning tree, and potentially one further edge that forms a C_4 with the edges of the spanning tree. Note however that, in the case of* MSL*,* *we have a weaker requirement on labelings, namely that* only terminal *vertices need to be temporally connected, instead of* all vertices *as in the case of* ML. *Therefore, in* MSL *we have an additional difficulty: can the abundance of non-terminal vertices (i.e., of vertices that do not need to be temporally connected) lead to a solution for an* MSL *instance that is neither a tree nor a tree with a C_4, but still has fewer labels than any solution that is either a tree or a tree with a C_4? As we prove in Lemma 15, this cannot happen, i.e., also in the case of* MSL *it suffices to search for solutions that have this special topological structure. To do so, we specify how an arbitrary optimal solution for* MSL *(see the example of Figure 8 for an illustration) can be transformed into another optimal solution that is a tree or a tree with a C_4.*

Intuitively speaking, this insight allows us to 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 15. *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 15 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.

Proof of Lemma 15. Assume there is a labeling λ for G that labels all edges in the subgraph H of G with k labels such that all vertices in R are pairwise temporally connected. 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 leaves 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 sublabeling 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 leaves 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 labeling λ' of size k for G such that there are two trees $H_{r^*}^+, H_{r^*}^-$ such that $H_{r^*}^+$ is 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 $(H_{r^*}^+, \lambda_{r^*}^+)$ and $H_{r^*}^-$ is a minimum subgraph of H and $\lambda_{r^*}^-$ a minimum sublabeling of λ' for $H_{r^*}^-$ such that each vertex in $R \setminus \{r^*\}$ can temporally reach r^* in $(H_{r^*}^-, \lambda_{r^*}^-)$, 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

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

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 5). 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 5). 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 $H^+_{r^*}, H^-_{r^*}$ such that $H^+_{r^*}$ is 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 $(H^+_{r^*}, \lambda^+_{r^*})$ and $H^-_{r^*}$ is a minimum subgraph of H and $\lambda^-_{r^*}$ a minimum sublabeling of λ' for $H^-_{r^*}$ such that each vertex in $R \setminus \{r^*\}$ can temporally reach r^* in $(H^-_{r^*}, \lambda^-_{r^*})$, 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 such that $H^+_{r^*}$ 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 $(H^+_{r^*}, \lambda^+_{r^*})$ and $H^-_{r^*}$ be a minimum subgraph of H and $\lambda^-_{r^*}$ a minimum sublabeling of λ' for $H^-_{r^*}$ such that each vertex in $R \setminus \{r^*\}$ can temporally reach r^* in $(H^-_{r^*}, \lambda^-_{r^*})$. Furthermore, let $|E(H^+_{r^*})| + |E(H^-_{r^*})| - |E^*_{r^*}| = k-x$ 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 leaves 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 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.

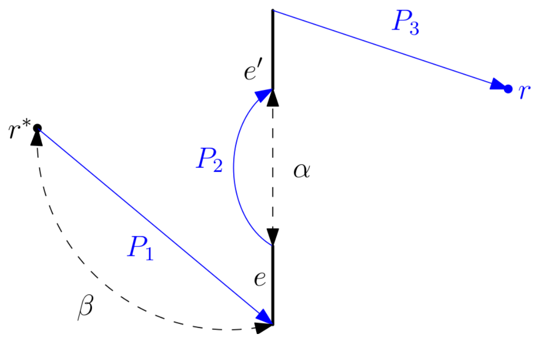
We partition the temporal path P into the part P_1 from r^* to e, the part consisting of e itself, the part P_2 between e and e', the part consisting of e' itself, and the part P_3 from e' to r. Now in $H^-_{r^*}$ we can have two different scenarios. For illustrations of all variations of Case A see Figures 9 to 13.

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'.

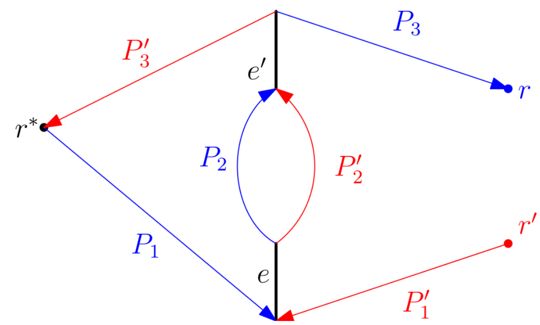
We partition the temporal path P' into the part P'_1 from r' to e, the part consisting of e itself, the part P'_2 between e and e', the part consisting of e' itself, and the part P'_3 from e' to r^*.

The analysis of each one follows from the observation that the labels in P'_3 are larger than the ones in P_1.

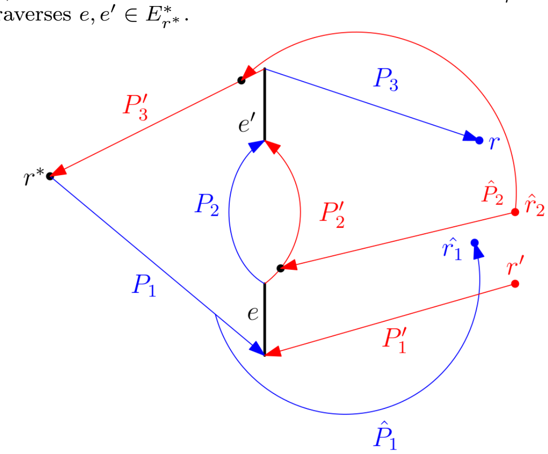
Case A-1-i. Assume there is a path $\hat{P}_1$ in $H^+_{r^*}$ starting at a vertex that is visited by P_1 and ending at $\hat{r}_1 \in R \setminus \{r^*\}$ such that $\hat{r}_1 = r'$ or $\hat{P}_1$ and P'_1 intersect in a vertex. For our analysis, we treat these two cases the same since in both cases 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



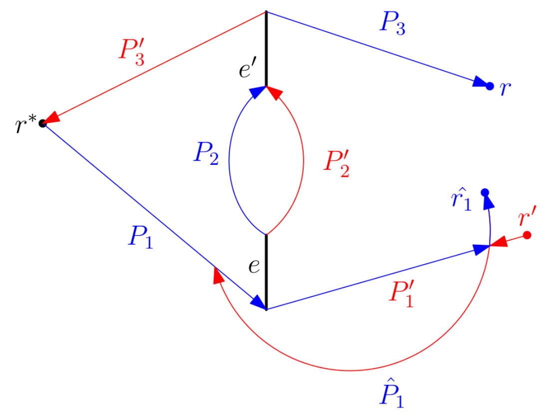
(a) Case A: an example of a path P from r^* in $H^+_{r^*}$, that traverses $e, e' \in E^*_{r^*}$.



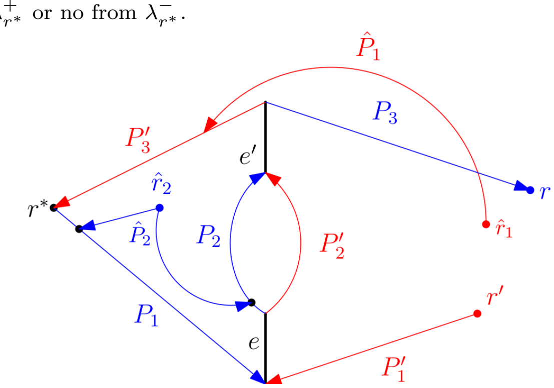
(b) Case A-1: an example of P in $H^+_{r^*}$ and P' in $H^-_{r^*}$, that share $e, e' \in E^*_{r^*}$.



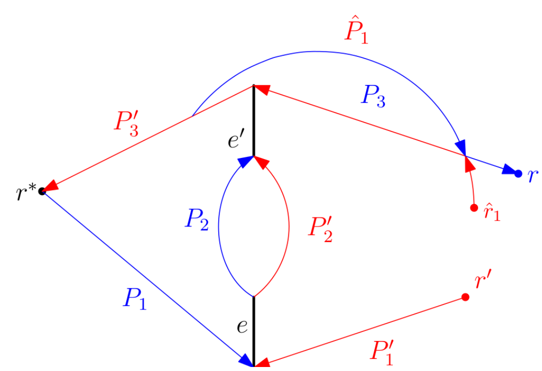
(c) Case A-1-i: P^* from $\hat{r}_2$ to $\hat{r}_1$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



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

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 $\lambda^-_{r^*}$.
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.

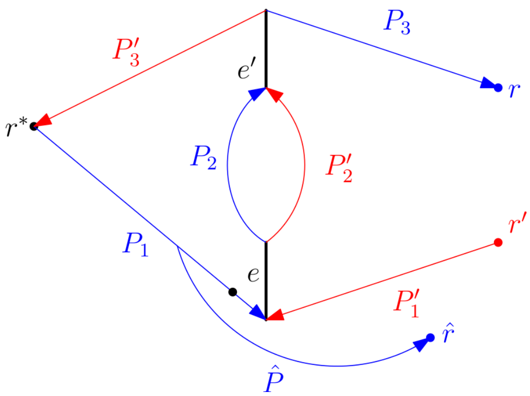
- We remove P_1's edges and labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively, add P_1's edges to $H^-_{r^*}$, and add new labels for the edges of P_1 to $\lambda^-_{r^*}$ such that there are temporal paths from both endpoints of e to r^* that only use the new labels.

- 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.

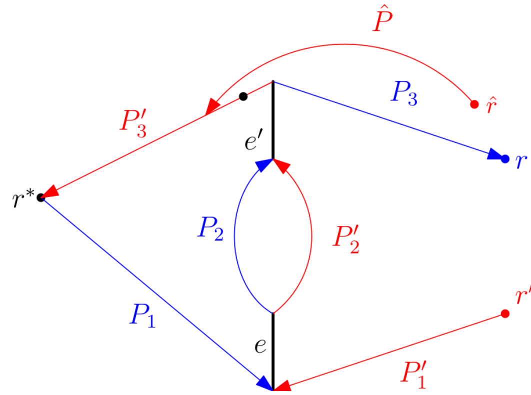
In both modification above, we assume w.l.o.g. that the smallest and the largest label assigned to an edge of P_1 by $\lambda^+_{r^*}$ before the modification are equal the smallest and the largest label, respectively, assigned to an edge of P'_3 by $\lambda^+_{r^*}$ after the modification. Symmetrically, 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.

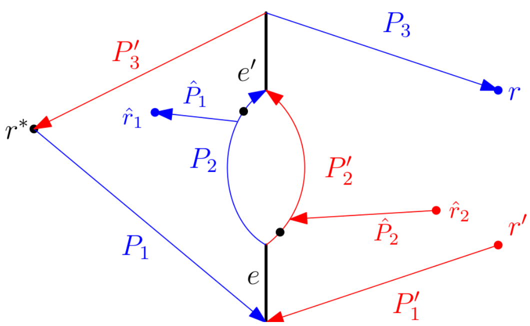
- If Case A-1-vi applies, then we remove e, e', and the edges of P'_2 and their labels from $H^-_{r^*}$ and $\lambda^-_{r^*}$, respectively. Furthermore, we exchange the labels of e and e' and the



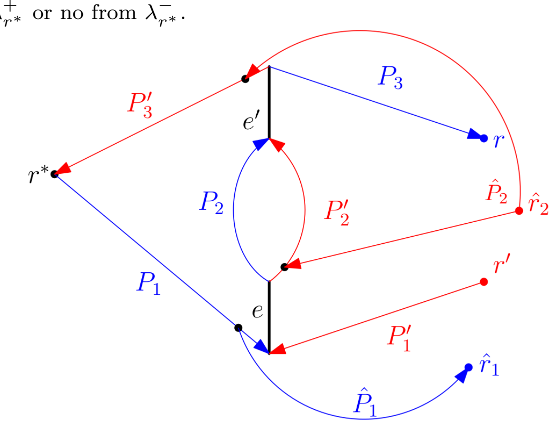
(a) Case A-1-iii: P^* from r' to $\hat{r}$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



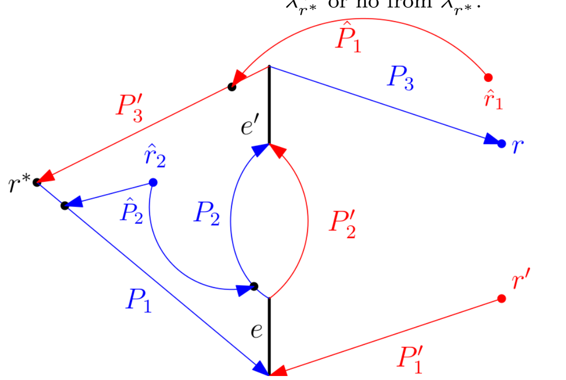
(b) Case A-1-iv: P^* from $\hat{r}$ to r either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.



(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 $\lambda^-_{r^*}$.

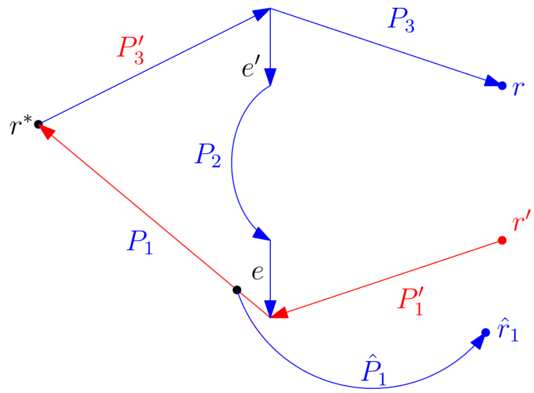


(e) Case A-1-vii: P^* from $\hat{r}_1$ to $\hat{r}_2$ either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

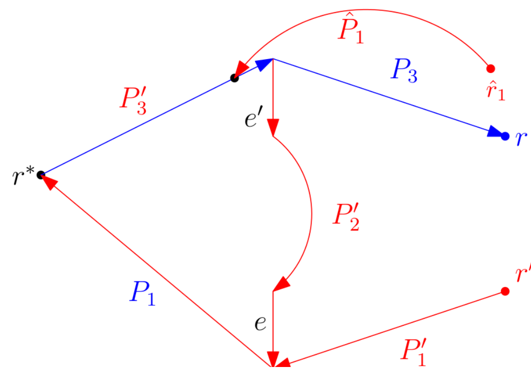
Figure 10: Cases A-1-iii – A-1-vii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

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 $(H^+_{r^*}, \lambda^+_{r^*})$.

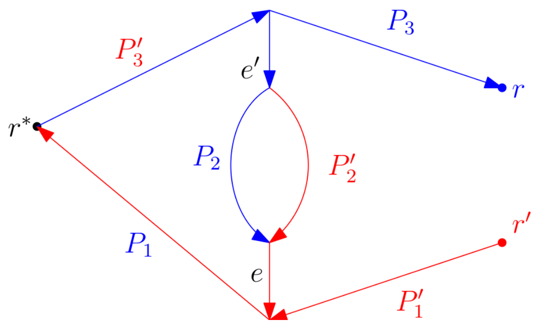
- If Case A-1-vii applies, then we remove e, e', and the edges of P_2 and their labels from $H^+_{r^*}$ and $\lambda^+_{r^*}$, respectively. Furthermore, we exchange the labels of e and e' and the



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



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



(c) Modification when none of the cases A-1-vi or A-1-vii apply.

Figure 11: Modifications for cases A-1-vi – A-1-vii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

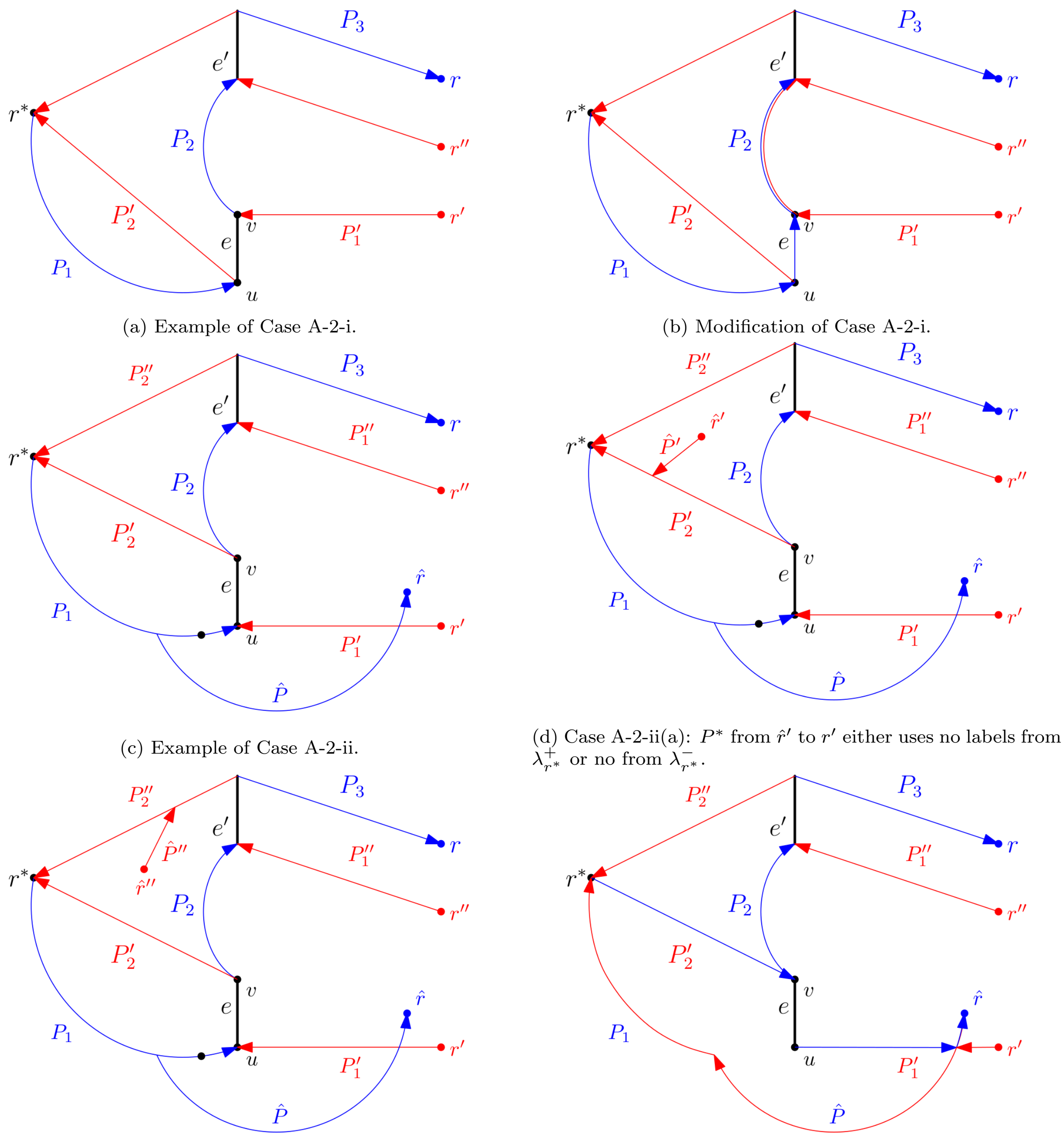
edges of P_2' assigned by $\lambda^-_{r^*}$ in a way that there is a temporal path from $\hat{r}_1$ (see Case A-1-vii) to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$.

- 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



(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 $\lambda^-_{r^*}$.

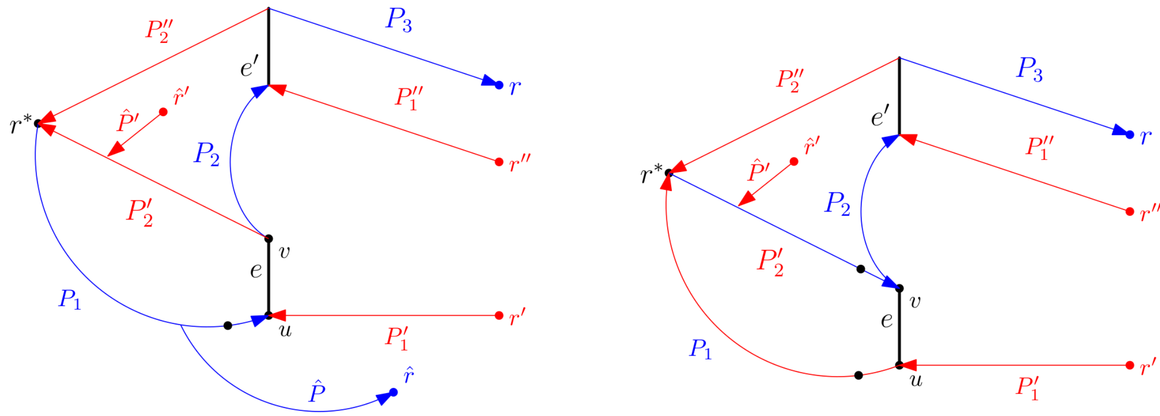
(e) Case A-2-ii(b): P^* from $\hat{r}''$ to r' either uses no labels from $\lambda^+_{r^*}$ or no from $\lambda^-_{r^*}$.

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

Figure 12: Cases A-2-i – A-2-ii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

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.



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

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

Figure 13: Case A-2-iii, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

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 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.

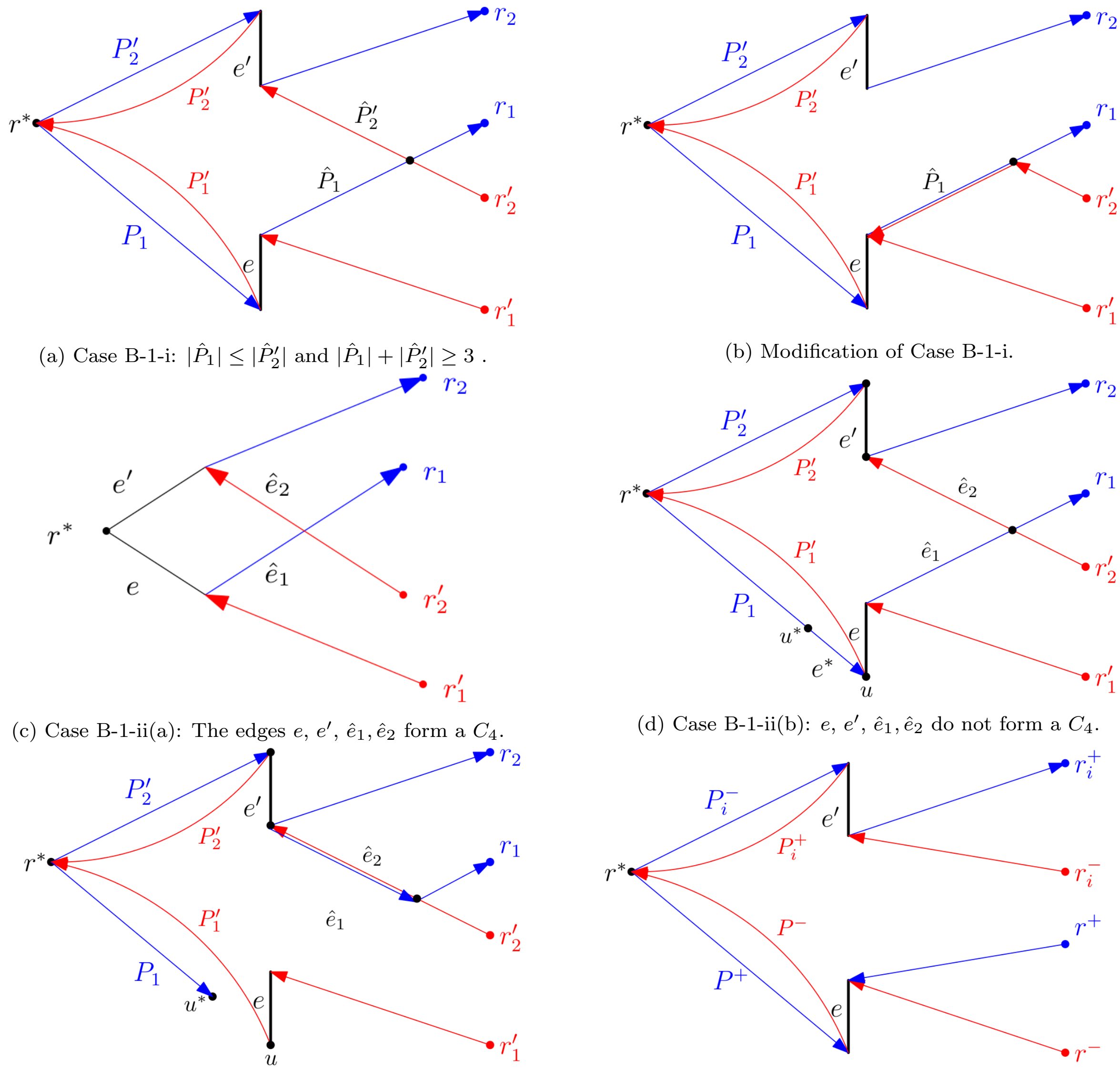
Note that now there is a path from $\hat{r}'$ to r^* in $(H^-_{r^*}, \lambda^-_{r^*})$ that does not use edge e. Further note that after the modifications $H^+_{r^*}$ and $H^-_{r^*}$ are still trees, and $\lambda^+_{r^*}$ and $\lambda^-_{r^*}$ still assign at most one label per edge. Furthermore, we 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 14.

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



(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 14: Cases B-1 – B-2, where *blue* color corresponds to the labeling $\lambda^+_{r^*}$ and *red* to $\lambda^-_{r^*}$.

$r'_1 \in R \setminus \{r^*\}$ 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 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 15, 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 15 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 16. MSL *is in FPT when parameterized by the number* $|R|$ *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.

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 a 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 that our algorithm claims that (G, R, k) is a YES instance of MSL. We prove that in this case (G, R, k) is indeed a YES instance of MSL.

First, consider the case where there exists a C_4 in G such that $k_{C_4} = k^*$. Then there exists a subtree of G (a not necessarily induced subgraph of G that is a tree) 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 5. This requires $2k^* - 2$ labels and clearly afterwards all terminals can pairwise reach each other. Hence, we have that if $k \geq 2k^* - 2$, and then (G, R, k) is a YES instance of MSL.

Next, consider the case where 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 5. This requires $2k^* - 1$ labels and clearly afterwards all terminals can pairwise reach each other. Hence, we have that if $k \geq 2k^* - 1$, and then (G, R, k) is a YES instance of MSL.

($\Rightarrow$): Assume that (G, R, k) is a YES instance of MSL. We prove that our algorithm correctly determines that (G, R, k) is a YES instance of MSL.

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 15, 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 leaves of H are terminals, we have that every vertex in (H, λ) can temporally reach every other vertex. By Theorem 5 we know that then λ needs $2k^* - 1$ labels. This implies that $k \geq k_{\text{opt}} \geq 2k^* - 1$ and the algorithm correctly determines that (G, R, k) is a YES instance.

Now assume that k_{opt} is even. Then by Lemma 15 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 5 that $k_{\text{opt}} < 2k^* - 2$. However, since all leaves of H' are terminals, we have that every vertex in (H, λ) can temporally reach every other vertex. Hence, Theorem 5 implies that $k_{\text{opt}} \geq 2k^* - 2$. It follows that $k_{\text{opt}} < 2k^* - 2$ leads to a contradiction and we have $k \geq k_{\text{opt}} \geq 2k^* - 2$ and the algorithm correctly determines that (G, R, k) is a YES instance.

Running time: We can use the FPT-algorithm for Steiner Tree parameterized by the number of terminals by Dreyfus and Wagner [14] for algorithm $\mathcal{A}$, by trying out values for k^* from 1 until we find that (G, R, k^*) is a YES-instance of Steiner Tree or $k < 2k^* - 1$. Note that this increases the running time of the algorithm by Dreyfus and Wagner [14] only by a linear factor. Furthermore, 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 13 and 14. In this section, we prove that MASL is W[1]-hard when parameterized by the number k of labels, even if the restriction a on the age is a constant. Note that the number of terminals can be upper-bounded by a function of the number of labels, since by Theorem 5 we know that to temporally connect $|R|$ at least $2|R| - 4$ labels are necessary. Hence, our results also implies that MASL is W[1]-hard when parameterized by the number $|R|$ of terminals, even if the restriction a on the age is a constant.

To show our parameterized hardness result, we provide a parameterized reduction from Multicolored Clique. This, together with Theorem 16, implies that MASL is strictly harder than MSL (parameterized by the number $|R|$ of terminals), unless FPT=W[1].

Theorem 17. MASL *is W[1]-hard when parameterized by the number k of labels, 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 number of labels, even if the restriction 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 [20].

Multicolored Clique

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

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 (for an illustration see Figure 15). The vertex set $V(G^*)$ consists of the following vertices:

- a "color-vertex" corresponding to every color of $V(G)$: $C = \{c_i \mid i \in \{1, 2, \dots, k\}\}$,
- a "vertex-vertex" corresponding to every vertex of G: $U_V = \{u_v \mid v \in V(G)\}$,
- an "edge-vertex" corresponding to every edge of G: $U_E = \{u_e \mid e \in E(G)\}$,
- a "color-combination-vertex" corresponding to a pair of two colors of $V(G)$: $W = \{c_{i,j} \mid i, j \in \{1, 2, \dots, k\}, i < j\}$, 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 $c_{j,k}$, 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.

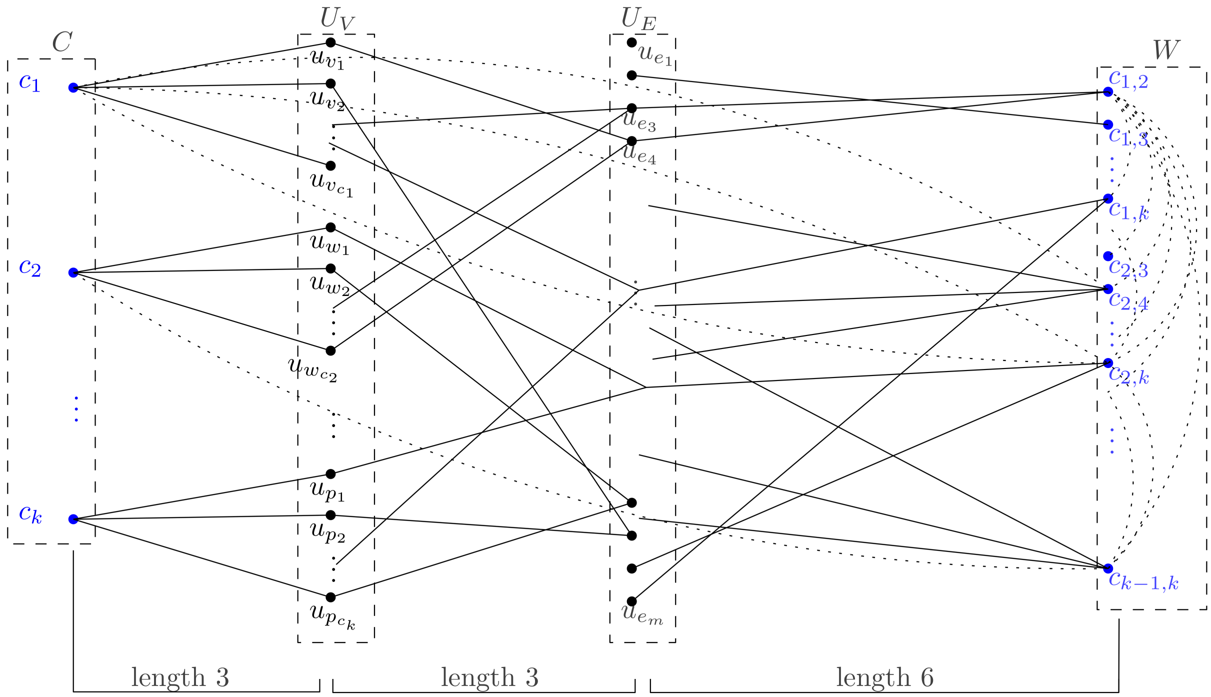


Figure 15: Illustration of the MASL instance produced by the reduction presented in the proof of Theorem 17. For better readability, some paths among the vertices in W and paths among $c_i \in C$ and $c_{j,k} \in W$ $(i \neq j \neq k)$, are not depicted.

We set $R^* = C \cup W$, $a^* = 12$ and $k^* = 6k + 6(k^2 - k) + 6(k^2 - k) + 3(k^4 - 2k^3 - k^2 + 2k) + 12(k^3 - 3k^2 + 2k)$. Note that $k^* \in O(k^4)$, hence the parameter number of labels of the MASL instance is upper-bounded by a function of k. Furthermore, observe that the restriction on the age is a constant. This finishes the construction. It is not hard to see that this construction can be performed in polynomial time. 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 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 that 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 $c(v) = i$, 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 $c_{i,j}$ 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 is equal to $3(k^4 - 2k^3 - k^2 + 2k)$.

Finally, let $c_{i'}$ and $c_{i,j}$ 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 $12(k^3 - 3k^2 + 2k)$. In the case when $i' \in \{i, j\}$ (w.l.o.g. $i' = i$) we connect the vertices using the following path. In S there exists a unique vertex of color i, denote it v. By the definition of S there is also vertex $w \in S$ of color j, which is connected to v with some edge, denote it e. Therefore, to obtain a $(c_i, c_{i,j})$-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 $(c_{i,j}, c_i)$-temporal path uses the same edges, 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-combination-vertex c_{ij} and reused all labels on the (c_i, u_e)-temporal paths. Repeating this for every color-combination-vertex we use $2 \cdot 6|W|$ new labels, since $|W| = \binom{k}{2}$ this is equal 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 on how to reach c_j from c_i. One when the path connecting them passes through the

set U_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-combination-vertex $c_{i,j}$. Since the path between any color-vertex and color-combination-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 $c_{i,j}$ and $c_{i'}$ be two arbitrary vertices from the color-combination-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 $(c_{i'}, c_{i,j})$-path must either go from $c_{i'}$ to some color-combination-vertex $c_{i',j'}$, which is then connected with a path of length 12 to the $c_{i,j}$, or go to one of the color-vertices and then continue to the $c_{i,j}$. 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 $vw = e \in E(G)$ be such that $c(w) = j$. Then there is a path from c_i to $c_{i,j}$ 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-combination-vertex $c_{i,j}$ (using a path of length 6). Hence the constructed $(c_i, c_{i,j})$-path is of length 12. There exists also another $(c_i, c_{i,j})$-path, that goes through some other $c_{i,j'}$ color-combination-vertex, but it is longer than 12.

Case C. Let $c_{i,j}$ and $c_{i',j'}$ be two arbitrary color-combination-vertices. By construction of G^*, there is a path of length 12 connecting them. Any other $(c_{i,j}, c_{i',j'})$-path, must use at least one vertex-vertex, which is at distance 9 from the color-combination-vertices (therefore the path through it would be of length at least 18), or a color-vertex, which is at distance 12 from the color-combination-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 each pair of terminal vertices must use all labels $1, 2, \ldots, 12$ on the temporal paths between any two color-combination-vertices $c_{i,j}$ and $c_{i',j'}$, and between a color-vertex $c_{i'}$ and a color-combination-vertex $c_{i,j}$, where $i' \notin \{i, j\}$. This is true as by construction there are unique paths of length 12 connecting each pair of them. For these temporal paths we must use $2 \cdot 12\frac{|W|(|W|-1)}{2}$ labels (since $|W| = \binom{k}{2}$ this is equal to $3(k^4 - 2k^3 - k^2 + 2k)$) and $2 \cdot 12k\binom{k-1}{2}$ labels (which equals $12(k^3 - 3k^2 + 2k)$). 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 (where v is one of the endpoints of 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 $7, 8, \ldots, 12$, or to the vertex-vertex $u_{v'}$ corresponding to the other endpoint of edge e (the edge e is between v and v'). This finishes the construction of the temporal path from a color-vertex to the color-combination-vertex and the temporal paths among color-vertices. It remains to connect a color-combination-vertex with its corresponding color-vertices. The temporal path must go through some edge-vertex u_e, that is at distance 6 from it, therefore the labeling must use the labels $1, 2, \ldots, 6$. 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 $6(k^2 - k)$ labels) on the edges between the color-combination-vertices in W and the edge-vertices in U_E and at least $2 \cdot 6\binom{k}{2}$ labels (which equals $6(k^2 - k)$ labels) on the edges between the edge-vertices in U_E and vertex-vertices in U_V. Since all this together equals 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 argument 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 (possibly the same) 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 c_j. To obtain the $(c_i, c_{i,j})$-temporal path we must label the edges from u_e to $c_{i,j}$ with the labels $6, 7, \ldots, 12$, since the edge-vertex u_e is the only edge-vertex connected to the color-combination-vertex $c_{i,j}$ 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-combination-vertex $c_{i,j}$, it must use the same labels between u_e and $c_{i,j}$ (using the same reasoning as before). Therefore the path from c_j to u_e (through) u_w uses also the labels $1, 2, \ldots, 6$. 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. This implies that we must have that $u_v = u_{v'}$. 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 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. □

5. Concluding remarks

In this paper we studied four natural temporal labeling problems. We distinguished the settings where we have an age restriction on the labeling or not. Furthermore, we investigated settings where the labeling has to temporally connect *every* vertex pair and settings where only a given set of *terminal* vertices have to be pairwise temporally connected. One variant (ML) is polynomial-time solvable, whereas the tree other variants (MAL, MSL, and MASL) turn out the be NP-hard. For the latter two we also give parameterized complexity results with respect to the number of labels and to the number of terminals as parameters. Our work spawns several future research directions.

Recall that a labeling λ satisfying MAL with the age restriction being the diameter of the graph, is of the size $O(n^2)$ (see Observation 1). In Lemma 2, we show that on cycles, the labeling uses $\Theta(n^2)$ labels. Therefore, it would be interesting to study, for which graph classes the optimal labeling uses $o(n^2)$ or $O(n)$ labels. We show that MAL is NP-complete when the upper age bound is equal to the diameter d of the input graph G. On the other hand, if the upper age bound is $2r$, where r is the radius of G, MAL can be computed in polynomial time. Indeed, using the results of Section 2.1, it easily follows that if G contains (or does not contain) a C_4, then the labeling consists of $2n-4$ (or $2n-3$) labels. An interesting question that arises now is: For which values of an upper age bound a, where $d \leq a \leq 2r$, can MAL be solved efficiently? Furthermore, it would be interesting to analyse the parameterized complexity of MAL. A canonical starting point would be to consider the number of time labels as a parameter.

Our results for MSL and MASL also leave some open questions and several natural future research directions. Recall that the number k of labels is a larger parameter than the number of terminals. Hence, the parameterized complexity with respect to those two parameters of MSL is resolved. For MASL it remains open whether we can obtain an XP algorithm for those parameters.

More generally, it would be interesting to investigate structural parameterizations for all NP-hard problem variants of this paper. We conjecture that all problem variants are polynomial-time solvable if the input graph G is a tree. Consequently, parameters that measure tree-likeness, such as treewidth, are promising candidates for obtaining FPT results.

References

[1] E. C. Akrida, L. Gasieniec, G. B. Mertzios, and P. 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] E. C. Akrida, L. Gasieniec, G. B. Mertzios, and P. G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017.

[3] E. C. Akrida, G. B. Mertzios, S. E. Nikoletseas, C. L. Raptopoulos, P. G. Spirakis, and V. Zamaraev. How fast can we reach a target vertex in stochastic temporal graphs? *Journal on Computer and System Sciences*, 114:65–83, 2020.

[4] E. C. Akrida, G. B. Mertzios, P. G. Spirakis, and V. Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020.

[5] P. Alimonti and V. 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] K. Axiotis and D. 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] M. Bentert, A.-S. Himmel, H. Molter, M. Morik, R. Niedermeier, and R. Saitenmacher. Listing all maximal k-plexes in temporal graphs. *ACM Journal of Experimental Algorithmics*, 24(1):13:1–13:27, 2019.

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

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

[10] S. Buß, H. Molter, R. Niedermeier, and M. 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] A. Casteigts, J. G. Peters, and J. Schoeters. Temporal cliques admit sparse spanners. *Journal of Computer and System Sciences*, 121:1–17, 2021.

[12] A. Deligkas, E. Eiben, and G. Skretas. Minimizing reachability times on temporal graphs via shifting labels. *CoRR*, abs/2112.08797, 2021.

[13] A. Deligkas and I. Potapov. Optimizing reachability sets in temporal graphs by delaying. *Information and Computation*, 285(Part B):104890, 2022.

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

[15] J. Enright, K. Meeks, G. B. Mertzios, and V. Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. *Journal of Computer and System Sciences*, 119:60–77, 2021.

[16] J. Enright, K. Meeks, and F. Skerman. Assigning times to minimise reachability in temporal graphs. *Journal of Computer and System Sciences*, 115:169–186, 2021.

[17] J. A. Enright, K. Meeks, and H. Molter. Counting temporal paths. In *Proceedings of the 40th International Symposium on Theoretical Aspects of Computer Science (STACS '23)*, volume 254 of *LIPIcs*, pages 30:1–30:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[18] T. Erlebach, M. Hoffmann, and F. Kammer. On temporal graph exploration. *Journal of Computer and System Sciences*, 119:1–18, 2021.

[19] T. Erlebach and J. 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.

[20] M. R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

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

[22] R. Haag, H. Molter, R. Niedermeier, and M. Renken. Feedback edge sets in temporal graphs. *Discrete Applied Mathematics*, 307:65–78, 2022.

[23] T. Hamm, N. Klobas, G. B. Mertzios, and P. G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Proceedings of the 36th Conference on Artificial Intelligence (AAAI)*, pages 10193–10201, 2022.

[24] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[25] P. Holme and J. Saramäki. *Temporal network theory*, volume 2. Springer, 2019.

[26] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[27] D. Kempe, J. M. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

[28] N. Klobas, G. B. Mertzios, H. Molter, R. Niedermeier, and P. Zschoche. Interference-free walks in time: temporally disjoint paths. *Autonomus Agents Multi Agent Systems*, 37(1):1, 2023.

[29] N. Klobas, G. B. Mertzios, H. Molter, and P. G. Spirakis. The complexity of computing optimum labelings for temporal connectivity. In *Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, page 62:1–62:15, 2022.

[30] N. Klobas, G. B. Mertzios, H. Molter, and P. G. Spirakis. Realizing temporal graphs from fastest travel times. *CoRR*, abs/2302.08860, 2023.

[31] P. Kunz, H. Molter, and M. Zehavi. In which graph structures can we efficiently find temporally disjoint paths and walks? In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 180–188, 2023.

[32] A. Marino and A. Silva. Coloring temporal graphs. *Journal of Computer and System Sciences*, 123:171–185, 2022.

[33] G. B. Mertzios, O. Michail, and P. G. Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019.

[34] G. B. Mertzios, H. Molter, R. Niedermeier, V. Zamaraev, and P. Zschoche. Computing maximum matchings in temporal graphs. *Journal of Computer and System Sciences*, 137:1–19, 2023.

[35] G. B. Mertzios, H. Molter, M. Renken, P. G. Spirakis, and P. Zschoche. The complexity of transitively orienting temporal graphs. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 202, pages 75:1–75:18, 2021.

[36] G. B. Mertzios, H. Molter, and V. Zamaraev. Sliding window temporal graph coloring. *Journal of Computer and System Sciences*, 120:97–115, 2021.

[37] O. Michail and P. G. Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016.

[38] O. Michail and P. G. Spirakis. Elements of the theory of dynamic networks. *Communications of the ACM*, 61(2):72–72, Jan. 2018.

[39] H. Molter, M. Renken, and P. Zschoche. Temporal reachability minimization: Delaying vs. deleting. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 76:1–76:15, 2021.

[40] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora. Graph metrics for temporal networks. In *Temporal Networks*. Springer, 2013.

[41] S. Thejaswi, J. Lauri, and A. Gionis. Restless reachability in temporal graphs. *CoRR*, abs/2010.08423, 2021.

[42] T. Viard, M. Latapy, and C. Magnien. Computing maximal cliques in link streams. *Theoretical Computer Science*, 609:245–252, 2016.