

Temporal graph realization from fastest paths^{*}

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, Israel*

^c*Department of Computer Science, University of Liverpool, United Kingdom*

Abstract

In this paper, we initiate the study of the *temporal graph realization* problem with respect to the fastest path durations among its vertices, while we focus on periodic temporal graphs. Given an $n \times n$ matrix D and a $\Delta \in \mathbb{N}$, the goal is to construct a Δ -periodic temporal graph with n vertices such that the duration of a *fastest path* from v_i to v_j is equal to $D_{i,j}$, or to decide that such a temporal graph does not exist. The variations of the problem on static graphs have been well studied and understood since the 1960s (e.g. [Erdős and Gallai, 1960], [Hakimi and Yau, 1965]). As it turns out, the periodic temporal graph realization problem has a very different computational complexity behavior than its static (i.e., non-temporal) counterpart.

First, we show that the problem is NP-hard in general, but polynomial-time solvable if the so-called underlying graph is a tree. Building upon those results, we investigate its parameterized computational complexity with respect to structural parameters of the underlying static graph which measure the “tree-likeness”. We prove a tight classification between such parameters that allow fixed-parameter tractability (FPT) and those which imply W[1]-hardness. We show that our problem is W[1]-hard when parameterized by the *feedback vertex number* (and therefore also any smaller parameter such as *treewidth*, *degeneracy*, and *cliquewidth*) of the underlying graph, while we show that it is in FPT when parameterized by the *feedback edge number* (and therefore also any larger parameter such as *maximum leaf number*) of the underlying graph.

Keywords: temporal graph, periodic temporal labeling, fastest temporal path, graph realization, temporal connectivity, parameterized complexity.

1. Introduction

The (static) *graph realization* problem with respect to a graph property \mathcal{P} is to find a graph that satisfies property \mathcal{P} , or to decide that no such graph exists. The motivation for graph realization problems stems both from “verification” and from network design applications in engineering. In *verification* applications, given the outcomes of some experimental measurements (resp. some computations) on a network, the aim is to (re)construct an input network that complies with them. If such a reconstruction is not possible, this proves that the measurements are incorrect or implausible (resp. that the algorithm that made the computations is incorrectly implemented). One example of a graph realization (or reconstruction) problem is the

^{*}An extended abstract of this work appeared in the proceedings of the 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND ’24, *Best Student Paper*) [49].

George B. Mertzios was supported by the EPSRC grant EP/P020372/1. Hendrik Molter was supported by the ISF, grant No. 1456/18, and the European Union’s Horizon Europe research and innovation programme under grant agreement 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)

recognition of probe interval graphs, in the context of the physical mapping of DNA, see [55, 54] and [39, Chapter 4]. In *network design* applications, the goal is to design network topologies having a desired property [4, 41]. Analyzing the computational complexity of the graph realization problems for various natural and fundamental graph properties \mathcal{P} requires a deep understanding of these properties. Among the most studied parameters for graph realization are constraints on the distances between vertices [8, 7, 44, 16, 10, 17], on the vertex degrees [38, 40, 43, 6, 24], on the eccentricities [5, 45, 9, 53], and on connectivity [37, 32, 15, 33, 34, 40], among others.

In the simplest version of a (static) graph realization problem with respect to vertex distances, we are given a symmetric $n \times n$ matrix D and we are looking for an n -vertex undirected and unweighted graph G such that $D_{i,j}$ equals the distance between vertices v_i and v_j in G . This problem can be trivially solved in polynomial time in two steps [44]: First, we build the graph $G = (V, E)$ such that $v_i v_j \in E$ if and only if $D_{i,j} = 1$. Second, from this graph G we compute the matrix D_G which captures the shortest distances for all pairs of vertices. If $D_G = D$ then G is the desired graph, otherwise there is no graph having D as its distance matrix. Non-trivial variations of this problem have been extensively studied, such as for weighted graphs [44, 63], as well as for cases where the realizing graph has to belong to a specific graph family [44, 7]. Other variations of the problem include the cases where every entry of the input matrix D may contain a range of consecutive permissible values [7, 64, 67], or even an arbitrary set of acceptable values [8] for the distance between the corresponding two vertices.

In this paper we make the first attempt to understand the complexity of the graph realization problem with respect to vertex distances in the context of *temporal graphs*, i. e., of graphs whose *topology changes over time*.

Definition 1 (temporal graph [46]). *A temporal graph is a pair (G, λ) , where $G = (V, E)$ is an underlying (static) graph and $\lambda : E \rightarrow 2^{\mathbb{N}}$ is a time-labeling function which assigns to every edge of G a set of discrete time-labels.*

Here, whenever $t \in \lambda(e)$, we say that the edge e is *active* or *available* at time t . In the context of temporal graphs, where the notion of vertex adjacency is time-dependent, the notions of path and distance also need to be redefined. The most natural temporal analog of a path is that of a *temporal* (or *time-dependent*) 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.

Definition 2 (fastest temporal path). *Let (G, λ) be a temporal graph. A temporal path in (G, λ) is a sequence $(e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$, where $P = (e_1, \dots, e_k)$ is a path in the underlying static graph G , $t_i \in \lambda(e_i)$ for every $i = 1, \dots, k$, and $t_1 < t_2 < \dots < t_k$. The duration of this temporal path is $t_k - t_1 + 1$. A fastest temporal path from a vertex u to a vertex v in (G, λ) is a temporal path from u to v with the smallest duration. The duration of the fastest temporal path from u to v is denoted by $d(u, v)$.*

In this paper, we consider *periodic* temporal graphs, i. e., temporal graphs in which the temporal availability of each edge of the underlying graph is periodic. Many natural and technological systems exhibit a periodic temporal behavior. For example, in railway networks, an edge is present at a time step t if and only if a train is scheduled to run on the respective rail segment at time t [3]. Similarly, a satellite, which makes pre-determined periodic movements, can establish a communication link (i. e., a temporal edge) with another satellite whenever they are sufficiently close to each other; the existence of these communication links is also periodic. In a railway (resp. satellite) network, a fastest temporal path from u to v represents a fastest railway connection between two stations (resp. the quickest communication delay between two moving satellites). Furthermore, periodicity appears also in (the otherwise quite complex) social

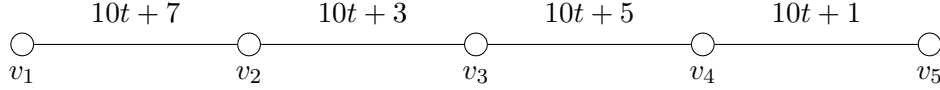


Figure 1: An example of a Δ -periodic temporal graph (G, λ, Δ) , where $\Delta = 10$ and the 10-periodic labeling $\lambda : E \rightarrow \{1, 2, \dots, 10\}$ is as follows: $\lambda(v_1v_2) = 7$, $\lambda(v_2v_3) = 3$, $\lambda(v_3v_4) = 5$, and $\lambda(v_4v_5) = 1$. Here, a fastest temporal path from v_1 to v_5 traverses the first edge v_1v_2 at time 7, second edge v_2v_3 at time 13, third edge v_3v_4 at time 15 and the last edge v_4v_5 at time 21. This results in the total duration of $21 - 7 + 1 = 15$ for a fastest temporal path from v_1 to v_5 .

networks which describe the dynamics of people meeting [52, 65], as every person individually follows mostly a weekly routine.

Expanding the work on periodic temporal graphs that has already been studied (see [13, Class 8] and [3, 27, 62, 61]), our study represents the first attempt to understand the complexity of a graph realization problem in the context of temporal graphs. Therefore, we focus in this paper on the most fundamental case, where all edges have the same period Δ (while in the more general case, each edge e in the underlying graph has a period Δ_e). As it turns out, the periodic temporal graph realization problem with respect to a given $n \times n$ matrix D of the fastest duration times has a very different computational complexity behavior than the classic graph realization problem with respect to shortest path distances in static graphs.

Formally, let $G = (V, E)$ and $\Delta \in \mathbb{N}$, and let $\lambda : E \rightarrow \{1, \dots, \Delta\}$ be an edge-labeling function that assigns to every edge of G exactly one of the labels from $\{1, \dots, \Delta\}$. Then we denote by (G, λ, Δ) the Δ -periodic temporal graph (G, L) , where for every edge $e \in E$ we have $L(e) = \{i\Delta + x : i \geq 0, x \in \lambda(e)\}$. In this case, we call λ a Δ -periodic labeling of G ; see Fig. 1 for an illustration. When it is clear from the context, we drop Δ from the notation and we denote the $(\Delta$ -periodic) temporal graph by (G, λ) . Given a duration matrix D , it is easy to observe that, similarly to the static case, if $D_{i,j} = 1$ then v_i and v_j must be connected by an edge. We call the graph defined by these edges the *underlying graph* of D .

Our contribution. We initiate the study of naturally motivated graph realization problems in the temporal setting. Our target is not to model unreliable communication, but instead to *verify* that particular measurements regarding fastest temporal paths in a periodic temporal graph are plausible (i. e., “realizable”). To this end, we introduce and investigate the following problem, capturing the setting described above:

SIMPLE PERIODIC TEMPORAL GRAPH REALIZATION (SIMPLE TGR)

Input: An integer $n \times n$ matrix D , a positive integer Δ .

Question: Does there exist a graph $G = (V, E)$ with vertices $\{v_1, \dots, v_n\}$ and a Δ -periodic labeling $\lambda : E \rightarrow \{1, 2, \dots, \Delta\}$ such that, for every i, j , the duration of a fastest temporal path from v_i to v_j in the Δ -periodic temporal graph (G, λ, Δ) is $D_{i,j}$?

We focus on exact algorithms. We start by showing NP-hardness of the problem (Theorem 1), even if Δ is a small constant. To establish a baseline for tractability, we show that SIMPLE TGR is polynomial-time solvable if the underlying graph is a tree (Theorem 3).

Building upon these initial results, we explore the possibilities to generalize our polynomial-time algorithm using the *distance-from-triviality* parameterization paradigm [29, 42]. That is, we investigate the parameterized computational complexity of SIMPLE TGR with respect to structural parameters of the underlying graph that measure its “tree-likeness”.

We obtain the following results. We show that SIMPLE TGR is W[1]-hard when parameterized by the feedback vertex number of the underlying graph (Theorem 2). To this end, we

first give a reduction from MULTICOLORED CLIQUE parameterized by the number of colors [28] to a variant of SIMPLE TGR where the period Δ is infinite, that is, when the labeling is non-periodic. We use a special gadget (the “infinity” gadget) which allows us to transfer the result to a finite period Δ . The latter construction is independent from the particular reduction we use, and can hence be treated as a reduction from the non-periodic to the periodic setting. Note that our parameterized hardness result rules out fixed-parameter tractability for several popular graph parameters such as *treewidth*, *degeneracy*, *cliquewidth*, *distance to chordal graphs*, and *distance to outerplanar graphs*.

We complement this hardness result by showing that SIMPLE TGR is fixed-parameter tractable (FPT) with respect to the *feedback edge number* k of the underlying graph (Theorem 4). This result also implies an FPT algorithm for any larger parameter, such as the *maximum leaf number*. A similar phenomenon of getting W[1]-hardness with respect to the feedback vertex number, while getting an FPT algorithm with respect to the feedback edge number, has been observed only in a few other temporal graph problems related to the connectivity between two vertices [14, 35, 23].

Our FPT algorithm works as follows on a high level. First, we distinguish $O(k^2)$ vertices which we call “important vertices”. Then, we guess the fastest temporal paths for each pair of these important vertices; as we prove, the number of choices we have for all these guesses is upper-bounded by a function of k . Then we also need to make several further guesses (again using a bounded number of choices), which altogether leads us to specify a small (i. e., bounded by a function of k) number of different configurations for the fastest paths between *all pairs* of vertices. For each of these configurations, we must then make sure that the labels of our solution will not allow any other temporal path from a vertex v_i to a vertex v_j have a *strictly smaller* duration than $D_{i,j}$. This naturally leads us to build one Integer Linear Program (ILP) for each of these configurations. We manage to formulate all these ILPs by having a number of variables that is upper-bounded by a function of k . Finally, we use Lenstra’s Theorem [51] to solve each of these ILPs in FPT time. In the end, our initial instance is a YES-instance if and only if at least one of these ILPs is feasible.

The above results provide a fairly complete picture of the parameterized computational complexity of SIMPLE TGR with respect to structural parameters of the underlying graph which measure “tree-likeness”. To obtain our results, we prove several properties of fastest temporal paths, which may be of independent interest.

Related work. Graph realization problems on static graphs have been studied since the 1960s. We provide an overview of the literature in the introduction. To the best of our knowledge, we are the first to consider graph realization problems in the temporal setting. Very recently, Erlebach et al. [26] have built upon our results and, among others, studied the case where edges might appear more than once in each period. Many other connectivity-related problems have been studied in the temporal setting [58, 2, 21, 60, 47, 19, 25, 31, 69, 12, 36], most of which are much more complex and computationally harder than their non-temporal counterparts, and some of which do not even have a non-temporal counterpart.

There are some problem settings that share similarities with ours, which we discuss now in more detail.

Several problems have been studied where the goal is to assign labels to (sets of) edges of a given static graph in order to achieve certain connectivity-related properties [48, 56, 1, 22]. The main difference to our problem setting is that in the mentioned works, the input is a graph and the sought labeling is not periodic. Furthermore, the investigated properties are temporal connectivity between all vertices [48, 56, 1], temporal connectivity among a subset of vertices [48], or reducing reachability among the vertices [22]. In all these cases, the duration

of the temporal paths has not been considered.

Finally, there are many models for dynamic networks in the context of distributed computing [50]. These models have some similarity to temporal graphs, in the sense that in both cases the edges appear and disappear over time. However, there are notable differences. For example, one important assumption in the distributed setting can be that the edge changes are adversarial or random (while obeying some constraints such as connectivity), and therefore they are not necessarily known in advance [50].

Preliminaries and notation. We already introduced the most central notions and concepts. There are some additional definitions we need, to present our proofs and results which we give in the following.

An interval in \mathbb{N} from a to b is denoted by $[a, b] = \{i \in \mathbb{N} : a \leq i \leq b\}$; similarly, $[a] = [1, a]$. An undirected graph $G = (V, E)$ consists of a set V of vertices and a set $E \subseteq V \times V$ of edges. For a graph G , we also denote by $V(G)$ and $E(G)$ the vertex and edge set of G , respectively. We denote an edge $e \in E$ between vertices $u, v \in V$ as a set $e = \{u, v\}$. For the sake of simplicity of the representation, an edge e is sometimes also denoted by uv . A path P in G is a subgraph of G with vertex set $V(P) = \{v_1, \dots, v_k\}$ and edge set $E(P) = \{\{v_i, v_{i+1}\} : 1 \leq i < k\}$ (we often represent path P by the tuple (v_1, v_2, \dots, v_k)). For two paths $P = (v_1, \dots, v_i)$ and $Q = (v_i, \dots, v_p)$, where $\{v_1, \dots, v_i\} \cap \{v_i, \dots, v_p\} = \{v_i\}$, we denote by $P \cup Q$ the path $(v_1, \dots, v_i, \dots, v_p)$. Furthermore, we denote by $P \setminus \{v_1, v_i\}$ the path (v_2, \dots, v_{i-1}) that is obtained from P by removing its first and last vertex.

Let v_1, v_2, \dots, v_n be the n vertices of the graph G . For simplicity of the presentation (and with a slight abuse of notation) we refer during the paper to the entry $D_{i,j}$ of the matrix D as $D_{a,b}$, where $a = v_i$ and $b = v_j$. That is, we put as indices of the matrix D the corresponding vertices of G whenever it is clear from the context.

Let $P = (u = v_1, v_2, \dots, v_p = v)$ be a path from u to v in G . Recall that, in our paper, every edge has exactly one time label in every period of Δ consecutive time steps. Therefore, as we are only interested in the fastest duration of temporal paths, many times we refer to (P, λ, Δ) as any of the temporal paths from $u = v_1$ to $v = v_p$ along the edges of P , which starts at the edge $v_1 v_2$ at time $\lambda(v_1 v_2) + c\Delta$, for some $c \in \mathbb{N}$, and then sequentially visits the rest of the edges of P as early as possible. We denote by $d(P, \lambda, \Delta)$, or simply by $d(P, \lambda)$ when Δ is clear from the context, the duration of any of the temporal paths (P, λ, Δ) ; note that they all have the same duration. Whenever we use the term *label of an edge* e , we actually mean $\lambda(e) \in [\Delta]$. Note that for a given path (P, λ, Δ) that passes through the edge e , the label used by P at that edge is $\lambda(e) + c\Delta$, for some $c \geq 0$. Many times we also refer to a path $P = (u = v_1, v_2, \dots, v_p = v)$ from u to v in G , as a temporal path in (G, λ, Δ) , where we actually mean that (P, λ, Δ) is a temporal path with P as its underlying (static) path.

We remark that a fastest path between two vertices in a temporal graph can be computed in polynomial time [11, 68]. Hence, given a Δ -periodic temporal graph (G, λ, Δ) , we can compute in polynomial-time the matrix D which consists of durations of fastest temporal paths among all pairs of vertices in (G, λ, Δ) .

We use standard terminology from parameterized complexity theory [20, 30, 18]. Let Σ denote a finite alphabet. A parameterized problem $L \subseteq \{(x, k) \in \Sigma^* \times \mathbb{N}_0\}$ is a subset of all instances (x, k) from $\Sigma^* \times \mathbb{N}_0$, where k denotes the *parameter*. A parameterized problem L is FPT (*fixed-parameter tractable*) if there is an algorithm that decides every instance (x, k) for L in $f(k) \cdot |x|^{O(1)}$ time, where f is any computable function only depending on the parameter. If a parameterized problem L is W[1]-hard, then it is presumably not fixed-parameter tractable.

Organization of the paper. In Section 2 we present our hardness results, first the NP-hardness in Section 2.1 and then the parameterized hardness in Section 2.2. In Section 3 we present our algorithmic results. First we give in Section 3.1 a polynomial-time algorithm for the case where the underlying graph is a tree. In Section 3.2 we generalize this and present our FPT result, which is the main result in the paper. Finally, we conclude in Section 4 and discuss some future work directions.

2. Hardness results for Simple TGR

In this section we present our main computational hardness results. In Section 2.1 we show that SIMPLE TGR is NP-hard even for constant Δ . In Section 2.2 we investigate the parameterized computational hardness of SIMPLE TGR with respect to structural parameters of the underlying graph. We show that SIMPLE TGR is W[1]-hard when parameterized by the feedback vertex number of the underlying graph.

2.1. NP-hardness of SIMPLE TGR

In this section, we prove that in general, it is NP-hard to determine a Δ -periodic temporal graph (G, λ) respecting a duration matrix D , even if Δ is a small constant.

Theorem 1. SIMPLE TGR is NP-hard for all $\Delta \geq 3$.

Proof. We present a polynomial-time reduction from the NP-hard problem MONOTONE NAE 3-SAT [66]. Here we are given a formula ϕ that is a conjunction of so-called NAE (not-all-equal) clauses, where each clause contains exactly 3 (distinct, non-negated) variables. A NAE clause evaluates to TRUE if and only if not all of its variables are set to equal values, that is, at least one variable is set to TRUE and at least one variable is set to FALSE. We are asked whether ϕ admits a satisfying assignment.

Given an instance ϕ of MONOTONE NAE 3-SAT, we construct an instance (D, Δ) of SIMPLE TGR as follows.

We start by describing the vertex set of the underlying graph G of D .

- For each variable x_i in ϕ , we create three variable vertices x_i, x_i^T, x_i^F .
- For each clause c in ϕ , we create one clause vertex c .
- We add one additional super vertex v .

Next, we describe the edge set of G .

- For each variable x_i in ϕ we add the following five edges: $\{x_i, x_i^T\}$, $\{x_i, x_i^F\}$, $\{x_i^T, x_i^F\}$, $\{x_i^T, v\}$, and $\{x_i^F, v\}$.
- For each pair of variables x_i, x_j in ϕ with $i \neq j$ we add the following four edges: $\{x_i^T, x_j^T\}$, $\{x_i^T, x_j^F\}$, $\{x_i^F, x_j^T\}$, and $\{x_i^F, x_j^F\}$.
- For each clause c in ϕ we add one edge for each variable in c . Let x_i appear in c , then we add edge $\{c, x_i^T\}$.

This finishes the construction of G . For an illustration see Fig. 2.

We set Δ to some constant larger than two, that is, $\Delta \geq 3$. Next, we specify the durations in the matrix D between all vertex pairs. We start by setting the value of $D_{u,v} = 1$ where u and v are two adjacent vertices in G .

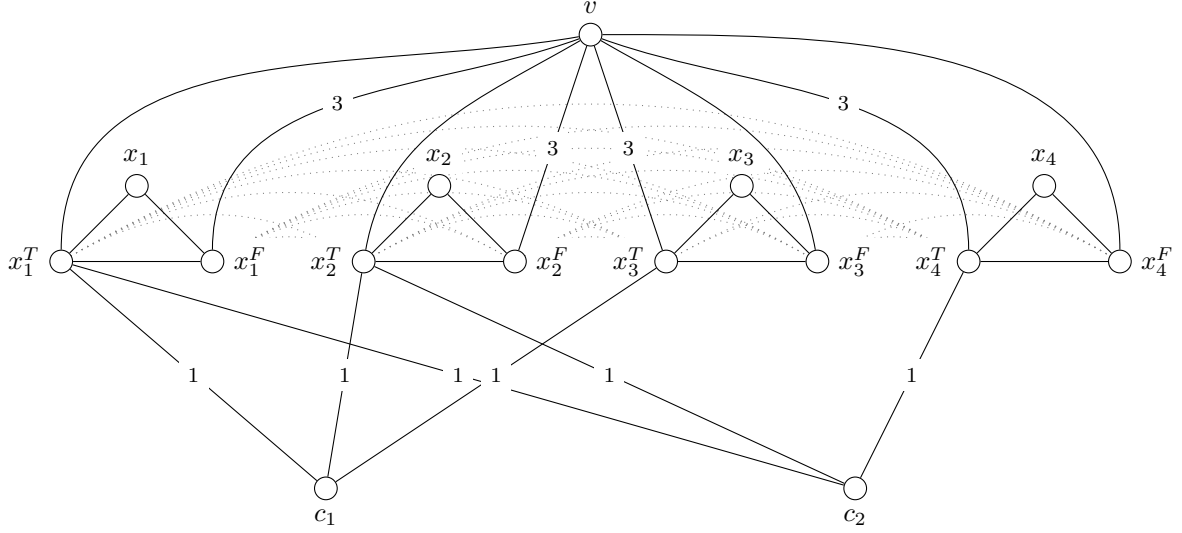


Figure 2: Illustration of the temporal graph (G, λ) from the NP-hardness reduction, where the NAE 3-SAT formula ϕ is of the form $\phi = \text{NAE}(x_1, x_2, x_3) \wedge \text{NAE}(x_1, x_2, x_4)$. To improve the readability, we draw edges between vertices x_i^T and x_j^F (where $i \neq j$) with gray dotted lines. Presented is the labeling of G corresponding to the assignment $x_1 = x_2 = \text{TRUE}$ and $x_3, x_4 = \text{FALSE}$, where all unlabeled edges get the label 2.

- For each variable x_i in ϕ and the super vertex v we specify the following durations: $D_{x_i, v} = 2$ and $D_{v, x_i} = \Delta$.
- For each clause c in ϕ and the super vertex v we specify the following durations: $D_{c, v} = 2$ and $D_{v, c} = \Delta - 1$.
- Let x_i be a variable that appears in clause c , then we specify the following durations: $D_{c, x_i} = 2$ and $D_{x_i, c} = \Delta$, $D_{c, x_i^F} = 2$, and $D_{x_i^F, c} = \Delta$.
- Let x_i be a variable that does *not* appear in clause c , then we specify the following durations: $D_{x_i, c} = 2\Delta$, $D_{c, x_i} = \Delta + 2$ and $D_{c, x_i^T} = D_{c, x_i^F} = 2$, and $D_{x_i^T, c} = D_{x_i^F, c} = \Delta$.
- For each pair of variables $x_i \neq x_j$ in ϕ we specify the following durations: $D_{x_i, x_j} = 2\Delta + 1$ and $D_{x_i, x_j^T} = D_{x_i, x_j^F} = \Delta + 1$.
- For each pair of clauses $c_i \neq c_j$ in ϕ we specify the following durations: $D_{c_i, c_j} = \Delta + 1$.

This finishes the construction of the instance (D, Δ) of SIMPLE TGR which can clearly be done in polynomial time. In the remainder, we show that (D, Δ) is a YES-instance of SIMPLE TGR if and only if the MONOTONE NAE 3-SAT formula ϕ is satisfiable.

(\Rightarrow): Assume the constructed instance (D, Δ) of SIMPLE TGR is a YES-instance. Then there exists a label $\lambda(e)$ for each edge $e \in E(G)$ such that for each vertex pair u, w in the temporal graph (G, λ, Δ) we have that a fastest temporal path from u to w is of duration $D_{u, w}$.

We construct a satisfying assignment for ϕ as follows. For each variable x_i , if $\lambda(\{x_i, x_i^T\}) = \lambda(\{x_i^T, v\})$, then we set x_i to TRUE, otherwise we set x_i to FALSE.

To show that this yields a satisfying assignment, we need to prove some properties of the labeling λ . First, observe that adding an integer t to all time labels does not change the duration of any temporal paths. Second, observe that if for two vertices u, w we have that $D_{u, w}$ equals the distance between u and w in G (i.e., the duration of a fastest temporal path from u to w equals the distance of the shortest path between u and w), then there is a shortest path P from u to w in G such that the labeling λ assigns consecutive time labels to the edges of P .

Let $\lambda(\{x_i, x_i^T\}) = t$ and $\lambda(\{x_i, x_i^F\}) = t'$, for an arbitrary variable x_i . If both $\lambda(\{x_i^T, v\}) \neq t+1$ and $\lambda(\{x_i^F, v\}) \neq t'+1$, then $D_{x_i, v} > 2$, which is a contradiction. Thus, for every variable x_i , we have that $\lambda(\{x_i^T, v\}) = t+1$ or $\lambda(\{x_i^F, v\}) = t'+1$ (or both). In particular, this means that if $\lambda(\{x_i, x_i^F\}) = \lambda(\{x_i^F, v\})$, then we set x_i to FALSE, since in this case $\lambda(\{x_i, x_i^T\}) \neq \lambda(\{x_i^T, v\})$.

Now assume for a contradiction that the described assignment is not satisfying. Then there exists a clause c that is not satisfied. Suppose that x_1, x_2, x_3 are three variables that appear in c . Recall that $D_{c, v} = 2$ and $D_{v, c} = \Delta - 1$. The fact that $D_{c, v} = 2$ implies that we must have a temporal path consisting of two edges from c to v , such that the two edges have consecutive labels. By construction of G there are three candidates for such a path, one for each variable of c . Assume that the temporal path realizing $D_{c, v} = 2$ goes through vertex x_1^T . Let us denote with $t = \lambda(\{x_1^T, v\})$. It follows that $\lambda(\{x_1^T, c\}) = \lambda(\{x_1^T, v\}) - 1 = t - 1$. Furthermore, since $D_{c, x_1} = 2$ we also have that $\lambda(\{x_1^T, c\}) = \lambda(\{x_1, x_1^T\}) - 1$. Therefore $\lambda(\{x_1, x_1^T\}) = \lambda(\{x_1^T, v\}) = t$. This implies that x_1 is set to TRUE. Let us observe paths from v to c . We know that $D_{v, c} = \Delta - 1$. The underlying path of a fastest temporal path from v to c , which goes through x_1^T , is the path $P = (v, x_1^T, c)$. Since $\lambda(\{x_1^T, c\}) < \lambda(\{x_1^T, v\})$ we get that the duration of the temporal path (P, λ) is equal to $D_{P, \lambda} = (\Delta + t - 1) - t + 1 = \Delta$. This implies that (P, λ) is not a fastest temporal path from v to c and therefore does not pass through x_1^T . Since there are only two other vertices connected to c , we have only two other edges incident to c , that can be used on a fastest temporal path from v to c . Suppose now that the temporal path realizing $D_{v, c} = \Delta - 1$ goes through vertex x_2^T . Let us denote with $t' = \lambda(\{x_2^T, v\})$. Since the duration of a fastest temporal path from v to c is $\Delta - 1$, and the edge $\{x_2^T, c\}$ is the only edge incident to vertex c and edge $\{x_2^T, v\}$, it follows that $\lambda(\{x_2^T, c\}) \geq \lambda(\{x_2^T, v\}) - 2 = t' - 2$. Since $D_{x_2, v} = 2$ it follows that $\lambda(\{x_2, x_2^T\}) = \lambda(\{x_2^T, v\}) - 1 = t' - 1$. Knowing this and the fact that $D_{x_2, c} = 2$, we get that $\lambda(\{x_2^T, c\})$ must be equal to $t' - 2$. Therefore a fastest temporal path from v to c passes through edges $\{x_2^T, v\}$ and $\{x_2^T, c\}$. In the above we have also determined that $\lambda(\{x_2, x_2^T\}) \neq \lambda(\{x_2^T, v\})$, which implies that x_2 is set to FALSE. But now we have that x_1, x_2 both appear in c , where one of them is TRUE, while the other is FALSE, which implies that the clause c is satisfied, a contradiction.

(\Leftarrow): Assume that ϕ is satisfiable. Then there exists a satisfying assignment for the variables in ϕ . We construct a labeling λ as follows.

- All edges incident with a clause vertex c obtain label one.
- If variable x_i is set to TRUE, we set $\lambda(\{x_i^F, v\}) = 3$.
- If variable x_i is set to FALSE, we set $\lambda(\{x_i^T, v\}) = 3$.
- We set the labels of all other edges to two.

For an example of the constructed temporal graph see Fig. 2. We now verify that all durations are realized. For the remainder of the proof denote by $d(u, v)$ the duration of the fastest path from u to v in the constructed temporal graph.

- For each variable x_i in ϕ we have to check that $d(x_i, v) = 2$ and $d(v, x_i) = \Delta$.

If x_i is set to TRUE, then there is a temporal path from x_i to v via x_i^F of duration 2, since $\lambda(\{x_i, x_i^F\}) = 2$ and $\lambda(\{x_i^F, v\}) = 3$. For a temporal path from v to x_i we observe the following. The only possible labels to leave the vertex v are 2 and 3, which take us from v to x_j^T or x_j^F of some variable x_j . The only two edges incident to x_i have labels 2, therefore the fastest path from v to x_i cannot finish before the time $\Delta + 2$. The fastest way to leave v and enter to x_i would then be to leave v at edge $\{x_i^F, v\}$ with label 3, and continue to x_i at time $\Delta + 2$, which gives us the desired duration Δ .

If x_i is set to FALSE, then, by similar arguing, there is a temporal path from x_i to v via x_i^T of duration 2, and a temporal path from v to x_i , through x_i^F of duration Δ .

- For each clause c in ϕ we have to check that $d(c, v) = 2$ and $d(v, c) = \Delta - 1$:

Suppose x_i, x_j, x_k appear in c . Since we have a satisfying assignment at least one of the variables in c is set to TRUE and at least one to FALSE. Suppose x_i is the variable that is TRUE in c , and x_j is the variable that is FALSE in c . Then there is a temporal path from c to v through x_i^T such that $\lambda(\{x_i^T, c\}) = 1$ and $\lambda(\{x_i^T, v\}) = 2$. Furthermore, there is a temporal path from v to c through x_j^T such that $\lambda(\{x_j^T, v\}) = 3$ and $\lambda(\{x_j^T, c\}) = 1$, which results in a temporal path from v to c of duration $\Delta - 1$.

- Let x_i be a variable that appears in clause c . We have to check that $d(c, x_i) = d(c, x_i^F) = 2$ and $d(x_i, c) = d(x_i^F, c) = \Delta$.

There is a temporal path from c to x_i via x_i^T and also a temporal path from c to x_i^F via x_i^T such that $\lambda(\{x_i^T, c\}) = 1$ and $\lambda(\{x_i, x_i^T\}) = \lambda(\{x_i^F, x_i^T\}) = 2$, which proves the first equality. There are also the following two temporal paths, first, from x_i to c through x_i^T and second, from x_i^F to c through x_i^T . Both of the temporal paths start on the edge with label 2, as $\lambda(\{x_i, x_i^T\}) = \lambda(\{x_i^F, x_i^T\}) = 2$ and finish on the edge with label 1, as $\lambda(\{x_i^T, c\}) = 1$.

- Let x_i be a variable that does *not* appear in clause c , then we have to check that first, $d(c, x_i^T) = d(c, x_i^F) = 2$, second, $d(x_i^T, c) = d(x_i^F, c) = \Delta$, third, $d(c, x_i) = \Delta + 2$, and fourth $d(x_i, c) = 2\Delta$.

Let x_j be a variable that appears in c . Then there is a temporal path from c to x_i^T via x_j^T and also a temporal path from c to x_i^F via x_j^T such that $\lambda(\{x_j^T, c\}) = 1$ and $\lambda(\{x_j^T, x_i^T\}) = \lambda(\{x_j^T, x_i^F\}) = 2$, which proves the first equality. Using the same temporal path in the opposite direction, i.e., first the edge $x_j^T c$ and then one of the edges $\{x_j^T, x_i^T\}$ or $\{x_j^T, x_i^F\}$ at times 2 and $\Delta + 1$, respectively, yields the second equality. For a temporal path from c to x_i we traverse the following three edges $\{x_j^T, c\}$, $\{x_j^T, x_i^F\}$, and $\{x_i^F, x_i\}$, with labels 1, 2, and 2 respectively (i.e., the path traverses them at time 1, 2 and $\Delta + 2$, respectively), which proves the third equality. Now for the case of a temporal path from x_i to c , we use the same three edges, but in the opposite direction, namely $\{x_i^F, x_i\}$, $\{x_j^T, x_i^F\}$, and $\{x_j^T, c\}$, again at times 2, $\Delta + 2$, and $2\Delta + 1$, respectively, which proves the last equality. Note that all of the above temporal paths are also the shortest possible, and since the labels of the first and last edges (of these paths) are unique, it follows that we cannot find faster temporal paths.

- For each pair of variables $x_i \neq x_j$ in ϕ we have to check that $d(x_i, x_j) = 2\Delta + 1$ and $d(x_i, x_j^T) = d(x_i, x_j^F) = \Delta + 1$.

There is a path from x_i to x_j that passes first through one of the vertices x_i^T or x_i^F , and then through one of the vertices x_j^T or x_j^F . This temporal path is of length 3, where all of the edges have label 2, which proves the first equality. Now, a temporal path from x_i to x_j^T (resp. x_j^F), passes through one of the vertices x_i^T or x_i^F . This path is of length two, where all of the edges have label 2, which proves the second equality. Note that all of the above temporal paths are also the shortest possible, and since the labels of the first and last edges (of these paths) are unique, it follows that we cannot find faster temporal paths.

- For each pair of clauses $c_i \neq c_j$ in ϕ we have to check that $d(c_i, c_j) = \Delta + 1$.

Let x_k be a variable that appears in c_i and x_ℓ the variable that appears in c_j . There is a path of length three from c_i to c_j that passes first through vertex x_k^T and then through vertex x_ℓ^T . Therefore, the temporal path from c_i to c_j uses the edges $\{x_k^T, c_i\}$, $\{x_\ell^T, c_j\}$, and $\{x_k^T, x_\ell^T\}$, with labels 1, 2, and 1 (at times 1, 2, and $\Delta + 1$), respectively, which proves the desired equality. Note also that this is the shortest path between c_i and c_j , and that the first and the last edge must have the label 1, therefore, it follows that this is a fastest temporal path.

Lastly, observe that the above constructed labeling λ uses values $\{1, 2, 3\} \subseteq [\Delta]$, therefore the reduction holds for any $\Delta \geq 3$. \square

2.2. Parameterized hardness of SIMPLE TGR

In this section, we investigate the parameterized hardness of SIMPLE TGR with respect to structural parameters of the underlying graph. We show that the problem is W[1]-hard when parameterized by the feedback vertex number of the underlying graph. The *feedback vertex number* of a graph G is the cardinality of a minimum vertex set $X \subseteq V(G)$ such that $G - X$ is a forest. The set X is called a *feedback vertex set*. Note that, in contrast to the result of the previous section (Theorem 1), the reduction we use to obtain the following result does not produce instances with a constant Δ .

Theorem 2. *SIMPLE TGR is W[1]-hard when parameterized by the feedback vertex number of the underlying graph.*

Proof. We present a parameterized reduction from the W[1]-hard problem MULTICOLORED CLIQUE parameterized by the number of colors [28]. Here, given a k -partite graph $H = (W_1 \uplus W_2 \uplus \dots \uplus W_k, F)$, we are asked whether H contains a clique of size k . If $w \in W_i$, then we say that w has *color* i . W.l.o.g. we assume that $|W_1| = |W_2| = \dots = |W_k| = n$. Furthermore, for all $i \in [k]$, we assume the vertices in W_i are ordered in some arbitrary but fixed way, that is, $W_i = \{w_1^i, w_2^i, \dots, w_n^i\}$. Let $F_{i,j}$ with $i < j$ denote the set of all edges between vertices from W_i and W_j . We assume w.l.o.g. that $|F_{i,j}| = m$ for all $i < j$ (if not we can add $k \max_{i,j} |F_{i,j}|$ vertices to each W_i and use those to add up to $\max_{i,j} |F_{i,j}|$ additional isolated edges to each $F_{i,j}$). Furthermore, for all $i < j$ we assume that the edges in $F_{i,j}$ are ordered in some arbitrary but fixed way, that is, $F_{i,j} = \{e_1^{i,j}, e_2^{i,j}, \dots, e_m^{i,j}\}$.

We give a reduction to a variant of SIMPLE TGR where the period Δ is infinite (that is, the sought temporal graph is not periodic and the labeling function $\lambda : E \rightarrow \mathbb{N}$ maps to the natural numbers) and we allow D to have infinity entries, meaning that the two respective vertices are not temporally connected. Note that, given the matrix D , we can easily compute the underlying graph G , as follows. Two vertices v, v' are adjacent in G if and only if $D_{v,v'} = 1$, as having an edge between v and v' is the only way that there exists a temporal path from v to v' with duration 1. For simplicity of the presentation of the reduction, we describe the underlying graph G (which directly implies the entries of D where $D_{v,v'} = 1$) and then we provide the remaining entries of D . At the end of the proof, we show how to obtain the result for a finite Δ and a matrix D of durations of fastest paths, that only has finite entries.

In the following, we give an informal description of the main ideas of the reduction. The construction uses several gadgets, where the main ones are an “edge selection gadget” and a “verification gadget”.

Every *edge selection gadget* is associated with a color combination i, j in the MULTICOLORED CLIQUE instance, and its main purpose is to “select” an edge connecting a vertex from color i with a vertex from color j . Roughly speaking, the edge selection gadget consists of m paths, one for every edge in $F_{i,j}$ (see Fig. 3 for reference). The distance matrix D will enforce that the

labels on those paths effectively order them temporally, that is, in particular, the labels on one of the paths will be smaller than the labels on all other paths. The edge corresponding to this path is selected.

We have a *verification gadget* for every color i . They interact with the edge selection gadgets as follows. The verification gadget for color i is connected to all edge selection gadgets that involve color i . More specifically, this is connected to every path corresponding to an edge at a position in the path that encodes the endpoint of color i of that edge (again, see Fig. 3 for reference). Intuitively, the distances in the verification gadget are only realizable if the selected edges all have the same endpoint of color i . Hence, the distances of all verification gadgets can be realized if and only if the selected edges form a clique.

Furthermore, we use an *alignment gadget* which, intuitively, ensures that the labelings of all gadgets use the same range of time labels. Finally, we use *connector gadgets* which create shortcuts between all vertex pairs that are irrelevant for the functionality of the other gadgets. This allows us to easily fill in the distance matrix with the corresponding values. We ensure that all our gadgets have a constant feedback vertex number, hence the overall feedback vertex number is quadratic in the number of colors of the MULTICOLORED CLIQUE instance and we get the parameterized hardness result.

In the following, for every gadget, we first give a formal description of the underlying graph of this gadget (i. e., not the complete distance sub-matrix of the gadget). Afterwards, we define the corresponding entries in the distance matrix D .

Given an instance H of MULTICOLORED CLIQUE, we construct an instance D of SIMPLE TGR (with infinity entries and no periods) as follows.

Edge selection gadget. We first introduce an *edge selection gadget* $G_{i,j}$ for color combination i, j with $i < j$. We start with describing the vertex set of the gadget.

- A set $X_{i,j}$ of vertices x_1, x_2, \dots, x_m .
- Vertex sets U_1, U_2, \dots, U_m with $4n + 1$ vertices each, that is, $U_\ell = \{u_0^\ell, u_1^\ell, u_2^\ell, \dots, u_{4n}^\ell\}$ for all $\ell \in [m]$.
- Two special vertices $v_{i,j}^*, v_{i,j}^{**}$.

The gadget has the following edges.

- For all $\ell \in [m]$ we have edge $\{x_\ell, v_{i,j}^*\}$, $\{v_{i,j}^*, u_0^\ell\}$, and $\{u_{4n}^\ell, v_{i,j}^{**}\}$.
- For all $\ell \in [m]$ and $\ell' \in [4n]$, we have edge $\{u_{\ell'-1}^\ell, u_{\ell'}^\ell\}$.

Verification gadget. For each color i , we introduce the following vertices. What we describe in the following will be used as a *verification gadget for color i* .

- We have one vertex y^i and $k + 1$ vertices v_ℓ^i for $0 \leq \ell \leq k$.
- For every $\ell \in [m]$ and every $j \in [k] \setminus \{i\}$ we have $5n$ vertices $a_1^{i,j,\ell}, a_2^{i,j,\ell}, \dots, a_{5n}^{i,j,\ell}$ and $5n$ vertices $b_1^{i,j,\ell}, b_2^{i,j,\ell}, \dots, b_{5n}^{i,j,\ell}$.
- We have a set \hat{U}_i of $13n + 1$ vertices $\hat{u}_1^i, \hat{u}_2^i, \dots, \hat{u}_{13n+1}^i$.

We add the following edges. We add edge $\{y^i, v_0^i\}$. For every $\ell \in [m]$, every $j \in [k] \setminus \{i\}$, and every $\ell' \in [5n - 1]$ we add edge $\{a_{\ell'}^{i,j,\ell}, a_{\ell'+1}^{i,j,\ell}\}$ and we add edge $\{b_{\ell'}^{i,j,\ell}, b_{\ell'+1}^{i,j,\ell}\}$.

Let $1 \leq j < i$ (skip if $i = 1$), let $e_\ell^{j,i} \in F_{j,i}$, and let $w_{\ell'}^i \in W_i$ be incident with $e_\ell^{j,i}$. Then we add edge $\{v_{j-1}^i, a_1^{i,j,\ell}\}$ and we add edge $\{a_{5n}^{i,j,\ell}, u_{\ell'-1}^\ell\}$ between $a_{5n}^{i,j,\ell}$ and the vertex $u_{\ell'-1}^\ell$ of the edge selection gadget of color combination j, i . Furthermore, we add edge $\{v_j^i, b_1^{i,j,\ell}\}$ and edge $\{b_{5n}^{i,j,\ell}, u_{\ell'}^\ell\}$ between $b_{5n}^{i,j,\ell}$ and the vertex $u_{\ell'}^\ell$ of the edge selection gadget of color combination j, i .

We add edge $\{v_{i-1}^i, \hat{u}_1^i\}$ and for all $\ell'' \in [13n]$ we add edge $\{\hat{u}_{\ell''}^i, \hat{u}_{\ell''+1}^i\}$. Furthermore, we add edge $\{\hat{u}_{13n+1}^i, v_i^i\}$.

Let $i < j \leq k$ (skip if $i = k$), let $e_\ell^{i,j} \in F_{i,j}$, and let $w_{\ell'}^i \in W_i$ be incident with $e_\ell^{i,j}$. Then we add edge $\{v_{j-1}^i, a_1^{i,j,\ell}\}$ and edge $\{a_{5n}^{i,j,\ell}, u_{3n+\ell'-1}^\ell\}$ between $a_{5n}^{i,j,\ell}$ and the vertex $u_{3n+\ell'-1}^\ell$ of the edge selection gadget of color combination i, j . Furthermore, we add edge $\{v_j^i, b_1^{i,j,\ell}\}$ and edge $\{b_{5n}^{i,j,\ell}, u_{3n+\ell'}^\ell\}$ between $b_{5n}^{i,j,\ell}$ and the vertex $u_{3n+\ell'}^\ell$ of the edge selection gadget of color combination i, j .

Connector gadget. Next, we describe *connector gadgets*. Intuitively, these gadgets will be used to connect many vertex pairs by fast paths, which will make arguing about possible labelings in YES-instances much easier. Connector gadgets consist of six vertices $\hat{v}_0, \hat{v}_0', \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_3'$. Each connector gadget is associated with two sets A, B with $B \subseteq A$ containing vertices of other gadgets. Let V^* denote the set of all vertices from all edge selection gadgets and all verification gadgets. The sets A and B will only play a role when defining the matrix D later. Informally speaking, vertices in A should reach all vertices in V^* quickly through the gadget, except the ones in B . We have the following edges.

- Edges $\{\hat{v}_0, \hat{v}_1\}, \{\hat{v}_0', \hat{v}_1\}, \{\hat{v}_1, \hat{v}_2\}, \{\hat{v}_2, \hat{v}_3\}, \{\hat{v}_2, \hat{v}_3'\}$.
- An edge between \hat{v}_1 and each vertex in V^* .
- An edge between \hat{v}_2 and each vertex in V^* .

We add two connector gadgets for each edge selection gadget and two connector gadgets for each verification gadget.

The *first connector gadget for the edge selection gadget of color combination i, j with $i < j$* has the following sets.

- Sets A and B contain all vertices in $X_{i,j}$ and vertex $v_{i,j}^{**}$.

The *second connector gadget for the edge selection gadget of color combination i, j with $i < j$* has the following sets.

- Set A contains all vertices from the edge selection gadget $G_{i,j}$ except vertices in $X_{i,j}$.
- Set B is empty.

The *first connector gadget for the verification gadget of color i* has the following sets.

- Sets A and B contain all vertices v_ℓ^i with $0 \leq \ell \leq k$.

The *second connector gadget for the verification gadget of color i* has the following sets.

- Set A contains all vertices of the verification gadget except vertices v_ℓ^i with $0 \leq \ell \leq k$.
- Set B is empty.

Alignment gadget. Lastly, we introduce an *alignment gadget*. It consists of one vertex w^* and a set of vertices \hat{W} containing one vertex for each edge selection gadget, one vertex for each verification gadget, and one vertex for each connector gadget. Vertex w^* is connected to each vertex in \hat{W} . The vertex x_1 of each edge selection gadget, the vertex y^i of each verification gadget, and the vertex \hat{v}_1 of each connector gadget are each connected to one vertex in \hat{W} such that all vertices in \hat{W} have degree two. Intuitively, this gadget is used to relate labels of different gadgets to each other.

Feedback vertex number. This finished the description of the underlying graph G . For an illustration see Fig. 3. We can observe that the vertex set containing

- vertices $v_{i,j}^*$ and $v_{i,j}^{**}$ of each edge selection gadget,
- vertices v_ℓ^i with $0 \leq \ell \leq k$ of each verification gadget,
- vertices \hat{v}_1 and \hat{v}_2 of each connector gadget, and
- vertex w^* of the alignment gadget

forms a feedback vertex set in G with size $O(k^2)$.

Duration matrix D . We proceed with describing the matrix D of durations of fastest paths. The durations we define are numbered, and during the proofs, we will reference them using those numbers. For a more convenient presentation, we use the notation $d(v, v') := D_{v,v'}$. For all vertices v, v' that are neighbors in G we have that $d(v, v') = 1$ and $d(v', v) = 1$.

Next, consider a connector gadget consisting of vertices $\hat{v}_0, \hat{v}'_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}'_3$ and with sets A and B . Informally, the connector gadget makes sure that all vertices in A can reach all other vertices (of edge selection gadgets and verification gadgets) except the ones in B . We set the following durations. Recall that V^* denotes the set of all vertices from all edge selection gadgets and all verification gadgets.

1. We set $d(\hat{v}_0, \hat{v}_2) = d(\hat{v}_3, \hat{v}_1) = d(\hat{v}_2, \hat{v}'_0) = d(\hat{v}_1, \hat{v}'_3) = 2$, and $d(\hat{v}_0, \hat{v}'_0) = d(\hat{v}_3, \hat{v}'_3) = d(\hat{v}_0, \hat{v}'_3) = d(\hat{v}_3, \hat{v}'_0) = 3$.
2. Let $v \in A$, then we set $d(v, \hat{v}'_0) = 3$ and $d(v, \hat{v}'_3) = 3$.
3. Let $v \in V^* \setminus B$, then we set $d(\hat{v}_0, v) = 3$ and $d(\hat{v}_3, v) = 3$.
4. Let $v \in A$ and $v' \in V^* \setminus B$ such that v and v' are not neighbors, then we set $d(v, v') = 3$.

Now consider two connector gadgets, one with vertices $\hat{v}_0, \hat{v}'_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}'_3$ and with sets A and B , and one with vertices $\hat{v}_0, \hat{v}''_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}''_3$ and with sets A' and B' .

5. If there is a vertex $v \in A$ with $v \notin A'$, then we set $d(\hat{v}_1, \hat{v}'_1) = 3$.
6. If there is a vertex $v \in A$ with $v \in A' \setminus B'$, then we set $d(\hat{v}_1, \hat{v}'_2) = 3$.
7. If there is a vertex $v \in V^* \setminus (A \setminus B)$ with $v \notin A'$, then we set $d(\hat{v}_2, \hat{v}'_1) = 3$.
8. If there is a vertex $v \in V^* \setminus (A \setminus B)$ with $v \in A' \setminus B'$, then we set $d(\hat{v}_2, \hat{v}'_2) = 3$.

Next, consider the edge selection gadget for color combination i, j with $i < j$.

9. Let $1 \leq \ell < \ell' \leq m$. We set $d(x_\ell, x_{\ell'}) = 2n \cdot (i + j) \cdot ((\ell')^2 - \ell^2) + 1$.
10. For all $\ell \in [m]$ we set $d(x_\ell, v_{i,j}^{**}) = 8n + 5$.

Next, consider the verification gadget for color i . For all $0 \leq j < j' < i$ and all $i \leq j < j' \leq k$ we set the following.

11. We set $d(v_j^i, v_{j'}^i) = (20n + 6)(j' - j) - 1$.

For all $0 \leq j < i$ and all $i \leq j' \leq k$ we set the following.

12. We set $d(v_j^i, v_{j'}^i) = (20n + 6)(j' - j) + 6n - 1$.

Finally, we consider the alignment gadget. Let x_1 belong to the edge selection gadget of color combination i, j and let $w \in \hat{W}$ denote the neighbor of x_1 in the alignment gadget. Let \hat{v}_1 and \hat{v}_2 belong to the first connector gadget of the edge selection gadget for color combination i, j . Let \hat{V} contain all vertices \hat{v}_1 and \hat{v}_2 belonging to the other connector gadgets (different from the first one of the edge selection gadget for color combination i, j).

13. We set $d(w^*, x_1) = (20n + 6)(i + j)$.
14. We set $d(w^*, \hat{v}_1) = n^9$, $d(w, \hat{v}_2) = n^9$, $d(w, \hat{v}_1) = n^9 - (20n + 6)(i + j) + 1$, and $d(w, \hat{v}_2) = n^9 - (20n + 6)(i + j) + 1$.
15. For each vertex $v \in (V^* \cup \hat{V}) \setminus (X_{i,j} \cup \{v_{i,j}^{**}\})$ we set $d(w^*, v) = n^9 + 2$ and $d(w, v) = n^9 - (20n + 6)(i + j) + 3$.

Let y^i belong to the verification gadget of color i and let $w' \in \hat{W}$ denote the neighbor of y^i in the alignment gadget. Let \hat{v}_1 and \hat{v}_2 belong to the connector gadget of the verification gadget for color i . Let \hat{V} contain all vertices \hat{v}_1 and \hat{v}_2 belonging to the other connector gadgets (different from the one of the verification gadget for color i). Let V_i denote the set of all vertices of the verification gadget of color i .

16. We set $d(w^*, y^i) = n^8 - 1$, $d(w', v_0^i) = 2$, and $d(w^*, v_0^i) = n^8$.
17. We set $d(w^*, \hat{v}_1) = n^9$, $d(w^*, \hat{v}_2) = n^9$, $d(w', \hat{v}_1) = n^9 - n^8$, and $d(w', \hat{v}_2) = n^9 - n^8$.
18. For each vertex $v \in (V^* \cup \hat{V}) \setminus V_i$ we set $d(w^*, v) = n^9 + 1$, $d(w, v) = n^9 - n^8 + 2$, and $d(y^i, v) = n^9 - n^8 + 2$.

Let \hat{v}_0 belong to some connector gadget.

19. We set $d(w^*, \hat{v}_0) = n^9$.

All fastest path durations between non-adjacent vertex pairs that are not specified above are set to infinity.

Correctness. This finishes the construction of SIMPLE PERIODIC TEMPORAL GRAPH REALIZATION instance D , which can clearly be computed in polynomial time. For an illustration see Fig. 3. As discussed earlier, we have that the vertex cover number of the underlying graph of the instance is in $O(k^2)$.

In the remainder, we prove that D is a YES-instance of SIMPLE PERIODIC TEMPORAL GRAPH REALIZATION if and only if the H is a YES-instance of MULTICOLORED CLIQUE.

(\Rightarrow): Assume D is a YES-instance of SIMPLE PERIODIC TEMPORAL GRAPH REALIZATION and let (G, λ) be a solution. We have that the underlying graph G is uniquely defined by D . We first prove a number of properties of λ that we need to define a set of vertices in H which we claim to be a multicolored clique.

To start, consider the alignment gadget. We can observe that all edges incident with w^* have the same label.

Claim 1. For all $w \in \hat{W}$ we have that $\lambda(\{w^*, w\}) = t$ for some $t \in \mathbb{N}$.

Proof of Claim. Assume for contradiction that there are $w, w' \in \hat{W}$ such that $\lambda(\{w^*, w\}) = t$ and $\lambda(\{w^*, w'\}) = t'$ with $t \neq t'$. Let w.l.o.g. $t < t'$. Then w can reach w' , however we have that $d(w, w') = \infty$, a contradiction. \diamond

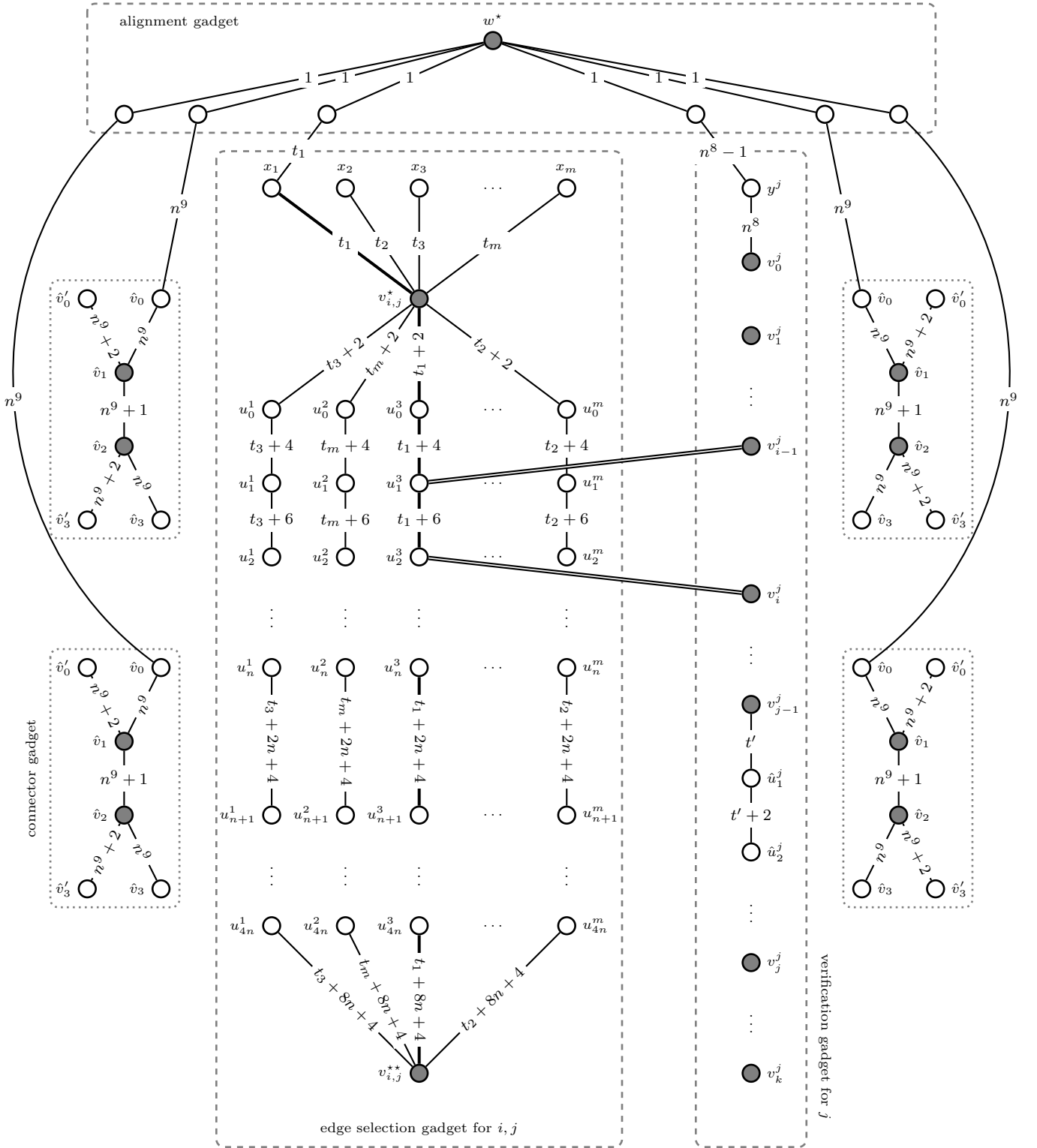


Figure 3: Illustration of part of the underlying graph G and a possible labeling. Edges incident with vertices \hat{v}_1, \hat{v}_2 of connector gadgets are omitted. Gray vertices form a feedback vertex set. The double line connections, between a vertex v_{i-1}^j in the verification gadget, and u_1^3 in the edge selection gadget, and, between a vertex u_2^3 in the edge selection gadget, and v_i^j in the verification gadget, consist of $5n$ vertices $a_1^{j,i,3}, a_2^{j,i,3}, \dots, a_{5n}^{j,i,3}$ and $b_1^{j,i,3}, b_2^{j,i,3}, \dots, b_{5n}^{j,i,3}$, respectively.

Claim 1 states that, in the temporal labeling λ , all edges $\{w^*, w\}$ have equal label $\lambda(\{w^*, w\}) = t$ for every $w \in \hat{W}$. Therefore, by adding $\Delta - t + 1$ to *every* edge of the graph, we end up with an equivalent temporal graph, where $\lambda(\{w^*, w\}) = 1$ for every $w \in \hat{W}$. Therefore, from now on, we assume w.l.o.g. that all edges incident with vertex w^* of the alignment gadget have label 1.

Next, we analyze the labelings of connector gadgets. We show that all edges incident with vertices of connector gadgets have labels of at least n^9 and at most $n^9 + 2$. More precisely, we show the following.

Claim 2. *Let $\hat{v}_0, \hat{v}'_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}'_3$ be the vertices of a connector gadget with sets A and B . Then we have that $\lambda(\{\hat{v}_0, \hat{v}_1\}) = n^9$, $\lambda(\{\hat{v}'_0, \hat{v}_1\}) = n^9 + 2$, $\lambda(\{\hat{v}_1, \hat{v}_2\}) = n^9 + 1$, $\lambda(\{\hat{v}_2, \hat{v}_3\}) = n^9$, and $\lambda(\{\hat{v}_2, \hat{v}'_3\}) = n^9 + 2$. Furthermore, for all $v \in V^*$ we have $n^9 \leq \lambda(\{\hat{v}_1, v\}) \leq n^9 + 2$ and $n^9 \leq \lambda(\{\hat{v}_2, v\}) \leq n^9 + 2$.*

Proof of Claim. Let $w \in \hat{W}$ denote the vertex of the alignment gadget that is a neighbor of w^* and \hat{v}_0 . We have $d(w^*, \hat{v}_0) = n^9$ (see duration nr. 19). It follows that $\lambda(\{w, \hat{v}_0\}) = n^9$. Since $d(\hat{v}_1, w) = \infty$ and $d(w, \hat{v}_1) = \infty$, we have that $\lambda(\{\hat{v}_0, \hat{v}_1\}) = n^9$. Note that \hat{v}_1 is the only common neighbor of \hat{v}_0 and \hat{v}_2 and the only common neighbor of \hat{v}_0 and \hat{v}'_0 . Since $d(\hat{v}_0, \hat{v}_2) = 2$ and $d(\hat{v}_0, \hat{v}'_0) = 3$ (see duration nr. 1) we have that $\lambda(\{\hat{v}_1, \hat{v}_2\}) = n^9 + 1$ and $\lambda(\{\hat{v}'_0, \hat{v}_1\}) = n^9 + 2$. Similarly, we have that \hat{v}_2 is the only common neighbor of \hat{v}_3 and \hat{v}_1 and the only common neighbor of \hat{v}_3 and \hat{v}'_3 . Since $d(\hat{v}_3, \hat{v}_1) = 2$ and $d(\hat{v}_3, \hat{v}'_3) = 3$ (see duration nr. 1) we have that $\lambda(\{\hat{v}_2, \hat{v}_3\}) = n^9$ and $\lambda(\{\hat{v}_2, \hat{v}'_3\}) = n^9 + 2$.

Let $v \in V^*$. Note that $d(v, \hat{v}_0) = \infty$ and $d(v, \hat{v}_3) = \infty$. It follows that $\lambda(\{\hat{v}_1, v\}) \geq n^9$ and $\lambda(\{\hat{v}_2, v\}) \geq n^9$. Otherwise, there would be a temporal path from v to \hat{v}_0 via \hat{v}_1 or a temporal path from v to \hat{v}_3 via \hat{v}_2 , a contradiction. Furthermore, note that $d(\hat{v}'_0, v) = \infty$ and $d(\hat{v}'_3, v) = \infty$. It follows that $\lambda(\{\hat{v}_1, v\}) \leq n^9 + 2$ and $\lambda(\{\hat{v}_2, v\}) \leq n^9 + 2$. Otherwise, there would be a temporal path from \hat{v}'_0 to v via \hat{v}_1 or a temporal path from \hat{v}'_3 to v via \hat{v}_2 , a contradiction. \diamond

Now we take a closer look at the edge selection gadgets. We make a number of observations that will allow us to define a set of vertices in H that we claim to be a multicolored clique.

Claim 3. *For all $1 \leq i < j \leq k$ and $\ell \in [m]$ we have that $\lambda(\{u_{4n}^\ell, v_{i,j}^{**}\}) \leq n^9 + 2$, where u_{4n}^ℓ belongs to the edge selection gadget for i, j .*

Proof of Claim. Consider the first connector gadget of the edge selection gadget for i, j with vertices $\hat{v}_0, \hat{v}'_0, \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}'_3$ and sets A, B . Recall that $v_{i,j}^{**} \in B$ and hence we have that $d(\hat{v}_0, v_{i,j}^{**}) = \infty$. Furthermore, we have that $u_{4n}^\ell \notin B$ and hence $d(\hat{v}_0, u_{4n}^\ell) = 3$ (see duration nr. 3). By Claim 2 and the fact that $d(w^*, \hat{v}_0) = n^9$ we have that both edges incident with \hat{v}_0 have label n^9 . It follows that a fastest temporal path from \hat{v}_0 to u_{4n}^ℓ arrives at u_{4n}^ℓ at time $n^9 + 2$. Now assume for contradiction that $\lambda(\{u_{4n}^\ell, v_{i,j}^{**}\}) > n^9 + 2$. Then there exists a temporal walk from \hat{v}_0 to $v_{i,j}^{**}$ via u_{4n}^ℓ , a contradiction to $d(\hat{v}_0, v_{i,j}^{**}) = \infty$. \diamond

Claim 4. *For all $1 \leq i < j \leq k$ and $\ell \in [m]$ we have that $\lambda(\{x_\ell, v_{i,j}^*\}) = (i+j) \cdot (2n\ell^2 + 18n + 6)$, where x_ℓ belongs to the edge selection gadget for i, j .*

Proof of Claim. We first determine the label of $\{x_1, v_{i,j}^*\}$, where x_1 belongs to the edge selection gadget for i, j . Note that x_1 is connected to the alignment gadget. Let $w \in \hat{W}$ be the vertex of the alignment gadget that is a neighbor of x_1 . Since $d(w^*, x_1) = (20n + 6)(i + j)$ (see duration nr. 13) we have that $\lambda(\{w, x_1\}) = (20n + 6)(i + j)$.

First, assume that $\lambda(\{x_1, v_{i,j}^*\}) < (20n + 6)(i + j)$. Then there is a temporal path from $v_{i,j}^*$ to w via x_1 . However, we have that $d(x_{i,j}^*, w) = \infty$, a contradiction. Next, assume that $(20n + 6)(i + j) < \lambda(\{x_1, v_{i,j}^*\}) < n^9 + 2$. Then there is a temporal path from w to

$v_{i,j}$ via x_1 with duration strictly less than $n^9 - (20n + 6)(i + j) + 3$. However, we have that $d(w, v_{i,j}^*) = n^9 - (20n + 6)(i + j) + 3$ (see duration nr. 15), a contradiction. Finally, assume that $\lambda(\{x_1, v_{i,j}^*\}) \geq n^9 + 2$. Consider a fastest temporal path from x_1 to $v_{i,j}^{**}$. This temporal path cannot visit w as its first vertex, since from there it cannot continue. From this assumption and Claim 2 it follows, that the first edge of the temporal path has a label with value at least n^9 . However, by Claims 2 and 3 we have that all edges incident with $v_{i,j}^{**}$ have a label with value at most $n^9 + 2$. It follows that $d(x_1, v_{i,j}^{**}) \leq 3$, a contradiction to duration nr. 10.

We can conclude that $\lambda(\{x_1, v_{i,j}^*\}) = (20n + 6)(i + j)$. Now let $1 < \ell \leq m$. We have that $d(x_1, x_\ell) = 2n \cdot (i + j) \cdot (\ell^2 - 1) + 1$ (see duration nr. 9) which implies that $\lambda(\{x_\ell, v_{i,j}^*\}) \geq (i + j) \cdot (2n\ell^2 + 18n + 6)$. Assume that $(i + j) \cdot (2n\ell^2 + 18n + 6) < \lambda(\{x_\ell, v_{i,j}^*\}) \leq n^9 + 2$. Then the temporal path from x_1 to x_ℓ via $v_{i,j}^*$ is not a fastest temporal path from x_1 to x_ℓ . Again, we have that a fastest temporal path from x_1 to x_ℓ cannot visit w as its first vertex, since from there it cannot continue. By Claim 2, all other edges incident with x_1 (that is, all different from the one to $v_{i,j}^*$ and the one to w) have a label of at least n^9 and at most $n^9 + 2$. Similarly, by Claim 2 we have that all other edges incident with x_ℓ (that is, all different from the one to $v_{i,j}^*$) have a label of at least n^9 and at most $n^9 + 2$. It follows that any temporal path from x_1 to x_ℓ that visits $v_{i,j}^*$ as its first vertex has a duration strictly larger than $2n \cdot (i + j) \cdot (\ell^2 - 1) + 1$. Any temporal path from x_1 to x_ℓ that visits a vertex different from $v_{i,j}^*$ as its first vertex has duration of at most 3. In both cases, we have a contradiction to duration nr. 9. Lastly, assume that $\lambda(\{x_\ell, v_{i,j}^*\}) > n^9 + 2$. Consider a fastest temporal path from x_ℓ to $v_{i,j}^{**}$. Now this temporal path has duration at most 3 since by Claim 2 and the just made assumption all edges incident with x_ℓ have label at least n^9 whereas by Claims 2 and 3 all edges incident with $v_{i,j}^{**}$ have label at most $n^9 + 2$, a contradiction to duration nr. 10. \diamond

Claim 5. *For all $1 \leq i < j \leq k$ there exist a permutation $\sigma_{i,j} : [m] \rightarrow [m]$ such that for all $\ell \in [m]$ we have that $\lambda(\{u_{4n}^\ell, v_{i,j}^{**}\}) = (i + j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6) + 8n + 4$, where u_{4n}^ℓ belongs to the edge selection gadget for i, j .*

*Furthermore, a fastest temporal path from x_ℓ (of the edge selection gadget for i, j) to $v_{i,j}^{**}$ visits $v_{i,j}^*$ as its second vertex, and $u_{4n}^{\ell'}$ with $\sigma_{i,j}(\ell') = \ell$ (of the edge selection gadget for i, j) as its second last vertex.*

Proof of Claim. For every $\ell \in [m]$ we have that $d(x_\ell, v_{i,j}^{**}) = 8n + 5$ (see duration nr. 10), where x_ℓ belongs to the edge selection gadget for i, j . From Claims 2 and 4 follows that all edges incident with x_ℓ have a label of at least n^9 except the one to $v_{i,j}^*$ and, if $\ell = 1$, the edge connecting x_1 to the alignment gadget. In the latter case, no temporal path from x_1 from $v_{i,j}^{**}$ can continue to the neighbor of x_1 in the alignment gadget, since it cannot continue from there.

Now consider $v_{i,j}^{**}$. By Claims 2 and 3 we have that all edges incident with $v_{i,j}^{**}$ have a label of at most $n^9 + 2$. It follows that a fastest temporal path P from x_ℓ to $v_{i,j}^{**}$ has to visit $v_{i,j}^*$ after x_ℓ , since otherwise we have $d(x_\ell, v_{i,j}^{**}) \leq 2$, a contradiction to duration nr. 10.

Furthermore, we have by Claim 2 that all edges incident with $v_{i,j}^{**}$ have a label of at least n^9 except the ones incident to $u_{4n}^{\ell'}$ for $\ell' \in [m]$. By Claim 4 we have that $\lambda(\{x_\ell, v_{i,j}^*\}) \leq 4n^6$. It follows that a fastest temporal path from x_ℓ to $v_{i,j}^{**}$ has to visit $u_{4n}^{\ell'}$ for some $\ell' \in [m]$ as its second last vertex. Otherwise, we have $d(x_\ell, v_{i,j}^{**}) > 8n + 5$ (for sufficiently large n), a contradiction to duration nr. 10.

We can conclude that a fastest temporal path from x_ℓ to $v_{i,j}^{**}$ has to visit $v_{i,j}^*$ as its second vertex and $u_{4n}^{\ell'}$ for some $\ell' \in [m]$ as its second last vertex. Recall that in a temporal path, the difference between the labels of the first and last edge determine its duration (minus one). Hence, we have that $\lambda(\{u_{4n}^{\ell'}, v_{i,j}^{**}\}) - \lambda(\{x_\ell, v_{i,j}^*\}) + 1 = 8n + 5$. By Claim 4 we have that $\lambda(\{x_\ell, v_{i,j}^*\}) =$

$(i+j) \cdot (2n\ell^2 + 18n + 2)$. It follows that $\lambda(\{u_{4n}^{\ell'}, v_{i,j}^{**}\}) = (i+j) \cdot (2n\ell^2 + 18n + 6) + 8n + 4$. We set $\sigma_{i,j}(\ell') = \ell$.

Finally, we show that $\sigma_{i,j}$ is a permutation on $[m]$. Assume for contradiction that there are $\ell, \ell' \in [m]$ with $\ell \neq \ell'$ such that $\sigma_{i,j}(\ell) = \sigma_{i,j}(\ell')$. Then we have that $\lambda(\{u_{4n}^{\ell}, v_{i,j}^{**}\}) = \lambda(\{u_{4n}^{\ell'}, v_{i,j}^{**}\})$. However, by Claim 4 we have that all edges from $v_{i,j}^*$ to a vertex in $X_{i,j}$ have distinct labels. Furthermore, we argued above that every fastest path from a vertex in $X_{i,j}$ to $v_{i,j}^{**}$ visits $v_{i,j}^*$ as its second vertex and a vertex from the set $\{u_{4n}^{\ell''} : \ell'' \in [m]\}$ as its second last vertex. Since for all $x_{\ell''}$ with $\ell'' \in [m]$ we have that $d(x_{\ell''}, v_{i,j}^{**}) = 8n + 5$ (see duration nr. 10), we must have that all edges from vertices in $\{u_{4n}^{\ell''} : \ell'' \in [m]\}$ to $v_{i,j}^{**}$ must have distinct labels. Hence, we have a contradiction and can conclude that $\sigma_{i,j}$ is indeed a permutation. \diamond

For all $1 \leq i < j \leq k$, let $\sigma_{i,j}$ be the permutation on $[m]$ as defined in Claim 5. We call $\sigma_{i,j}$ the *permutation of color combination i, j* . Now we have enough information to define a set of vertices of H that form a multicolored clique. To this end, consider the following set X of edges from H .

$$X = \{e_{\ell}^{i,j} \in F_{i,j} : \sigma_{i,j}(\ell) = 1\}$$

We claim that $\bigcup_{e \in X} e$ forms a multicolored clique in H . From now on, denote $\{e_{i,j}\} = X \cap F_{i,j}$. We show that for all $i \in [k]$ we have that $|\bigcap_{1 \leq j < i} e_{j,i} \cap \bigcap_{i < j \leq k} e_{i,j}| = 1$, that is, for every color i , all edges of a color combination involving i have the same vertex of color i as endpoint. This implies that $\bigcup_{e \in X} e$ is a multicolored clique in H .

Before we proceed, we show some further properties of λ . First, let us focus on the labels on edges of the edge selection gadgets.

Claim 6. *For all $1 \leq i < j \leq k$, $\ell \in [m]$, and $\ell' \in [4n]$ we have that $\lambda(\{u_{\ell'-1}^{\ell}, u_{\ell'}^{\ell}\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6) + 2\ell' + 2$, where $u_{\ell'-1}^{\ell}$ and $u_{\ell'}^{\ell}$ belong to the edge selection gadget for i, j and $\sigma_{i,j}$ is the permutation of color combination i, j .*

Proof of Claim. Let $1 \leq i < j \leq k$ and $\ell \in [m]$. By Claim 5 we know that a fastest temporal path from $x_{\sigma_{i,j}(\ell)}$ (of the edge selection gadget for i, j) to $v_{i,j}^{**}$ visits $v_{i,j}^*$ as its second vertex, and u_{4n}^{ℓ} (of the edge selection gadget for i, j) as its second last vertex. Furthermore, by Claim 4 we have that $\lambda(\{x_{\sigma_{i,j}(\ell)}, v_{i,j}^*\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 2)$ and by Claim 5 we have that $\lambda(\{u_{4n}^{\ell}, v_{i,j}^{**}\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 2) + 8n + 4$. It follows that there exist a temporal path P from $v_{i,j}^*$ to u_{4n}^{ℓ} that starts at $v_{i,j}^*$ later than $(i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6)$ and arrives at u_{4n}^{ℓ} earlier than $(i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6) + 8n + 4$. Hence, the temporal path P has duration at most $8n + 3$.

We investigate the temporal path P from its destination u_{4n}^{ℓ} back to its start vertex $v_{i,j}^*$. Consider the neighbors of u_{4n}^{ℓ} that are different from $v_{i,j}^{**}$. By Claim 2 we have that all edges from u_{4n}^{ℓ} to neighbors of u_{4n}^{ℓ} that are vertices of connector gadgets have a label of at least n^9 . Hence, P does not visit any of those neighbors. Next, consider neighbors of u_{4n}^{ℓ} in verification gadgets. Assume u_{4n}^{ℓ} has a neighbor in the verification gadget of color i' for some $i' \in [k]$. Then this neighbor is vertex $b_{5n}^{i',j,\ell}$. Note that if P visits $b_{5n}^{i',j,\ell}$, then it also visits all of $\{b_{\ell'}^{i',j,\ell} : \ell' \in [5n]\}$, since all these vertices have degree two. Now consider the second connector gadget of a verification gadget i' with sets A, B , we have that all vertices $\{b_{\ell'}^{i',j,\ell} : \ell' \in [5n]\}$ are contained in A and are not contained in B . Hence, we have that all non-adjacent pairs of vertices in $\{b_{\ell'}^{i',j,\ell} : \ell' \in [5n]\}$ are on duration 3 apart (see duration nr. 4), and that $|\lambda(\{b_{\ell'}^{i',j,\ell}, b_{\ell'+1}^{i',j,\ell}\}) - \lambda(\{b_{\ell'+1}^{i',j,\ell}, b_{\ell'+2}^{i',j,\ell}\})| \geq 2$ for all $\ell' \in [5n-2]$. It follows that P would have a duration larger than $8n + 3$. We can conclude that P does not visit $b_{5n}^{i',j,\ell}$. It follows that P visits u_{4n-1}^{ℓ} . Here, we can make an analogous investigation. Additionally, we have to consider the case that P

visits a neighbor of u_{4n-1}^ℓ in verification gadget of color i' for some $i' \in [k]$ that is vertex $a_{5n}^{i',j,\ell}$. However, we can exclude this by a similar argument as above.

By repeating the above arguments, we can conclude that P visits (exactly) all vertices in $\{u_{\ell'}^\ell : 0 \leq \ell' \leq 4n\}$ and $v_{i,j}^*$. Consider the second connector gadget of the edge selection gadget of i, j with set A and B . Note that all vertices visited by P are contained in $A \setminus B$. It follows that all pairs of non-adjacent vertices visited by P are on duration 3 apart (see duration nr. 4). In particular, we have $d(u_{\ell'-1}^\ell, u_{\ell'+1}^\ell) = 3$ for all $\ell' \in [4n-1]$ and $d(v_{i,j}^*, u_1^\ell) = 3$ (see duration nr. 4). It follows that for every $\ell' \in [4n-1]$ we have that $\lambda(\{u_{\ell'}^\ell, u_{\ell'+1}^\ell\}) - \lambda(\{u_{\ell'-1}^\ell, u_{\ell'}^\ell\}) \geq 2$ and $\lambda(\{u_1^\ell, u_2^\ell\}) - \lambda(\{v_{i,j}^*, u_1^\ell\}) \geq 2$.

By investigating the sets A, B of the first connector gadget of the edge selection gadget of i, j , we get that $d(x_{\sigma_{i,j}(\ell)}, u_1^\ell) = 3$ (see duration nr. 4) and hence $\lambda(\{v_{i,j}^*, u_1^\ell\}) - \lambda(\{x_{\sigma_{i,j}(\ell)}, v_{i,j}^*\}) \geq 2$. Furthermore, we get that $d(u_{4n-1}^\ell, v_{i,j}^{**}) = 3$ (see duration nr. 4) and hence $\lambda(\{v_{i,j}^{**}, u_{4n}^\ell\}) - \lambda(\{u_{4n-1}^\ell, u_{4n}^\ell\}) \geq 2$. Considering that P visits $4n+2$ vertices, we have that all mentioned inequalities of differences of labels have to be equalities, otherwise P has a duration larger than $8n+3$ or we have that $\lambda(\{v_{i,j}^*, u_1^\ell\}) - \lambda(\{x_{\sigma_{i,j}(\ell)}, v_{i,j}^*\}) < 2$ or $\lambda(\{v_{i,j}^{**}, u_{4n}^\ell\}) - \lambda(\{u_{4n-1}^\ell, u_{4n}^\ell\}) < 2$. Since by Claims 4 and 5 the labels $\lambda(\{x_{\sigma_{i,j}(\ell)}, v_{i,j}^*\})$ and $\lambda(\{v_{i,j}^{**}, u_{4n}^\ell\})$ are determined, then also all labels of edges traversed by P are determined and the claim follows. \diamond

Next, we investigate the labels of the verification gadgets.

Claim 7. *For all $i \in [k]$ we have that $\lambda(\{y^i, v_0^i\}) = n^8$.*

Proof of Claim. Let $w \in \hat{W}$ denote the neighbor of y^i in the alignment gadget. Note that we have $d(w^*, y^i) = n^8 - 1$. It follows that $\lambda(\{w, y^i\}) = n^8 - 1$ (see duration nr. 16). Furthermore, we have that $d(w, v_0^i) = 2$ (see duration nr. 16) and note that y^i has degree 2. It follows that $\lambda(\{y^i, v_0^i\}) = n^8$. \diamond

Claim 8. *For all $1 < i \leq k$ and all $\ell \in [m]$ we have that $\lambda(\{v_0^i, a_1^{i,1,\ell}\}) \leq n^8$ or $\lambda(\{v_0^i, a_1^{i,1,\ell}\}) \geq n^9 + 2$. For $i = 1$ we have that $\lambda(\{v_0^i, \hat{u}_1^i\}) \leq n^8$ or $\lambda(\{v_0^i, \hat{u}_1^i\}) \geq n^9 + 2$.*

Proof of Claim. Let $1 < i \leq k$ and $\ell \in [m]$. Assume that $n^8 < \lambda(\{v_0^i, a_1^{i,1,\ell}\}) < n^9 + 2$. Then, since by Claim 7 we have $\lambda(\{y^i, v_0^i\}) = n^8$, there is a temporal path from w^* to $a_1^{i,1,\ell}$ via v_0^i that arrives at $a_1^{i,1,\ell}$ strictly earlier than $n^9 + 2$. However, we have $d(w^*, a_1^{i,1,\ell}) = n^9 + 2$ (see duration nr. 15), a contradiction. The argument for the case where $i = 1$ is analogous. \diamond

Claim 9. *For all $1 \leq i < k$ and all $\ell \in [m]$ we have that $\lambda(\{v_k^i, b_1^{i,k,\ell}\}) \leq n^9 + 2$. For $i = k$ we have that $\lambda(\{v_k^i, \hat{u}_{13n+1}^i\}) \leq n^9 + 2$.*

Proof of Claim. Let $1 \leq i < k$ and $\ell \in [m]$. Consider the first connector gadget of verification gadget for color i with vertices $\hat{v}_0, \hat{v}_0', \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_3'$ and sets A, B . Recall that $v_k^i \in B$ and hence we have that $d(\hat{v}_0, v_k^i) = \infty$. Furthermore, we have that $b_1^{i,k,\ell} \notin B$ and hence $d(\hat{v}_0, b_1^{i,k,\ell}) = 3$ (see duration nr. 3). By Claim 2 and the fact that $d(w^*, \hat{v}_0) = n^9$ (see duration nr. 19) we have that both edges incident with \hat{v}_0 have label n^9 . It follows that a fastest temporal path from \hat{v}_0 to $b_1^{i,k,\ell}$ arrives at $b_1^{i,k,\ell}$ at time $n^9 + 2$. Now assume for contradiction that $\lambda(\{v_k^i, b_1^{i,k,\ell}\}) > n^9 + 2$. Then there exists a temporal walk from \hat{v}_0 to v_k^i via $b_1^{i,k,\ell}$, a contradiction to $d(\hat{v}_0, v_k^i) = \infty$. The argument for the case where $i = k$ is analogous. \diamond

Now we are ready to prove for all $i \in [k]$ that $|(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j})| = 1$. Assume for contradiction that for some color $i \in [k]$ we have that $|(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j})| \neq 1$. Consider the verification gadget for color i . Recall that $d(v_0^i, v_k^i) = k(20n+6) + 6n - 1$ (see duration nr. 12). Let P be a fastest temporal path from v_0^i to v_k^i . We first argue that P cannot visit any vertex of a connector gadget or the alignment gadget.

Claim 10. *Let $i \in [k]$. Let P be a fastest temporal path from v_0^i to v_k^i . Then P does not visit any vertex of a connector gadget.*

Proof of Claim. Assume for contradiction that P visits a vertex of a connector gadget. Then by Claim 2 we have that the arrival time of P is at least n^9 . By Claim 2 and Claim 9 we have that the arrival time of P is at most $n^9 + 2$. This means that the second vertex visited by P cannot be a vertex from a connector gadget, because by Claim 2 this would imply $d(v_0^i, v_k^i) \leq 2$, a contradiction to duration nr. 12. Now we can deduce with Claim 8 that P must have a starting time of at most n^8 . It follows that the arrival time of P must be smaller than n^9 , a contradiction to the assumption that P visits a vertex of a connector gadget. \diamond

Claim 11. *Let $i \in [k]$. Let P be a fastest temporal path from v_0^i to v_k^i . Then P does not visit any vertex of the alignment gadget.*

Proof of Claim. Note that P starts outside the alignment gadget. This means that if P visits a vertex of the alignment gadget, then the first vertex of the alignment gadget visited by P is a neighbor of w^* . However, these vertices have degree two and the edge to w^* has label one. It follows that P cannot continue from the vertex of the alignment gadget, a contradiction. \diamond

It follows that the second vertex visited by P is a vertex $a_1^{i,1,\ell}$ for some $\ell \in [m]$ or vertex \hat{u}_1^i if $i = 1$. In the former case, P has to follow the path segment consisting of vertices in $\{a_{\ell'}^{i,1,\ell} : \ell' \in [5n]\}$ until it reaches the edge selection gadget of color combination $1, i$. From there it can reach vertex v_1^i by traversing some path segment consisting of vertices $\{b_{\ell''}^{i,1,\ell'} : \ell'' \in [5n]\}$ for some $\ell' \in [m]$. Alternatively, it can reach vertex v_{i-1}^1 or v_i^1 by traversing some path segment consisting of vertices $\{a_{\ell''}^{1,i,\ell'} : \ell'' \in [5n]\}$ for some $\ell' \in [m]$ or $\{b_{\ell''}^{1,i,\ell'} : \ell'' \in [5n]\}$ for some $\ell' \in [m]$, respectively. In the latter case ($i = 1$), the temporal path P has to follow the path segment consisting of vertices in $\{\hat{u}_\ell^i : \ell \in [13n + 1]\}$ until it reaches v_1^i . More generally, we can make the following observation.

Claim 12. *Let $i, j \in \{0, 1, \dots, k\}$. Let P be a temporal path starting at v_j^i and visiting at most $13n + 1$ vertices and no vertex of a connector gadget or the alignment gadget. Then P cannot visit vertices in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\} \setminus \{v_{j-1}^i, v_j^i, v_{j+1}^i, v_{i-1}^j, v_i^j, v_{i-1}^{j+1}, v_i^{j+1}\}$.*

Proof of Claim. Consider the edge selection gadget of color combination i', j' for some $i', j' \in [k]$ and let $u_{\ell'}^\ell$ be a vertex of that gadget. Disregarding connections via connector gadgets and the alignment gadget, we have that $u_{\ell'}^\ell$ is (potentially) connected to the verification gadget for color i' and the verification gadget of color j' . More specifically, by construction of G , we have that $u_{\ell'}^\ell$ is potentially connected to

- vertex $v_{j'-1}^{i'}$ by a path along vertices $\{a_{\ell''}^{i',j',\ell} : \ell'' \in [5n]\}$,
- vertex $v_{j'}^{i'}$ by a path along vertices $\{b_{\ell''}^{i',j',\ell} : \ell'' \in [5n]\}$,
- vertex $v_{i'-1}^{j'}$ by a path along vertices $\{a_{\ell''}^{j',i',\ell} : \ell'' \in [5n]\}$, and
- vertex $v_{i'}^{j'}$ by a path along vertices $\{b_{\ell''}^{j',i',\ell} : \ell'' \in [5n]\}$.

Furthermore, by construction of G , we have that the duration of a fastest path from $u_{\ell'}^\ell$ to any $v_{j''}^{i''}$ with $i'', j'' \in \{0, 1, \dots, k\}$ not mentioned above is at least $10n$ (disregarding edges incident with connector gadgets or the alignment gadget).

Now consider v_j^i and assume $i < j$ ($i > j$). This vertex is (if $j \neq i - 1$ and $j \neq k$) connected to some vertex $u_{\ell'}^\ell$ in the edge selection gadget for color combination $i, j + 1$ ($j + 1, i$) via a path

along vertices $\{a_{\ell''}^{i,j,\ell} : \ell'' \in [5n]\}$. Furthermore, v_j^i is (if $j \neq 0$ and $j \neq i$) connected to some vertex $u_{\ell''}^{\ell'}$ in the edge selection gadget for color combination i, j (j, i) via a path along vertices $\{b_{\ell''}^{i,j,\ell'} : \ell'' \in [5n]\}$.

We can conclude that v_j^i can reach a vertex $u_{\ell'}^{\ell}$ of the edge selection gadget for $i, j+1$ (or $j+1, i$) and a vertex $u_{\ell''}^{\ell'}$ of the edge selection gadget for color combination i, j (or j, i), each along paths of length at least $5n$. From $u_{\ell'}^{\ell}$ and $u_{\ell''}^{\ell'}$ we have that any other vertex of the edge selection gadget for $i, j+1$ (or $j+1, i$) and the edge selection gadget for color combination i, j (or j, i), respectively, can be reached by a path of length at most $3n$. Together with the observation made in the beginning of the proof, we can conclude that v_j^i can potentially reach any vertex in $\{v_{j-1}^i, v_j^i, v_{j+1}^i, v_{i-1}^j, v_i^j, v_{i+1}^j, v_{i-1}^{j+1}, v_i^{j+1}\}$ by a path that visits at most $13n+1$ vertices.

Lastly, consider the case that $j = i-1$ or $j = i$. Then we have that v_{i-1}^i and v_i^i are connected via a path inside the verification gadget for color i , visiting the $13n+1$ vertices in $\{\hat{u}_\ell^i : \ell \in [13n+1]\}$. The claim follows. \diamond

Furthermore, we can make the following observation on the duration of the temporal paths characterized in Claim 12.

Claim 13. *Let $i, j \in \{0, 1, \dots, k\}$. Let P be a temporal path from v_j^i to a vertex in $\{v_{j-1}^i, v_j^i, v_{j+1}^i, v_{i-1}^j, v_i^j, v_{i+1}^j, v_{i-1}^{j+1}, v_i^{j+1}\}$ and visiting no vertex of a connector gadget or the alignment gadget. Then P has duration at least $20n$.*

Proof of Claim. As argued in the proof of Claim 12, a temporal path P from v_j^i to a vertex in $\{v_{j-1}^i, v_j^i, v_{j+1}^i, v_{i-1}^j, v_i^j, v_{i+1}^j, v_{i-1}^{j+1}, v_i^{j+1}\}$ has to either traverse two segments of $5n$ vertices in $\{a_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ or $\{b_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ for some $\ell \in [m]$ and $i', j' \in \{i-1, i, j, j+1\}$ or a segment of the $13n+1$ vertices in $\{\hat{u}_\ell^i : \ell \in [13n+1]\}$. We analyze the former case first.

Consider the second connector gadget of a verification gadget i' with sets A, B , we have that all vertices $\{a_{\ell'}^{i',j',\ell} : \ell' \in [5n], j' \in [k] \setminus \{i'\}\} \cup \{b_{\ell'}^{i',j',\ell} : \ell' \in [5n], j' \in [k] \setminus \{i'\}\}$ are contained in A and are not contained in B . It follows that all non-adjacent pairs of vertices in $\{a_{\ell'}^{i',j',\ell} : \ell' \in [5n], j' \in [k] \setminus \{i'\}\} \cup \{b_{\ell'}^{i',j',\ell} : \ell' \in [5n], j' \in [k] \setminus \{i'\}\}$ are on duration 3 apart (see duration nr. 4). It follows that $|\lambda(\{a_{\ell'}^{i',j',\ell}, a_{\ell'+1}^{i',j',\ell}\}) - \lambda(\{a_{\ell'+1}^{i',j',\ell}, a_{\ell'+2}^{i',j',\ell}\})| \geq 2$ for all $\ell' \in [5n-2]$ and $j' \in [k] \setminus \{i'\}$. Analogously, we have that $|\lambda(\{b_{\ell'}^{i',j',\ell}, b_{\ell'+1}^{i',j',\ell}\}) - \lambda(\{b_{\ell'+1}^{i',j',\ell}, b_{\ell'+2}^{i',j',\ell}\})| \geq 2$ for all $\ell' \in [5n-2]$ and $j' \in [k] \setminus \{i'\}$. It follows that two segments of $5n$ vertices in $\{a_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ or $\{b_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ for some $\ell \in [m]$ and $i', j' \in \{i-1, i, j, j+1\}$ traversed by P both have duration $10n$ and hence P has duration at least $20n$.

In the latter case, where P traverses a segment of the $13n+1$ vertices in $\{\hat{u}_\ell^i : \ell \in [13n+1]\}$, we can make an analogous argument, since all vertices in $\{\hat{u}_\ell^i : \ell \in [13n+1]\}$ are contained in the set A of the second connector gadget of the verification gadget of color i but not in the set B of that connector gadget. \diamond

Recall that P denotes a fastest temporal path from v_0^i to v_k^i and that $d(v_0^i, v_k^i) = k(20n+6) + 6n - 1$ (see duration nr. 12). By Claims 10 to 12 we have that P needs to visit at least one vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\} \setminus \{v_0^i, v_k^i\}$. Next, we analyze which vertices in this set are visited by P .

Claim 14. *Let $i \in [k]$. Let P be a fastest temporal path from v_0^i to v_k^i . Then P visits all vertices in $\{v_j^j : 0 \leq j \leq k\}$ and no vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\} \setminus \{v_j^j : 0 \leq j \leq k\}$. Furthermore, P visits the vertices in order $v_0^i, v_1^i, v_2^i, \dots, v_{k-1}^i, v_k^i$.*

Proof of Claim. Let $X \subseteq \{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ denote the set of vertices in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ that are visited by P . By Claims 12 and 13 we have that $|X| \leq k+1$, since

otherwise the duration of P would be at least $20n(k+1) > k(20n+6) + 6n - 1$, a contradiction to duration nr. 12.

To prove the claim, we use the notion of a *potential* p^i with respect to i of a vertex $v_j^{i'}$. We say that the first potential of vertex $v_j^{i'}$ with respect to i is $p^i(v_j^{i'}) = i' + j - i$. The temporal path P starts at vertex v_0^i with $p^i(v_0^i) = 0$, and ends at vertex v_k^i with $p^i(v_k^i) = k$.

Assume the path P is at some vertex $v_j^{i'}$ with $p_1^i(v_j^{i'}) = i' + j - i$. By Claim 12 we have that the next vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ visited by P is some $v_{j''}^{i''} \in \{v_{j-1}^{i'}, v_j^{i'}, v_{j+1}^{i'}, v_{i'-1}^j, v_i^j, v_{i'-1}^{j+1}, v_{i'}^{j+1}\}$. We can observe that $|p^i(v_j^{i'}) - p^i(v_{j''}^{i''})| \leq 1$, that is, the first potential changes at most by one when P goes from one vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ to the next one. Since $|X| \leq k+1$ we and $p^i(v_k^i) - p^i(v_0^i) = k$ have that the potential has to increase by exactly one every time P goes from one vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ to the next one. We can conclude that $|X| = k+1$. Furthermore, we have that if the path P is at some vertex $v_j^{i'}$, the next vertex in $\{v_{j'}^{i'} : i', j' \in \{0, 1, \dots, k\}\}$ visited by P is either $v_{j+1}^{i'}$ or $v_{i'}^{j+1}$.

By Claim 13 we have that the temporal path segments from $v_j^{i'}$ to $v_{j+1}^{i'}$ and $v_{i'}^{j+1}$, respectively, have duration at least $20n$. However, for the temporal path from $v_j^{i'}$ to $v_{i'}^{j+1}$ (with $j \neq i' - 1$) we can obtain a larger lower bound. As argued in the proof of Claim 11, a temporal path segment from $v_j^{i'}$ to $v_{i'}^{j+1}$ has to either traverse two segments of $5n$ vertices in $\{a_{\ell'}^{i', j+1, \ell} : \ell' \in [5n]\}$ or $\{b_{\ell'}^{i', j+1, \ell} : \ell' \in [5n]\}$ for some $\ell' \in [m]$ and $i', j' \in \{i-1, i, j, j+1\}$. More precisely, the temporal path segment has to traverse part of the edge selection gadget of color combination $i', j+1$. To this end, it traverses the $5n$ vertices in $\{a_{\ell''}^{i', j+1, \ell} : \ell'' \in [5n]\}$ for some $\ell' \in [m]$. Then it traverses some vertices in the edge selection gadget, and then it traverses the $5n$ vertices in $\{b_{\ell''}^{j+1, i', \ell'} : \ell'' \in [5n]\}$ for some $\ell' \in [m]$.

By construction of G , the first vertex of the edge selection gadget visited by the path segment (after traversing vertices in $\{a_{\ell''}^{i', j+1, \ell} : \ell'' \in [5n]\}$) is some vertex $u_{\ell''}^{\ell'}$ with $\ell'' \in \{0, 1, \dots, 4n\}$. The last vertex of the edge selection gadget visited by the path segment is (before traversing the vertices in $\{b_{\ell''}^{j+1, i', \ell'} : \ell'' \in [5n]\}$) some vertex $u_{\ell''}^{\ell'}$ with $\ell'' \in \{0, 1, \dots, 4n\}$. By construction of G , the duration of a fastest path between $u_{\ell''}^{\ell'}$ and $u_{\ell''}^{\ell'}$ (in G) is at least $3n$. Investigating the second connector gadget of the edge selection gadget for $i', j+1$ we can see that a temporal path from $u_{\ell''}^{\ell'}$ and $u_{\ell''}^{\ell'}$ has duration at least $6n$.

It follows that the temporal path segment from $v_j^{i'}$ to $v_{i'}^{j+1}$ (with $j \neq i' - 1$) has duration at least $26n$. Furthermore, recall that P starts at v_0^i and ends at v_k^i . We have that if P contains a path segment from some $v_j^{i'}$ to $v_{i'}^{j+1}$ some (with $j \neq i' - 1$), then P visits a vertex $v_{j''}^{i''}$ with $i'' \neq i$. Hence, it needs to contain at least one additional path segment from some $v_j^{i'}$ to some $v_{i'}^{j+1}$ (with $j \neq i - 1$). However, then we have that the duration of P is at least $20kn + 12n > k(20n+6) + 6n - 1$, a contradiction to duration nr. 12.

We can conclude that P only contains temporal path segments from v_{j-1}^i to v_j^i for $j \in [k]$ and the claim follows. \diamond

Now we have by Claims 12 and 14 that we can divide P into k segments, the subpaths from v_{j-1}^i to v_j^i for $j \in [k]$. We show that all subpaths except the one from v_{i-1}^i to v_i^i have duration $20n + 5$. The subpath from v_{i-1}^i to v_i^i has duration $26n + 5$.

Claim 15. *Let $i \in [k]$ and $j \in [k] \setminus \{i\}$. Let P be a temporal path from v_{j-1}^i to v_j^i that does not visit vertices from connector gadgets and the alignment gadget. If P has duration at most $20n + 5$, then it visits exactly two vertices $u_{\ell'-1}^{\ell}, u_{\ell'}^{\ell}$ with $\ell \in [m]$, and $\ell' \in [4n]$ of the edge selection gadget for color combination i, j (or j, i).*

Proof of Claim. By the construction of G (and as also argued in the proofs of Claims 12

and 13), a temporal path P with duration at most $20n + 5$ that does not visit vertices from connector gadgets and the alignment gadget from v_{j-1}^i to v_j^i has to first traverse a segment of $5n$ vertices in $\{a_{\ell'}^{i,j-1,\ell} : \ell' \in [5n]\}$ and then a segment of $5n$ vertices $\{b_{\ell'}^{i,j,\ell} : \ell' \in [5n]\}$ for some $\ell \in [m]$. By construction of G , the two vertices visited in the edge selection gadget for color combination i, j (or j, i) are $u_{\ell'-1}^\ell$ and $u_{\ell'}^\ell$ for some $\ell' \in [4n]$. By inspecting the connector gadgets in an analogous way as in the proof of Claim 13 we can deduce that all consecutive edges traversed by P must have labels that differ by at least 2. It follows that if all consecutive edges have labels that differ by exactly two, then P has duration $20n + 5$. \diamond

Claim 16. *Let $i \in [k]$. Let P be a temporal path from v_{i-1}^i to v_i^i that does not visit vertices from connector gadgets and the alignment gadget. Then P has duration at least $26n + 5$.*

Proof of Claim. By construction of G we have that v_{i-1}^i and v_i^i are connected via a path inside the verification gadget for color i , visiting the $13n + 1$ vertices in $\{\hat{u}_\ell^i : \ell \in [13n + 1]\}$. Assume P follows this path. By inspecting the connector gadgets of the verification gadget of color i , we can see that all consecutive edges traversed by P must have labels that differ by at least two. It follows that P has duration at least $26n + 5$. By construction of G we have that if P does not follow the vertices in $\{\hat{u}_\ell^i : \ell \in [13n + 1]\}$ it has to visit at least three different edge selection gadgets: The one of color combination $i - 1, i$, then one of $i - 1, i + 1$, and then the one of $i, i + 1$. It follows that P needs to visit at least four segments of length $5n$ composed of vertices $\{a_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ or $\{b_{\ell'}^{i',j',\ell} : \ell' \in [5n]\}$ for some $\ell \in [m]$ and $i', j' \in [k]$. By inspecting the connector gadgets of the verification gadgets we know that it takes at least $10n$ time steps to traverse such a segment. Hence, the duration of P is at least $40n$. \diamond

Furthermore, we need the following observation which is relevant when we try to connect the above-mentioned segments to a temporal path.

Claim 17. *Let $i \in [k]$ and $0 \leq j \leq k$. The absolute difference of labels of any two different edges incident with v_j^i is at least two.*

Proof of Claim. This follows by inspecting the connector gadgets of the verification gadget of color i . \diamond

From Claims 10, 11 and 14 to 17 we get that a fastest temporal path P from v_0^i to v_k^i has the following properties.

1. The path P can be segmented into temporal path segments P_j from v_{j-1}^i to v_j^i for $j \in [k] \setminus \{i\}$ such that P_j is a temporal path from v_{j-1}^i to v_j^i that does not visit vertices from connector gadgets and the alignment gadget and has duration $20n + 5$.
2. The segment of P from v_{i-1}^i to v_i^i has duration $26n + 5$.
3. The path P dwells at each vertex v_j^i with $j \in [k - 1]$ for exactly two time steps, that is, the absolute difference of the labels on the edges incident with v_j^i that are traversed by P is exactly two.

If any of the properties does not hold, then we can observe that $d(v_0^i, v_k^i) > k(20n + 6) + 6n - 1$ would follow, a contradiction to duration nr. 12.

Now assume $i \in [k]$ and $j \in [k] \setminus \{i\}$ and consider a fastest temporal path P_j from v_{j-1}^i to v_j^i that does not visit vertices from connector gadgets and the alignment gadget and a fastest temporal path P_{j+1} from v_j^i to v_{j+1}^i that does not visit vertices from connector gadgets and the alignment gadget. By Claim 15 we know that P_j visits vertices $u_{\ell'-1}^\ell, u_{\ell'}^\ell$ with $\ell \in [m]$, and $\ell' \in [4n]$ of the edge selection gadget for color combination i, j . By Claim 6 we have that $\lambda(\{u_{\ell'-1}^\ell, u_{\ell'}^\ell\}) = (i + j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6) + 2\ell' + 2$, where $\sigma_{i,j}$ is the permutation of color combination i, j (or j, i). Analogously, we have by Claim 15 that P_{j+1} visits vertices $u_{\ell''-1}^{\ell''}, u_{\ell''}^{\ell''}$

with $\ell'' \in [m]$, and $\ell''' \in [4n]$ of the edge selection gadget for color combination $i, j+1$. By Claim 6 we have that $\lambda(\{u_{\ell''-1}^{\ell''}, u_{\ell'''}^{\ell'''}\}) = (i+j+1) \cdot (2n \cdot (\sigma_{i,j+1}(\ell''))^2 + 18n + 6) + 2\ell''' + 2$, where $\sigma_{i,j+1}$ is the permutation of color combination $i, j+1$ (or $j+1, i$). We have that

$$\begin{aligned} & \lambda(\{u_{\ell''-1}^{\ell''}, u_{\ell'''}^{\ell'''}\}) - \lambda(\{u_{\ell'-1}^{\ell'}, u_{\ell'}^{\ell'}\}) = \\ & (i+j+1) \cdot (2n \cdot (\sigma_{i,j+1}(\ell''))^2 + 18n + 6) + 2\ell''' + 2 \\ & - ((i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell))^2 + 18n + 6) + 2\ell' + 2) = \\ & (i+j+1) \cdot 2n \cdot (\sigma_{i,j+1}(\ell''))^2 - (i+j) \cdot 2n \cdot (\sigma_{i,j}(\ell))^2 + 2(\ell''' - \ell') + 18n + 6 \end{aligned}$$

By the arguments made before we also have that if P_j and P_{j+1} are both path segments of P , then

$$\lambda(\{u_{\ell''-1}^{\ell''}, u_{\ell'''}^{\ell'''}\}) - \lambda(\{u_{\ell'-1}^{\ell'}, u_{\ell'}^{\ell'}\}) = 20n + 6.$$

It follows that

$$(i+j+1) \cdot 2n \cdot (\sigma_{i,j+1}(\ell''))^2 - (i+j) \cdot 2n \cdot (\sigma_{i,j}(\ell))^2 + 2(\ell''' - \ell') = 2n.$$

Assume that $\sigma_{i,j}(\ell) \neq \sigma_{i,j+1}(\ell'')$, then we have that $(i+j+1) \cdot 2n \cdot (\sigma_{i,j+1}(\ell''))^2 - (i+j) \cdot 2n \cdot (\sigma_{i,j}(\ell))^2 < 6n$ or $(i+j+1) \cdot 2n \cdot (\sigma_{i,j+1}(\ell''))^2 - (i+j) \cdot 2n \cdot (\sigma_{i,j}(\ell))^2 > 10n$, since $|(\sigma_{i,j}(\ell''))^2 - (\sigma_{i,j}(\ell))^2| \geq 3$ and $(i+j) \geq 3$. However, we have that $\ell', \ell''' \in [4n]$ and hence $|2(\ell''' - \ell')| < 8n$. We can conclude that $\sigma_{i,j}(\ell) = \sigma_{i,j+1}(\ell'')$. In this case we have that $(i+j+1) \cdot 2n \cdot (\sigma_{i,j+1}(\ell''))^2 - (i+j) \cdot 2n \cdot (\sigma_{i,j}(\ell))^2 = 2n \cdot (\sigma_{i,j}(\ell'))^2$. It follows that $2n(\sigma_{i,j}(\ell))^2 - 2(\ell''' - \ell') = 2n$. Again, since $|2(\ell''' - \ell')| < 8n$, we have that $\sigma_{i,j}(\ell) = 1$ and in turn this implies that $\ell' = \ell'''$.

Note that if $i = 1$ or $i = k$ we can already conclude that $|(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j})| = 1$. By construction of G we have that for all $j \in [k] \setminus \{i\}$ that v_{j-1}^i and v_j^i are connected to $u_{\ell'-1}^{\ell'}$ and $u_{\ell'}^{\ell'}$ of the edge selection gadget of color combination i, j (or j, i), respectively, via paths using vertices $\{a_{\ell''}^{i,j,\ell} : \ell'' \in [5n]\}$ and $\{b_{\ell''}^{i,j,\ell} : \ell'' \in [5n]\}$, respectively, if the vertex $w_{\ell'}^i \in W_i$ (for $i = k$, or vertex $w_{\ell'-3n}^i \in W_i$ for $i = 1$) is incident with edge $e_{\ell'}^{i,j} \in F_{i,j}$. Note that since $\sigma_{i,j}(\ell) = 1$ we have that $e_{\ell'}^{i,j} \in X$. Since ℓ' is independent from ℓ and j , it follows that $(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j}) = \{w_{\ell'}^i\}$ for $i = k$ and $(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j}) = \{w_{\ell'-3n}^i\}$ for $i = 1$.

Assume now that $1 \neq i \neq k$. By Claim 16 we know that the duration of the path segment P_i from v_{i-1}^i to v_i^i is $26n + 5$. Consider the path segment P^* from v_{i-2}^i to v_{i+1}^i . By the arguments above we know that P^* visits vertices $u_{\ell'-1}^{\ell'}, u_{\ell'}^{\ell'}$ with $\sigma_{i-1,i}(\ell) = 1$, and $\ell' \in [4n]$ of the edge selection gadget for color combination $i-1, i$ and afterwards P^* visits vertices $u_{\ell''-1}^{\ell''}, u_{\ell''}^{\ell''}$ with $\sigma_{i,i+1}(\ell'') = 1$, and $\ell'' \in [4n]$ of the edge selection gadget for color combination $i, i+1$. By analogous arguments as above and the fact that the duration of P_i is $26n + 5$ we get that

$$\lambda(\{u_{\ell''-1}^{\ell''}, u_{\ell''}^{\ell''}\}) - \lambda(\{u_{\ell'-1}^{\ell'}, u_{\ell'}^{\ell'}\}) = 46n + 6.$$

It follows that

$$(2i+1) \cdot (20n+6) + 2\ell''' + 2 - ((2i-1) \cdot (20n+6) + 2\ell' + 2) = 46n + 6,$$

and hence $\ell''' - \ell' = 3n$. By construction of G we have that v_{i-2}^i and v_{i-1}^i are connected to $u_{\ell'-1}^{\ell'}$ and $u_{\ell'}^{\ell'}$ of the edge selection gadget of color combination $i-1, i$, respectively, via paths using vertices $\{a_{\ell'''}^{i,i-1,\ell} : \ell''' \in [5n]\}$ and $\{b_{\ell'''}^{i,i-1,\ell} : \ell''' \in [5n]\}$, respectively, if the vertex $w_{\ell'}^i \in W_i$ is incident with edge $e_{\ell'}^{i,i-1} \in F_{i-1,i}$. Furthermore, we have that v_i^i and v_{i+1}^i are connected to $u_{3n+\ell'-1}^{\ell''}$ and $u_{3n+\ell'}^{\ell''}$ of the edge selection gadget of color combination $i, i+1$, respectively, via

paths using vertices $\{a_{\ell''''}^{i,i+1,\ell''} : \ell'''' \in [5n]\}$ and $\{b_{\ell''''}^{i,i+1,\ell''} : \ell'''' \in [5n]\}$, respectively, if the vertex $w_{\ell'}^i \in W_i$ is incident with edge $e_{\ell''}^{i,i+1} \in F_{i,i+1}$.

Note that since $\sigma_{i-1,i}(\ell) = \sigma_{i,i+1}(\ell'') = 1$ we have that $e_{\ell}^{i-1,i} \in X$ and $e_{\ell''}^{i,i+1} \in X$. Since, again, ℓ' is independent from ℓ and j , it follows that $e_{\ell}^{i-1,i} \cap e_{\ell''}^{i,i+1} = \{w_{\ell'}^i\}$. By arguments analogous to the ones above we can also deduce that $\bigcap_{1 \leq j < i} e_{j,i} = \{w_{\ell'}^i\}$ and $\bigcap_{i < j \leq k} e_{i,j} = \{w_{\ell'}^i\}$. It follows that $(\bigcap_{1 \leq j < i} e_{j,i}) \cap (\bigcap_{i < j \leq k} e_{i,j}) = \{w_{\ell'}^i\}$.

We can conclude that indeed $\bigcup_{e \in X} e$ forms a multicolored clique in H .

(\Leftarrow): Assume H is a YES-instance of MULTICOLORED CLIQUE and let X be a solution. We construct the following labeling for the underlying graph G , see also Fig. 3 for an illustration.

We start with the labels for edges from the alignment gadget.

- For every $w \in \hat{W}$ we set $\lambda(\{w^*, w\}) = 1$.
- Let \hat{v}_0 belong to some connector gadget and let $w \in \hat{W}$ be neighbor of \hat{v}_0 . Then we set $\lambda(\{w, \hat{v}_0\}) = n^9$.
- Let y^i belong to the verification gadget of color i and let $w \in \hat{W}$ be neighbor of y^i . Then we set $\lambda(\{w, y^i\}) = n^8 - 1$. Furthermore, we set $\lambda(\{y_i, v_i^i\}) = n^8$.
- Let x_1 belong to the edge selection gadget for color combination i, j and let $w \in \hat{W}$ be neighbor of x_1 . Then we set $\lambda(\{w, x_1\}) = (i + j)(20n + 6)$.

Next, consider a connector gadget with vertices $\hat{v}_0, \hat{v}_0', \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_3'$ and set A, B .

- We set $\lambda(\{\hat{v}_0, \hat{v}_1\}) = \lambda(\{\hat{v}, \hat{v}_3\}) = n^9$.
- We set $\lambda(\{\hat{v}_0', \hat{v}_1\}) = \lambda(\{\hat{v}, \hat{v}_3'\}) = n^9 + 2$.
- We set $\lambda(\{\hat{v}_1, \hat{v}_2\}) = n^9 + 1$.
- For all vertices $v \in A \setminus B$ we set $\lambda(\{\hat{v}_1, v\}) = n^9$ and $\lambda(\{\hat{v}_2, v\}) = n^9 + 2$.
- For all vertices $v \in B$ we set $\lambda(\{\hat{v}_1, v\}) = \lambda(\{\hat{v}_2, v\}) = n^9$.
- For all vertices $v \in V^* \setminus A$ we set $\lambda(\{\hat{v}_1, v\}) = \lambda(\{\hat{v}_2, v\}) = n^9 + 2$. (Recall that V^* denotes the set of all vertices from all edge selection gadgets and all verification gadgets).

Recall that the following duration requirements were specified in the construction of the instance. It is straightforward to verify that duration requirements we recall in the following are all met, assuming no faster connections are introduced.

- We have set $d(\hat{v}_0, \hat{v}_2) = d(\hat{v}_3, \hat{v}_1) = d(\hat{v}_2, \hat{v}_0') = d(\hat{v}_1, \hat{v}_3') = 2$, and $d(\hat{v}_0, \hat{v}_0') = d(\hat{v}_3, \hat{v}_3') = d(\hat{v}_0, \hat{v}_3') = d(\hat{v}_3, \hat{v}_0') = 3$.
- Let $v \in A$, then we have set $d(v, \hat{v}_0') = 3$ and $d(v, \hat{v}_3') = 3$.
- Let $v \in V^* \setminus B$, then we have set $d(\hat{v}_0, v) = 3$ and $d(\hat{v}_3, v) = 3$.
- Let $v \in A$ and $v' \in V^* \setminus B$ such that v and v' are not neighbors, then we have set $d(v, v') = 3$.

For two connector gadgets, one with vertices $\hat{v}_0, \hat{v}_0', \hat{v}_1, \hat{v}_2, \hat{v}_3, \hat{v}_3'$ and with sets A and B , and one with vertices $\hat{v}_0', \hat{v}_0'', \hat{v}_1', \hat{v}_2', \hat{v}_3', \hat{v}_3''$ and with sets A' and B' , we have set the following durations.

- If there is a vertex $v \in A$ with $v \notin A'$, then we have set $d(\hat{v}_1, \hat{v}'_1) = 3$.
- If there is a vertex $v \in A$ with $v \in A' \setminus B'$, then we have set $d(\hat{v}_1, \hat{v}'_2) = 3$.
- If there is a vertex $v \in V^* \setminus (A \setminus B)$ with $v \notin A'$, then we have set $d(\hat{v}_2, \hat{v}'_1) = 3$.
- If there is a vertex $v \in V^* \setminus (A \setminus B)$ with $v \in A' \setminus B'$, then we have set $d(\hat{v}_2, \hat{v}'_2) = 3$.

For the alignment gadget, the following requirements were specified. Let x_1 belong to the edge selection gadget of color combination i, j and let $w \in \hat{W}$ denote the neighbor of x_1 in the alignment gadget. Let \hat{v}_1 and \hat{v}_2 belong to the first connector gadget of the edge selection gadget for color combination i, j . Let \hat{V} contain all vertices \hat{v}_1 and \hat{v}_2 belonging to the other connector gadgets (different from the first one of the edge selection gadget for color combination i, j).

- We have set $d(w^*, x_1) = (20n + 6)(i + j)$.
- We have set $d(w^*, \hat{v}_1) = n^9$, $d(w, \hat{v}_2) = n^9$, $d(w, \hat{v}_1) = n^9 - (20n + 6)(i + j) + 1$, and $d(w, \hat{v}_2) = n^9 - (20n + 6)(i + j) + 1$.
- For each vertex $v \in (V^* \cup \hat{V}) \setminus (X_{i,j} \cup \{v_{i,j}^{**}\})$ we have set $d(w^*, v) = n^9 + 2$ and $d(w, v) = n^9 - (20n + 6)(i + j) + 3$.

Let y^i belong to the verification gadget of color i and let $w' \in \hat{W}$ denote the neighbor of y^i in the alignment gadget. Let \hat{v}_1 and \hat{v}_2 belong to the connector gadget of the verification gadget for color i . Let \hat{V} contain all vertices \hat{v}_1 and \hat{v}_2 belonging to the other connector gadgets (different from the one of the verification gadget for color i). Let V_i denote the set of all vertices of the verification gadget of color i .

- We have set $d(w^*, y^i) = n^8 - 1$, $d(w', v_0^i) = 2$, and $d(w^*, v_0^i) = n^8$.
- We have set $d(w^*, \hat{v}_1) = n^9$, $d(w^*, \hat{v}_2) = n^9$, $d(w', \hat{v}_1) = n^9 - n^8$, and $d(w', \hat{v}_2) = n^9 - n^8$.
- For each vertex $v \in (V^* \cup \hat{V}) \setminus V_i$ we have set $d(w^*, v) = n^9 + 1$, $d(w, v) = n^9 - n^8 + 2$, and $d(y^i, v) = n^9 - n^8 + 2$.

Let \hat{v}_0 belong to some connector gadget. We have set $d(w^*, \hat{v}_0) = n^9$.

We will make sure that no faster connections are introduced by only using even numbers as labels and labels that are strictly smaller than $n^8 - 1$. Furthermore, we can already see that no vertex except the ones in \hat{W} can reach w^* and no two vertices $w, w' \in \hat{W}$ can reach each other, as required.

Next, consider the edge selection gadget for color combination i, j with $i < j$. To describe the labels, we define a permutation $\sigma_{i,j} : [m] \rightarrow [m]$ as follows. Let $\{w_{\ell'}^i\} = X \cap W_i$ and $\{w_{\ell''}^j\} = X \cap W_j$. Then, since X is a clique in H , we have that $\{w_{\ell'}^i, w_{\ell''}^j\} = e_{\ell}^{i,j} \in F_{i,j}$. We set $\sigma_{i,j}(\ell) = 1$ and $\sigma_{i,j}(1) = \ell$. For all $\ell''' \in [m]$ with $1 \neq \ell''' \neq \ell$ we set $\sigma_{i,j}(\ell''') = \ell'''$.

Let x_1, x_2, \dots, x_m belong to the edge selection gadget for color combination i, j .

- For all $\ell''' \in [m]$ we set $\lambda(\{x_{\ell'''}, v_{i,j}^*\}) = (i + j) \cdot (2n(\ell''')^2 + 18n + 6)$.

Note that by using these labels, we obey the following duration constraints.

- For all $1 \leq \ell''' < \ell'''' \leq m$ we have set $d(x_{\ell'''}, x_{\ell''''}) = 2n \cdot (i + j) \cdot ((\ell''')^2 - (\ell''')^2) + 1$.

Furthermore, we set the following labels.

- For all $\ell''' \in [m]$ we set $\lambda(\{u_0^{\ell'''}, v_{i,j}^{\star\star}\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell'''))^2 + 18n + 6) + 2$, where $u_0^{\ell'''}$ belongs to the edge selection gadget for i, j .
- For all $\ell''' \in [m]$ and $\ell'''' \in [4n]$ we set $\lambda(\{u_{\ell''''-1}^{\ell''''}, u_{\ell''''}^{\ell''''}\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell'''))^2 + 18n + 6) + 2\ell'''' + 2$, where $u_{\ell''''-1}^{\ell''''}$ and $u_{\ell''''}^{\ell''''}$ belong to the edge selection gadget for i, j .
- For all $\ell''' \in [m]$ we set $\lambda(\{u_{4n}^{\ell'''}, v_{i,j}^{\star\star}\}) = (i+j) \cdot (2n \cdot (\sigma_{i,j}(\ell'''))^2 + 18n + 6) + 8n + 4$, where $u_{4n}^{\ell'''}$ belongs to the edge selection gadget for i, j .

It is straightforward to verify that with these labels we get for all $\ell''' \in [m]$ that $d(x_{\ell'''}, v_{i,j}^{\star\star}) = 8n + 5$, as required. Furthermore, we get that for all $\ell''' \in [m]$ that $d(v_{i,j}^{\star\star}, x_{\ell''''}) = \infty$. To see this, consider the following. Vertex $v_{i,j}^{\star\star}$ is not temporally connected to vertices $x_{\ell''''}$ with $\ell''' \in [m]$ via any of the connector gadgets, since for all connector gadgets where $v_{i,j}^{\star\star} \in A$ we have that all vertices $x_{\ell''''}$ with $\ell''' \in [m]$ are either contained in B or they are not contained in A . By the construction of the labels of the connector gadgets, it follows that $v_{i,j}^{\star\star}$ cannot reach any vertex $x_{\ell''''}$ with $\ell''' \in [m]$ via the connector gadgets. We can observe that in all other connections in the underlying graph from $v_{i,j}^{\star\star}$ to a vertex $x_{\ell''''}$ with $\ell''' \in [m]$ are paths which have non-increasing labels, hence they also do not provide a temporal connection.

Furthermore, we get that for all $1 \leq \ell''' \leq \ell'''' \leq m$ we get that $d(x_{\ell'''}, x_{\ell''''}) = 2n \cdot (i+j) \cdot ((\ell''')^2 - (\ell''')^2) + 1$, through a temporal path via $v_{i,j}^{\star\star}$. By similar observations as in the previous paragraph, we also have that $d(x_{\ell''''}, x_{\ell''''}) = \infty$.

Finally, consider the verification gadget for color i . Let $1 \leq j < i$. Let $\{w_{\ell'}^i\} = X \cap W_i$ and $\{w_{\ell''}^j\} = X \cap W_j$ and $\{w_{\ell'}^i, w_{\ell''}^j\} = e_{\ell'}^{j,i} \in F_{j,i}$. Recall that we set $\sigma_{j,i}(\ell) = 1$ and $\sigma_{j,i}(1) = \ell$. For all $\ell'' \in [m]$ with $1 \neq \ell'' \neq \ell$ we set $\sigma_{j,i}(\ell'') = \ell''$. Recall that we set $\lambda(\{u_{\ell'-1}^{\ell}, u_{\ell'}^{\ell}\}) = (i+j) \cdot (20n+6) + 2\ell' + 2$, where $u_{\ell'-1}^{\ell}$ and $u_{\ell'}^{\ell}$ belong to the edge selection gadget for j, i . Now we set for all $\ell'' \in [5n-1]$ and all $\ell''' \in [m]$ the following.

- $\lambda(\{a_{5n}^{i,j,\ell'''}, u_{\ell'''}^{\ell''''}\}) = (i+j) \cdot (20n+6) + 2\ell'$ for all ℓ'''' such that this edge exists.
- $\lambda(\{a_1^{i,j,\ell'''}, v_{j-1}^i\}) = (i+j) \cdot (20n+6) + 2\ell' - 10n - 2$.
- $\lambda(\{a_{\ell'''}^{i,j,\ell'''}, a_{\ell'''+1}^{i,j,\ell''''}\}) = (i+j) \cdot (20n+6) + 2\ell' - 10n + 2\ell''$.
- $\lambda(\{b_{5n}^{i,j,\ell'''}, u_{\ell'''}^{\ell''''}\}) = (i+j) \cdot (20n+6) + 2\ell' + 4$ for all ℓ'''' such that this edge exists.
- $\lambda(\{b_1^{i,j,\ell'''}, v_j^i\}) = (i+j) \cdot (20n+6) + 2\ell' + 10n + 6$.
- $\lambda(\{b_{\ell'''}^{i,j,\ell'''}, b_{\ell'''+1}^{i,j,\ell''''}\}) = (i+j) \cdot (20n+6) + 2\ell' + 10n - 2\ell'' + 4$.

For all $\ell'' \in [13n]$ we set the following.

- $\lambda(\{\hat{u}_{\ell''}^i, \hat{u}_{\ell''+1}^i\}) = 2i \cdot (20n+6) + 2\ell' - 10n + 2\ell'' - 2$.
- $\lambda(\{v_{i-1}^i, \hat{u}_1^i\}) = 2i \cdot (20n+6) + 2\ell' - 10n - 2$.
- $\lambda(\{v_i^i, \hat{u}_{13n+1}^i\}) = 2i \cdot (20n+6) + 2\ell' + 16n + 4$.

Let $i < j \leq k$. Let $\{w_{\ell'}^i\} = X \cap W_i$ and $\{w_{\ell''}^j\} = X \cap W_j$ and $\{w_{\ell'}^i, w_{\ell''}^j\} = e_{\ell'}^{j,i} \in F_{i,j}$. Recall that we set $\sigma_{i,j}(\ell) = 1$ and $\sigma_{i,j}(1) = \ell$. For all $\ell'' \in [m]$ with $1 \neq \ell'' \neq \ell$ we set $\sigma_{i,j}(\ell'') = \ell''$. Recall that we set $\lambda(\{u_{3n+\ell'-1}^{\ell}, u_{3n+\ell'}^{\ell}\}) = (i+j) \cdot (20n+6) + 2\ell' + 6n + 2$, where $u_{3n+\ell'-1}^{\ell}$ and $u_{3n+\ell'}^{\ell}$ belong to the edge selection gadget for i, j . Now we set for all $\ell'' \in [5n-1]$ and all $\ell''' \in [m]$ the following.

- $\lambda(\{a_{5n}^{i,j,\ell'''}, u_{\ell'''}^{\ell'''}\}) = (i+j) \cdot (20n+6) + 2\ell' + 6n$ for all ℓ''' such that this edge exists.
- $\lambda(\{a_1^{i,j,\ell'''}, v_{j-1}^i\}) = (i+j) \cdot (20n+6) + 2\ell' - 4n - 2$.
- $\lambda(\{a_{\ell'''}^{i,j,\ell'''}, a_{\ell'''+1}^{i,j,\ell'''}\}) = (i+j) \cdot (20n+6) + 2\ell' - 4n + 2\ell''$.
- $\lambda(\{b_{5n}^{i,j,\ell'''}, u_{\ell'''}^{\ell'''}\}) = (i+j) \cdot (20n+6) + 2\ell' + 6n + 4$ for all ℓ''' such that this edge exists.
- $\lambda(\{b_1^{i,j,\ell'''}, v_j^i\}) = (i+j) \cdot (20n+6) + 2\ell' + 16n + 6$.
- $\lambda(\{b_{\ell'''}^{i,j,\ell'''}, b_{\ell'''+1}^{i,j,\ell'''}\}) = (i+j) \cdot (20n+6) + 2\ell' + 16n - 2\ell'' + 4$.

Now we verify that we meet the duration requirements. For all $0 \leq j < j' < i$ and all $i \leq j < j' \leq k$ we have set the following.

- We set $d(v_j^i, v_{j'}^i) = (20n+6)(j' - j) - 1$.

To see that this holds, we analyze the fastest paths from vertices v_{j-1}^i to vertices v_j^i for $j \in [k] \setminus \{i\}$. Let $\{w_{\ell'}^i\} = X \cap W_i$ and $\{w_{\ell'}^j\} = X \cap W_j$ and $\{w_{\ell'}^i, w_{\ell'}^j\} = e_{\ell'}^{i,j} \in F_{i,j}$. Then, starting at v_{j-1}^i , we follow the vertices in $\{a_{\ell''}^{i,j,\ell} : \ell'' \in [5n]\}$ to arrive at $u_{\ell'-1}^{\ell}$. From there we move to $u_{\ell'}^{\ell}$ and from there we continue along the vertices in $\{b_{\ell''}^{i,j,\ell} : \ell'' \in [5n]\}$ to arrive at v_j^i . By construction, this describes a fastest temporal path from v_{j-1}^i to v_j^i with duration $20n+5$. To get from v_j^i to $v_{j'}^i$ for $0 \leq j < j' < i$ we move from v_j^i to v_{j+1}^i in the above described fashion and from there to v_{j+1}^i and so on until we arrive at $v_{j'}^i$. By construction this yields a fastest temporal path from v_j^i to $v_{j'}^i$ with duration $(20n+6)(j' - j) - 1$, as required. The case where $i \leq j < j' \leq k$ is analogous.

For all $0 \leq j < i$ and all $i \leq j' \leq k$ we have set the following.

- We set $d(v_j^i, v_{j'}^i) = (20n+6)(j' - j) + 6n - 1$.

Here we move from v_j^i to v_{i-1}^i in the above-described fashion. Then we move from v_{i-1}^i to v_i^i along vertices $\{\hat{u}_{\ell''}^i : \ell'' \in [13+1]\}$ and then we move from v_i^i to $v_{j'}^i$, again in the above described fashion. By construction this yields a fastest temporal path from v_j^i to $v_{j'}^i$ with duration $(20n+6)(j' - j) + 6n - 1$, as required.

By similar observations as in the analysis for the edge selection gadgets, we also get that for all $1 \leq j < j' \leq k$ that $d(v_{j'}^i, v_j^i) = \infty$.

This finishes the proof.

Infinity gadget. Finally, we show how to get rid of the infinity entries in D and how to allow a finite Δ . To this end, we introduce the *infinity gadget*. We add four vertices z_1, z_2, z_3, z_4 to the graph and we set $\Delta = n^{11}$. Let V denote the set of all remaining vertices. We set the following durations.

- For all $v \in V$ we set $d(z_1, v) = 2$, $d(z_2, v) = d(v, z_2) = 1$, $d(z_3, v) = d(v, z_3) = 1$, and $d(z_4, v) = 2$. Furthermore, we set $d(v, z_1) = n^{11}$ and $d(v, z_4) = n^{10} - 1$.
- We set $d(z_1, z_2) = d(z_2, z_1) = 1$, $d(z_2, z_3) = d(z_3, z_2) = 1$, and $d(z_3, z_4) = d(z_4, z_3) = 1$.
- We set $d(z_1, z_3) = 3$, $d(z_3, z_1) = n^{11} - 1$, $d(z_2, z_4) = n^{10} - 2$, and $d(z_4, z_2) = n^{11} - n^{10} + 4$.
- We set $d(z_1, z_4) = n^{10}$ and $d(z_4, z_1) = 2n^{11} - n^{10} + 2$.

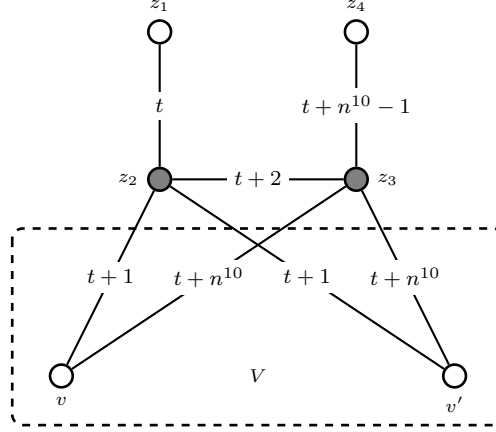


Figure 4: Illustration of the infinity gadget. Gray vertices need to be added to the feedback vertex set.

- For every pair of vertices $v, v' \in V$ where previously the duration of a fastest path from v to v' was specified to be infinite, we set $d(v, v') = n^{10}$.

Now we analyze which implications we get for the labels on the newly introduced edges. Assume $\lambda(\{z_1, z_2\}) = t$, then we get the following. For all $v \in V$ we have that $d(z_1, v) = 2$ and hence we get that $\lambda(\{z_2, v\}) = t + 1$. Since $d(z_1, z_4) = n^{10}$, we have that $\lambda(\{z_3, z_4\}) = t + n^{10} - 1$. From this follows that for all $v \in V$, since $d(z_4, v) = 2$, that $\lambda(\{z_3, v\}) = t + n^{10}$. Finally, since $d(z_1, z_3) = 3$, we have that $\lambda(\{z_2, z_3\}) = t + 2$. For an illustration see Fig. 4. It is easy to check that all duration requirements between vertex pairs in $\{z_1, z_2, z_3, z_4\}$ are met and that all duration requirements between each vertex $v \in V$ and each vertex in $\{z_1, z_2, z_3, z_4\}$ are met. Furthermore, it is easy to check that the gadget increases the feedback vertex set by two (z_2 and z_3 need to be added).

Lastly, consider two vertices $v, v' \in V$. Note that before the addition of the infinity gadget, by construction of G we have that $d(v, v') \leq n^9 + 2$ or $d(v, v') = \infty$. Furthermore, if D is a YES-instance, we have shown in the correctness proof of the reduction that the difference between the smallest label and the largest label is at most $n^9 + 1$. This implies that for a vertex pair $v, v' \in V$ with $d(v, v') = \infty$ we have in the periodic case with $\Delta = n^{11}$, that $d(v, v') \geq n^{11} - n^9 > n^{10}$. This means, after adding the vertices and edges of the infinity gadget, we indeed have that $d(v, v') = n^{10}$. For all vertex pairs v, v' where in the original construction we have $d(v, v') \neq \infty$, we can also see that adding the infinity gadget and setting $\Delta = n^{11}$ does not change the duration of a fastest path from v to v' , since all newly added temporal paths have duration at least n^{10} . We can conclude that the originally constructed instance D is a YES-instance if and only if it remains a YES-instance after adding the infinity gadget and setting $\Delta = n^{11}$. \square

3. Algorithms for Simple TGR

In this section, we provide several algorithms for SIMPLE TGR. By Theorem 1 we have that SIMPLE TGR is NP-hard in general, hence we start by identifying restricted cases where we can solve the problem in polynomial time. We first show in Section 3.1 that if the underlying graph G of an instance (D, Δ) of SIMPLE TGR is a tree, then we can determine desired Δ -periodic labeling λ of G in polynomial time. In Section 3.2 we generalize this result. We show that SIMPLE TGR is fixed-parameter tractable when parameterized by the feedback edge number of the underlying graph. Note that our parameterized hardness result (Theorem 2) implies that we presumably cannot replace the feedback edge number with the smaller parameter feedback

vertex number, or any other parameter that is smaller than the feedback vertex number, such as e.g. the treewidth.

3.1. Polynomial-time algorithm for trees

We now provide a polynomial-time algorithm for SIMPLE TGR when the underlying graph is a tree. Let D be the input matrix and let the underlying graph G of D be a tree on n vertices $\{v_1, v_2, \dots, v_n\}$. Let v_i, v_j be two arbitrary vertices in G , then we know that there exists a unique (static) path $P_{i,j}$ from v_i to v_j . We will heavily exploit this in our algorithm.

Theorem 3. SIMPLE TGR can be solved in polynomial time on trees.

Proof. Let D be an input matrix for problem SIMPLE TGR of dimension $n \times n$. Let us fix the vertices of the corresponding graph G of D as v_1, v_2, \dots, v_n , where vertex v_i corresponds to the row and column i of matrix D . This can be done in polynomial time as we need to loop through the matrix D once and connect vertices v_i, v_j for which $D_{i,j} = 1$. At the same time, we also check if $D_{i,i} = 0$, for all $i \in [n]$. When G is constructed we run a DFS algorithm on it and check that it has no cycles. If at any step we encounter a problem, our algorithm stops and returns a negative answer.

Having computed G , our algorithm proceeds as follows. We pick an arbitrary edge f and give it label one, that is, $\lambda(f) = 1$. Now we push all edges incident with f into a (initially empty) queue. Now we repeat the following as long as the queue is not empty:

- Pop edge $e = \{u, v\}$ from the queue. Since e was pushed into the queue, there is an edge e' incident with e that already obtained a label. Let w.l.o.g. $e' = \{v, w\}$. Then we set $\lambda(e) = (\lambda(e') - D_{u,w} + 1) \bmod \Delta$.
- Push all edges incident with e that have not received a label yet into the queue.

When the queue is empty, all edges have received a label. Iterate over all vertex pairs u, v and check whether a fastest path from u to v in (G, λ) has duration $D_{u,v}$. If this check succeeds for all vertex pairs, output the labeling λ , otherwise abort.

It is easy to see that the described algorithm runs in polynomial time. In the remainder, we prove that it is correct.

(\Rightarrow): Since the algorithm checks at the end whether all durations specified in D are realized by the corresponding fastest paths, we clearly face a yes-instance whenever the algorithm outputs a labeling.

(\Leftarrow): Assume we face a yes-instance, then there exists a labeling λ^* that realizes all durations specified in D . Let e^* denote the edge initially picked by the algorithm. For all edges e let $\lambda(e) = (\lambda^*(e) - \lambda^*(e^*) + 1) \bmod \Delta$. Clearly, the labeling λ also realizes all durations specified in D since λ is obtained by adding the constant $(1 - \lambda^*(e^*))$ modulo Δ to all labels of λ^* which does not change the duration of any temporal path, that is all durations in (G, λ^*) are the same as their counterparts in (G, λ) . We claim that our algorithm computes and outputs λ .

We prove that our algorithm computes λ by induction on the distance of the labeled edges to e^* , where the distance of two edges e, e' is defined as the length of a shortest path that uses e as its first edge and e' as its last edge.

Initially, our algorithm labels e^* with one, which equals $\lambda(e^*)$. Now let e be an edge popped off the queue by the algorithm in some iteration, that is on the distance i from e^* . Let e' be the edge incident with e that is on the distance $i - 1$ from e^* . Since G is a tree e' has already been considered by the algorithm and thus already has a label. By induction, we have that the algorithm labeled e' with $\lambda(e')$. Assume that $e = \{u, v\}$ and $e' = \{v, w\}$. Since G is a tree there is only one path from u to w in G and it uses edges e and e' . It follows that

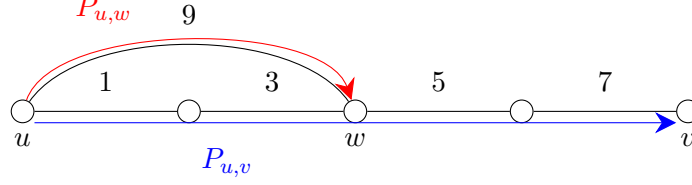


Figure 5: An example of a temporal graph (with $\Delta \geq 9$), where a fastest temporal path $P_{u,v}$ (in blue) from u to v is of duration 7, while a fastest temporal path $P_{u,w}$ (in red) from u to a vertex w , that is on a path $P_{u,v}$, is of duration 1 and is not a subpath of $P_{u,v}$.

$\lambda(e') - \lambda(e) + 1 = D_{u,w}$ if $\lambda(e') > \lambda(e)$, and $\lambda(e') - \lambda(e) + \Delta + 1 = D_{u,w}$ otherwise. Our algorithm labels e with $(\lambda(e') - D_{u,w} + 1) \bmod \Delta$. It is straightforward to verify that the label of e computed by the algorithm equals $\lambda(e)$. It follows that the algorithm computes λ . \square

3.2. FPT-algorithm for feedback edge number

Recall from Section 3.1 that the main reason, for which SIMPLE TGR is straightforward to solve on trees, is twofold:

- between any pair of vertices v_i and v_j in the tree T , there is a *unique* path P in T from v_i to v_j , and
- in any periodic temporal graph (T, λ, Δ) and any fastest temporal path $P = ((e_1, t_1), \dots, (e_i, t_i), \dots, (e_j, t_j), \dots, (e_{\ell-1}, t_{\ell-1}))$ from v_1 to v_ℓ we have that the sub-path $P' = ((e_i, t_i), \dots, (e_{j-1}, t_{j-1}))$ is also a fastest temporal path from v_i to v_j .

However, these two nice properties do not hold when the underlying graph is not a tree. For example, in Fig. 5, a fastest temporal path from u to v is $P_{u,v}$ (depicted in blue) goes through w , however, the sub-path of $P_{u,v}$ that stops at w is not a fastest temporal path from u to w . A fastest temporal path from u to w consists only of the single edge uw (with label 9 and duration 1, depicted in red).

Nevertheless, we prove in this section that we can still solve SIMPLE TGR efficiently if the underlying graph is similar to a tree; more specifically we show the following result, which turns out to be non-trivial.

Theorem 4. SIMPLE TGR is in FPT when parameterized by the feedback edge number of the underlying graph.

From Theorem 2 and Theorem 4 we immediately get the following, which is the main result of the paper.

Corollary 1. SIMPLE TGR is:

- in FPT when parameterized by the feedback edge number or any larger parameter, such as the maximum leaf number.
- $W[1]$ -hard when parameterized by the feedback vertex number or any smaller parameter, such as: treewidth, degeneracy, cliquewidth, distance to chordal graphs, and distance to outerplanar graphs.

Before presenting the structure of our algorithm for Theorem 4, observe that, in a static graph, the number of paths between two vertices can be upper-bounded by a function $f(k)$ of the feedback edge number k of the graph [14]. This is true as any such path can traverse $0, 1, 2, \dots, k$ feedback edges in different order. Therefore, for any fixed pair of vertices u and v ,

we can “guess” the edges of a fastest temporal path from u to v (by guess we mean enumerate and test all possibilities). However, for an FPT algorithm with respect to k , we cannot afford to guess the edges of a fastest temporal path for each of the $O(n^2)$ pairs of vertices. To overcome this difficulty, our algorithm follows this high-level strategy:

- We identify a small number $f(k)$ of “important vertices”.
- For each pair u, v of important vertices, we guess the edges of a fastest temporal path from u to v (and from v to u).
- From these guesses we can still not deduce the edges of the fastest temporal paths between many pairs of non-important vertices. However, as we prove, it suffices to guess only a small number of specific auxiliary structures (to be defined later).
- From these guesses we deduce fixed relationships between the labels of most of the edges of the graph.
- For all the edges, for which we have not deduced a label yet, we introduce a *variable*. With all these variables, we build an Integer Linear Program (ILP). Among the constraints in this ILP we have that, for each of the $O(n^2)$ pairs of vertices u, v in the graph, the duration of one specific temporal path from u to v (according to our guesses) is *equal* to the desired duration $D_{u,v}$, while the duration of each of the other temporal path from u to v is *at least* $D_{u,v}$.
- Each specific configuration of fastest temporal paths among all pairs of vertices corresponds to a specific ILP instance. By exhaustively trying all possible fastest temporal paths configurations it follows that our instance of SIMPLE TGR has a solution if and only if at least one of these ILPs has a feasible solution. As each ILP can be solved in FPT time with respect to k by Lenstra’s Theorem [51] (the number of variables is upper bounded by a function of k), we obtain our FPT algorithm for SIMPLE TGR with respect to k .

For the remainder of this section, we fix the following notation. Let D be the input matrix of SIMPLE TGR, i.e., the matrix of the durations of the fastest temporal paths between all pairs of n vertices, and let G be its underlying graph, on n vertices and m edges. With F we denote a minimum feedback edge set of G , and with k the feedback edge number of G . We are now ready to present our FPT algorithm. For easier readability, we split the description and analysis of the algorithm into five subsections. We start with a preprocessing procedure for graph G , where we define a set of interesting vertices which then allows us to guess the desired structures. Next, we introduce some extra properties of our problem, that we then use to create ILP instances and their constraints. At the end, we present how to solve all instances and produce the desired labeling λ of G , if possible.

3.2.1. Preprocessing of the input

From the underlying graph G of D we extract a (connected) graph G' by iteratively removing vertices of degree one from G , and denote with

$$Z = V(G) \setminus V(G').$$

Then we determine a minimum feedback edge set F of G' . Note that F is also a minimum feedback edge set of G . Lastly, we determine sets U , of *vertices of interest*, and U^* of the neighbors of vertices of interest, in the following way. Let T be a spanning tree of G' , with F

being the corresponding feedback edge set of G' . Let $V_1 \subseteq V(G')$ be the set of leaves in the spanning tree T , $V_2 \subseteq V(G')$ be the set of vertices of degree two in T , that are incident to at least one edge in F , and let $V_3 \subseteq V(G')$ be the set of vertices of degree at least 3 in T . Then $|V_1| + |V_2| \leq 2k$, since every leaf in T and every vertex in V_2 is incident to at least one edge in F , and $|V_3| \leq |V_1|$ by the properties of trees. We denote with

$$U = V_1 \cup V_2 \cup V_3$$

the set of vertices of interest. It follows that $|U| \leq 4k$. We set U^* to be the set of vertices in $V(G') \setminus U$ that are neighbors of vertices in U , i.e.,

$$U^* = \{v \in V(G') \setminus U : u \in U, v \in N(u)\}.$$

Again, using the tree structure, we get that for any $u \in U$ its neighborhood is of size $|N(u)| \in O(k)$, since every neighbor of u is the first vertex of a (unique) path to another vertex in U . It follows that $|U^*| \in O(k^2)$.

From the construction of Z (i.e., by exhaustively removing vertices of degree one from G), it follows that $G[Z]$ (the graph induced in G by Z) is a forest, i.e., consists of disjoint trees. Each of these trees has a unique neighbor v in G' . Denote by T_v the tree obtained by considering such a vertex v and all the trees from $G[Z]$ that are incident to v in G . We then refer to v as the *clip vertex* of the tree T_v . In the case where v is a vertex of interest we define also the set Z_v^* of *representative vertices* of T_v , as follows. We first create an empty set C_w for every vertex w that is a neighbor of v in G' . We then iterate through every vertex r that is in the first layer of the tree T_v (i.e., vertex that is a child of the root v in the tree T_v), check the matrix D and find a vertex $w \in N_{G'}(v)$ that is on the smallest duration from r . In other words, for an $r \in N_{T_v}(v)$ we find $w \in N_{G'}(v)$ such that $D_{r,w} \leq D_{r,w'}$ for all $w' \in N_{G'}(v)$. We add vertex r to C_w . In the case where there exists also another vertex $w' \in N_{G'}(v)$ for which $D_{r,w'} = D_{r,w}$, we add r also to the set $C_{w'}$. In fact, in this case $C_{w'} = C_w$. At the end we create $|N_{G'}(v)| \in O(k)$ sets C_w , whose union contains all children of v in T_v . For every two sets C_w and $C_{w'}$, where $w, w' \in N_{G'}(v)$, we have that either $C_w = C_{w'}$, or $C_w \cap C_{w'} = \emptyset$. We interpret each of these sets $\{C_w : w \in N_{G'}(v)\}$ as an *equivalence class* of the neighbors of v in the tree T_v . Now, from each equivalence class C_w we choose an arbitrary vertex $r_w \in C_w$ and put it into the set Z_v^* . We repeat the above procedure for all trees T_v with the clip vertex v from U , and define Z^* as

$$Z^* = \bigcup_{v \in U} Z_v^*. \quad (1)$$

By the above construction, we have that Z_v^* contains one different vertex r_w of the tree T_v for (at most) every neighbor w of v in G' (where v is the clip vertex of the tree T_v). Therefore, since $|N_{G'}(v)| = O(k)$, it follows that $|Z_v^*| = O(k)$. Thus, since $|U| = O(k)$, we have by equality (1) that $|Z^*| = O(k^2)$.

Finally, the set of *important vertices* is defined as the set $U \cup U^* \cup Z^*$. For an illustration see Fig. 6. Note that determining sets U, U^* and Z^* takes linear time.

Recall that a labeling λ of G satisfies D if the duration of a fastest temporal path from each vertex v_i to each other vertex v_j equals D_{v_i, v_j} . In order to find a labeling that satisfies this property we split our analysis in nine cases. We consider the fastest temporal paths where the starting vertex is in one of the sets $U, V(G') \setminus U, Z$, and similarly the destination vertex is in one of the sets $U, V(G') \setminus U, Z$. In each of these cases, we guess the underlying path P that at least one fastest temporal path from the vertex v_i to v_j follows, which results in one equality constraint for the labels on the path P . For all other temporal paths from v_i to v_j we know that they cannot be faster, so we introduce inequality constraints for them. This results

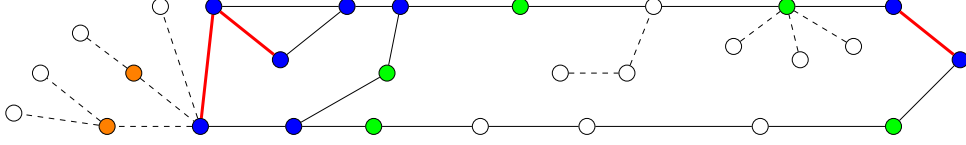


Figure 6: An example of a graph with its important vertices: U (in blue), U^* (in green) and Z^* (in orange). Corresponding feedback edges are marked with a thick red line, while dashed edges represent the edges (and vertices) “removed” from G' at the initial step.

in producing $f(k) \cdot |D|^{O(1)}$ constraints, where $|D| = n^2$. Note that we have to do this while keeping the total number of variables upper-bounded by some function in k .

For an easier understanding and analysis of the algorithm, we give the following definition.

Definition 3. Let $U \subseteq V(G')$ be a set of vertices of interest and let $u, v \in U$. A path $P = (u = v_1, v_2, \dots, v_p = v)$ with at least two edges in graph G' , where all inner vertices are not in U , i. e., $v_i \notin U$ for all $i \in \{2, 3, \dots, p-1\}$, is called a segment from u to v , which we denote as $S_{u,v}$.

Note from Definition 3 that $S_{u,v} \neq S_{v,u}$ since we consider paths to be directed. It is also worth emphasizing that $S_{v,u}$ is essentially the reverse path of $S_{u,v}$. Furthermore, it's important to observe that a temporal path in G' between two vertices of interest is either a segment or consists of a sequence of segments. Moreover, any inner vertex v_i in the segment $S_{u,v}$ ($v_i \in S_{u,v} \setminus \{u, v\}$) is part of precisely two segments: $S_{u,v}$ and $S_{v,u}$. Given that we have at most $4k$ interesting vertices in G' , we can deduce the following crucial result.

Corollary 2. There are $O(k^2)$ segments in G' .

3.2.2. Guessing necessary structures

Once the sets U, U^* and Z^* are determined, we are ready to start guessing the necessary structures. Note that whenever we say that we guess a fastest temporal path between two vertices, we mean that we guess an underlying path of a representative fastest temporal path between those two vertices. To describe the guesses, we introduce the following notation. Let u, v, x be three vertices in G' . We write $u \rightsquigarrow x \rightarrow v$ to denote a temporal path from u to v that passes through x , and then goes directly to v (via one edge or a unique path in G'). In other words, if a fastest path between two vertices is not uniquely determined we denote it by \rightsquigarrow , while if it is unique we denote it by \rightarrow . We guess the following paths.

- G-1. The fastest temporal paths between all pairs of vertices of U . For a pair u, v of vertices in U , there are $k^{O(k)}$ possible paths in G' between them. Therefore, we try all $k^{O(k)}$ possible paths, where at least one of them will be a fastest temporal path from u to v , respecting the values from D . Repeating this procedure for all pairs of vertices $u, v \in U$ we get $k^{O(k^3)}$ different variations of the fastest temporal paths between all pairs of vertices in U .
- G-2. The fastest temporal paths between all pairs of vertices in Z^* , which by similar arguing as for vertices in U , gives us $k^{O(k^5)}$ guesses, as $|Z^*| = O(k^2)$.
- G-3. The fastest temporal paths between all pairs of vertices in U^* . This gives us $k^{O(k^5)}$ guesses.
- G-4. The fastest temporal paths from vertices of U to vertices in U^* , and vice versa, the fastest temporal paths from vertices in U^* to vertices in U . This gives us $k^{O(k^4)}$ guesses.
- G-5. The fastest temporal paths from vertices of U to vertices in Z^* , and vice versa. This gives us $k^{O(k^4)}$ guesses.
- G-6. The fastest temporal paths from vertices of U^* to vertices in Z^* , and vice versa. This gives us $k^{O(k^5)}$ guesses.

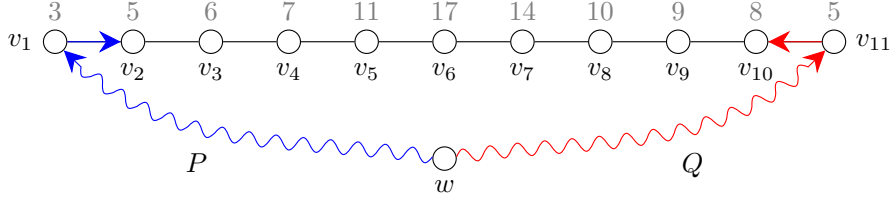


Figure 7: In the above graph vertices v_1, v_{11}, w are in U , while v_2, v_{10} are in U^* . Numbers above all v_i represent the values of the fastest temporal paths from w to each of them (i.e., the entries in the w -th row of matrix D). From the basic guesses we know a fastest temporal path P from w to v_2 (depicted in blue) and a fastest temporal path Q from w to v_{10} . From the values of durations from w to each v_i we cannot determine the fastest paths from w to all v_i . More precisely, we know that w reaches v_2, v_3, v_4, v_5 (resp. v_{10}, v_9, v_8, v_7) by first using the path P (resp. Q) and then proceeding through the vertices, but we do not know how w reaches v_6 the fastest. Therefore we have to introduce some more guesses.

With the information provided by the described guesses we are still not able to determine all fastest paths. For example consider the case depicted in Fig. 7. Therefore, we introduce additional guesses that provide us with sufficient information to determine all fastest paths. We guess the following structures.

G-7. Inner segment guess I. Let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ and $S_{w,z} = (w = z_1, z_2, \dots, z_r = z)$ be two segments in G' . We want to guess a fastest temporal path $v_2 \rightarrow u \rightsquigarrow w \rightarrow z_2$. We repeat this procedure for all pairs of segments. Since there are $O(k^2)$ segments in G' , there are $k^{O(k^5)}$ possible paths of this form.

Recall that $S_{u,v} \neq S_{v,u}$ for every $u, v \in U$. Furthermore note that we did not assume that $\{u, v\} \cap \{w, z\} = \emptyset$. Therefore, by repeatedly making the above guesses, we also guess the following fastest temporal paths: $v_2 \rightarrow u \rightsquigarrow z \rightarrow z_{r-1}$, $v_2 \rightarrow u \rightsquigarrow v \rightarrow v_{p-1}$, $v_{p-1} \rightarrow v \rightsquigarrow w \rightarrow z_2$, $v_{p-1} \rightarrow v \rightsquigarrow z \rightarrow z_{r-1}$, and $v_{p-1} \rightarrow v \rightsquigarrow u \rightarrow v_2$. For an example see Fig. 8a.

G-8. Inner segment guess II. Let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ be a segment in G' , and let $w \in U \cup Z^*$. We want to guess the following fastest temporal paths $w \rightsquigarrow u \rightarrow v_2$, $w \rightsquigarrow v \rightarrow v_{p-1} \rightarrow \dots \rightarrow v_2$, and $v_2 \rightarrow u \rightsquigarrow w$, $v_2 \rightarrow v_3 \rightarrow \dots \rightarrow v \rightsquigarrow w$.

For fixed $S_{u,v}$ and $w \in U \cup Z^*$ we have $k^{O(k)}$ different possible such paths, therefore we make $k^{O(k^5)}$ guesses for these paths. For an example see Fig. 8b.

G-9. Split vertex guess I. Let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ be a segment in G' , and let us fix a vertex $v_i \in S_{u,v} \setminus \{u, v\}$. In the case when $S_{u,v}$ is of length 4, the fixed vertex v_i is the middle vertex, else we fix an arbitrary vertex $v_i \in S_{u,v} \setminus \{u, v\}$. Let $S_{w,z} = (w = z_1, z_2, \dots, z_r = z)$ be another segment in G' . We want to determine the fastest paths from v_i to all inner vertices of $S_{w,z}$. We do this by inspecting the values in matrix D from v_i to inner vertices of $S_{w,z}$. We split the analysis into two cases.

- (a) There is a single vertex $z_j \in S_{w,z}$ for which the duration from v_i is the biggest. More specifically, $z_j \in S_{w,z} \setminus \{w, z\}$ is the vertex with the biggest value D_{v_i, z_j} . We call this vertex a *split vertex of v_i in the segment $S_{w,z}$* . Then it holds that $D_{v_i, z_2} < D_{v_i, z_3} < \dots < D_{v_i, z_j}$ and $D_{v_i, z_{r-1}} < D_{v_i, z_{r-2}} < \dots < D_{v_i, z_j}$. From this it follows that the fastest temporal paths from v_i to z_2, z_3, \dots, z_{j-1} go through w , and the fastest temporal paths from v_i to $z_{r-1}, z_{r-2}, \dots, z_{j+1}$ go through z . We now want to guess which vertex w or z is on a fastest temporal path from v_i to z_j . Similarly, all fastest temporal paths starting at v_i have to go either through u or through v , which also gives us two extra guesses for a fastest temporal path from v_i to z_j . Therefore, all together we have 4 possibilities on how a fastest temporal path from v_i to z_j starts and ends. Besides that, we want to guess also how the fastest temporal paths from

v_i to z_{j-1}, z_{j+1} start and end. Note that one of these is the subpath of a fastest temporal path from v_i to z_j , and the ending part is uniquely determined for both of them, i.e., to reach z_{j-1} a fastest temporal path travels through w , and to reach z_{j+1} a fastest temporal path travels through z . Therefore we have to determine only how the path starts, namely if it travels through u or v . This introduces two extra guesses. For a fixed $S_{u,v}, v_i$ and $S_{w,z}$ we find the vertex z_j in polynomial time, or determine that z_j does not exist. We then make four guesses where we determine how a fastest temporal path from v_i to z_j passes through vertices u, v and w, z and for each of them two extra guesses to determine a fastest temporal path from v_i to z_{j-1} and from v_i to z_{j+1} . We repeat this procedure for all pairs of segments, which results in producing $k^{O(k^5)}$ new guesses. Note, $v_i \in S_{u,v}$ is fixed when calculating the split vertex for all other segments $S_{w,z}$.

- (b) There are two vertices $z_j, z_{j+1} \in S_{w,z}$ for which the duration from v_i is the biggest. More specifically, $z_j, z_{j+1} \in S_{w,z} \setminus \{w, z\}$ are the vertices with the biggest value $D_{v_i, z_j} = D_{v_i, z_{j+1}}$. Then it holds that $D_{v_i, z_2} < D_{v_i, z_3} < \dots < D_{v_i, z_j} = D_{v_i, z_{j+1}} > D_{v_i, z_{j+2}} > \dots > D_{v_i, z_{r-1}}$. From this it follows that the fastest temporal paths from v_i to z_2, z_3, \dots, z_j go through w , and the fastest temporal paths from v_i to $z_{r-1}, z_{r-2}, \dots, z_{j+1}$ go through z . In this case, we only need to guess the following two fastest temporal paths $v_i \rightsquigarrow w \rightarrow z_2$ and $v_i \rightsquigarrow z \rightarrow z_{r-1}$. Each of these paths we then uniquely extend along the segment $S_{w,z}$ up to the vertex z_j , resp. z_{j+1} , which give us fastest temporal paths from v_i to z_j and from v_i to z_{j+1} . In this case, we introduce only two more guesses. We repeat this procedure for all pairs of segments, which results in creating $k^{O(k^5)}$ new guesses.

For an example see Fig. 8c.

- G-10. **Split vertex guess II.** Let $w \in U \cup Z^*$, and let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ be a segment in G' . We want to guess a split vertex of w in $S_{u,v}$, and a fastest temporal path that reaches it. We again have two cases, first one where v_i is a unique vertex in $S_{u,v}$ that is furthest away from w , and the second one where v_i, v_{i+1} are two incident vertices in $S_{u,v}$, that are furthest away from w . All together we make two guesses for each pair $w, S_{u,v}$. We repeat this for all vertices in $U \cup Z^*$, and all segments, which produces $k^{O(k^5)}$ new guesses. For an example see Fig. 8d.

There are two more guesses G-11 and G-12 that we make during the creation of the ILP instances, we explain these guesses in detail in Section 3.2.4. We will prove that, for all guesses G-1 to G-12, there are in total at most $f(k)$ possible choices, and for each one of them we create an ILP with at most $f(k)$ variables and at most $f(k) \cdot |D|^{O(1)}$ constraints. Each of these ILPs can be solved in FPT time by Lenstra's Theorem [51].

3.2.3. Properties of SIMPLE TGR

In this section, we study the properties of our problem, which then help us creating constraints for our ILP instances. Recall that with G we denote our underlying graph of D . We want to determine labeling λ of each edge of G . We start with an empty labeling of edges and try to specify each one of them. Note, that this does not necessarily mean that we assign numbers to the labels, but we might specify labels as variables or functions of other labels. We say that the label of an edge e' is *determined with respect to* the label of the edge e , if we have determined $\lambda(e')$ as a function of $\lambda(e)$.

We first start with defining certain notions, that will be of use when solving the problem.

Definition 4 (Travel delays). *Let (G, λ) be a temporal graph satisfying the conditions of SIMPLE TGR. Let $e_1 = uv$ and $e_2 = vz$ be two incident edges in G with $e_1 \cap e_2 = \{v\}$. We define the*

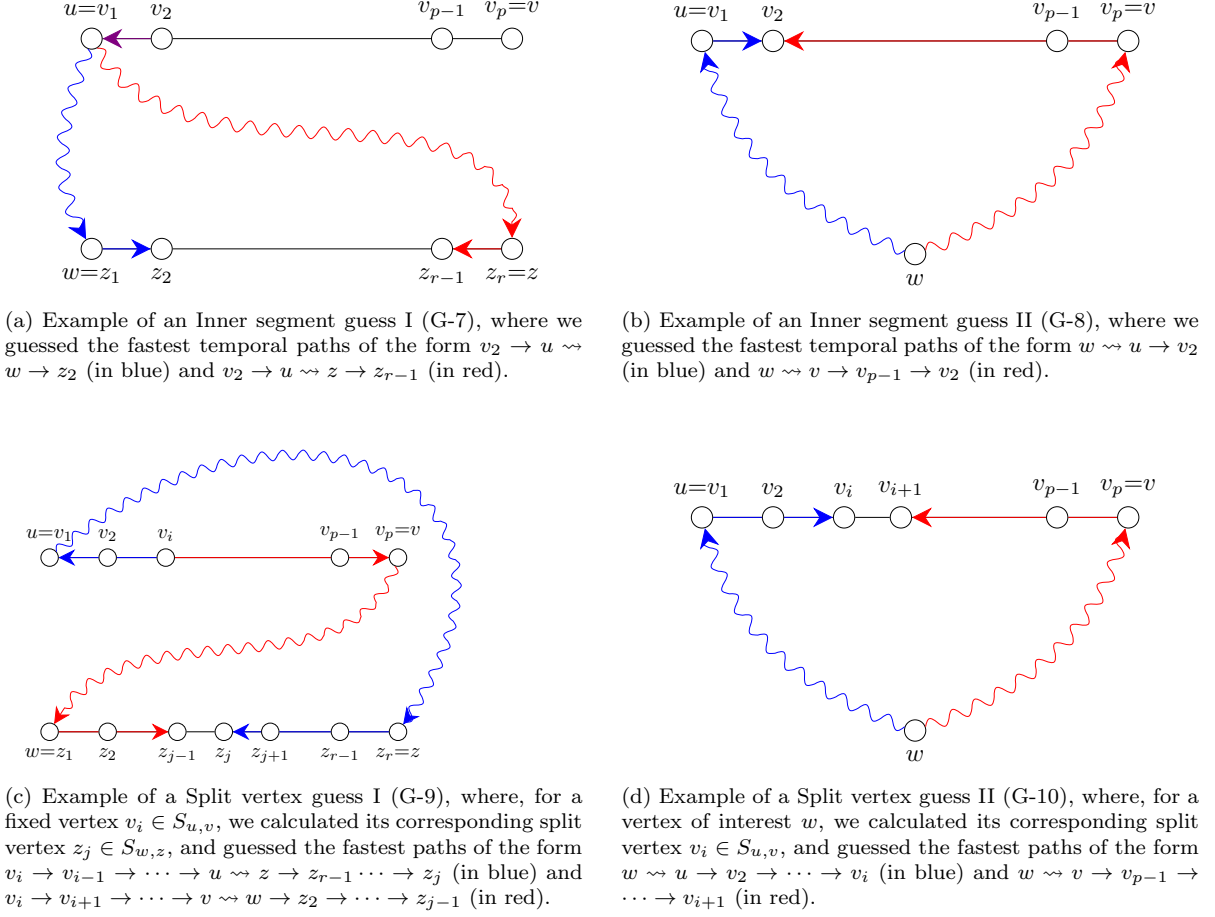


Figure 8: Illustration of the guesses G-7, G-8, G-9, and G-10.

travel delay from u to z at vertex v , denoted with τ_v^{uz} , as the difference of the labels of e_2 and e_1 , where we subtract the value of the label of e_1 from the label of e_2 , modulo Δ , if the labels are different, or we set it to Δ if the labels are equal. More specifically:

- if $\lambda(e_1) \neq \lambda(e_2)$ we have

$$\tau_v^{uz} \equiv \lambda(e_2) - \lambda(e_1) \pmod{\Delta}, \quad (2)$$

- if $\lambda(e_1) = \lambda(e_2)$ we have

$$\tau_v^{uz} = \Delta.$$

Intuitively, the value of τ_v^{uz} represents how long a temporal path waits at vertex v when first taking edge $e_1 = uv$ and then edge $e_2 = vz$.

From the above definition and the definition of the duration of the temporal path P we get the following two observations.

Observation 1. Let $P = (v_0, v_1, \dots, v_p)$ be the underlying path of the temporal path (P, λ) from v_0 to v_p . Then $d(P, \lambda) = \sum_{i=1}^{p-1} \tau_{v_i}^{v_{i-1}v_i} + 1$.

Proof. For the simplicity of the proof denote $t_i = \lambda(v_{i-1}v_i)$, and suppose that $t_i \leq t_{i+1}$, for all

$i \in \{1, 2, 3, \dots, p\}$. Then

$$\begin{aligned}
\sum_{i=1}^{p-1} \tau_{v_i}^{v_{i-1}v_i} + 1 &= \sum_{i=1}^{p-1} (t_{i+1} - t_i) + 1 \\
&= (t_2 - t_1) + (t_3 - t_2) + \dots + (t_p - t_{p-1}) + 1 \\
&= t_p - t_1 + 1 \\
&= d(P, \lambda)
\end{aligned}$$

Now in the case when $t_i > t_{i+1}$ we get that $\tau_{v_i}^{v_{i-1}v_{i+1}} = \Delta + t_{i+1} - t_i$. At the end this still results in the correct duration as the last time we traverse the path P is not exactly t_p but $\alpha\lambda + t_p$, for some integer α . \square

We also get the following two observations.

Observation 2. *Let (G, λ) be a temporal graph satisfying the conditions of the SIMPLE TGR problem. For any two incident edges $e_1 = uv$ and $e_2 = vz$ on vertices $u, v, z \in V$, with $e_1 \cap e_2 = \{v\}$, we have $\tau_v^{zu} = \Delta - \tau_v^{uz} \pmod{\Delta}$.*

Proof. Let $e_1 = uv$ and $e_2 = vz$ be two edges in G for which $e_1 \cap e_2 = \{v\}$. If $\lambda(e_2) = \lambda(e_1)$ then the statement follows immediately by Definition 4, as in this case $\tau_v^{zu} = \tau_v^{uz} = \Delta$. If $\lambda(e_2) \neq \lambda(e_1)$, then it follows by Definition 4 that $\tau_v^{uz} \equiv \lambda(e_2) - \lambda(e_1) \pmod{\Delta}$ and $\tau_v^{zu} \equiv \lambda(e_1) - \lambda(e_2) \pmod{\Delta}$. Summing now both equations we get $\tau_v^{uz} + \tau_v^{zu} \equiv \lambda(e_2) - \lambda(e_1) + \lambda(e_1) - \lambda(e_2) \pmod{\Delta}$, and therefore $\tau_v^{uz} + \tau_v^{zu} \equiv 0 \pmod{\Delta}$, which is equivalent as saying $\tau_v^{uz} \equiv -\tau_v^{zu} \pmod{\Delta}$ or $\tau_v^{zu} = \Delta - \tau_v^{uz} \pmod{\Delta}$. \square

Observation 3. *Let P be the underlying path of a fastest temporal path from u to v , where $e_1, e_p \in P$ are its first and last edge, respectively. Then, knowing the label $\lambda(e_1)$ of the first edge and the duration $d(P, \lambda)$ of the temporal path (P, λ) , we can uniquely determine the label $\lambda(e_p)$ of the last edge of P . Symmetrically, knowing $\lambda(e_p)$ and $d(P, \lambda)$, we can uniquely determine $\lambda(e_1)$.*

The correctness of the above statement follows directly from Definition 2. This is because the duration of (P, λ) is calculated as the difference of labels of last and first edge plus 1, where the label of last edge is considered with some delta periods, i. e., $d(P, \lambda) = p_i\Delta + \lambda(e_p) - \lambda(e_1) + 1$, for some $p_i \geq 0$. Therefore $d(P, \lambda) \pmod{\Delta} \equiv (\lambda(e_p) - \lambda(e_1) + 1) \pmod{\Delta}$. Note that if $\lambda(e_1)$ and $\lambda(e_p)$ are both unknown, then we can determine one with respect to the other.

In the following we prove that knowing the structure (the underlying path) of a fastest temporal path P from a vertex of interest u to a vertex of interest v , results in determining the labeling of each edge in a fastest temporal path from u to v (with the exception of some constant number of edges), with respect to the label of the first edge. More precisely, if path P from u to v is a segment, then we can determine labels of all edges as a function of the label of the first edge. If P consists of ℓ segments, then we can determine the labels of all but $\ell - 1$ edges as a function of the label of the first edge. For the exact formulation and proofs see Lemmas 1 and 2.

Lemma 1. *Let $u, v \in U$ be two arbitrary vertices of interest and suppose that $P = (u = v_1, v_2, \dots, v_p = v)$, where $p \geq 2$, is a path in G' , which is also the underlying path of a fastest temporal path from u to v . Moreover, suppose also that P is a segment. We can determine the labeling λ of every edge in P with respect to the label $\lambda(uv_2)$ of the first edge.*

Proof. We claim that u reaches all of the vertices in P the fastest, when traveling along P (i.e., by using a subpath of P). To prove this suppose for the contradiction that there is a vertex $v_i \in P \setminus \{u, v\}$, that is reached from v on a path different than $P_i = (u, v_2, v_3, \dots, v_i)$ faster than through P_i . Since the only vertices of interest of P are u and v , it follows that all other vertices on P are of degree 2. Then the only way to reach v_i from u , that differs from P , would be to go from u to v using a different path P_2 , and then go from v to $v_{p-1}, v_{p-2}, \dots, v_i$. But since P is a fastest temporal path from u to v , we get that $d(P_2) \geq d(P)$ and $d(P_2 \cup (v, v_{p-1}, \dots, v_i)) > d(P) > d(P_i)$.

Now, to determine the labeling λ of the path P we use the property that a fastest temporal path from u to any $v_i \in P$ is a subpath of P . We set the label of the first edge of P to be a constant $c \in [\Delta]$ and use Observation 3 to label all remaining edges, where the duration from u to v_i equals to D_{u,v_i} . This gives us a unique labeling λ of P where the label of each edge of P is a function of c . \square

Lemma 2. *Let $u, v \in U$ be two arbitrary vertices of interest and suppose that $P = (u = v_1, v_2, \dots, v_p = v)$, where $p \geq 2$, is a path in G' , which is also the underlying path of a fastest temporal path from u to v . Let $\ell_{u,v} \geq 1$ be the number of vertices of interest in P different to u, v , namely $\ell_{u,v} = |(P \setminus \{u, v\}) \cap U|$. We can determine the labeling λ of all but $\ell_{u,v}$ edges of P , with respect to the label $\lambda(uv_2)$ of the first edge, such that the labeling λ respects the values from D .*

For the proof of the above lemma, we first prove a weaker statement, for which we need to introduce some extra definitions and fix some notations. In the following we only consider *wasteless* temporal paths. We call a temporal path $P = ((e_1, t_1), \dots, (e_k, t_k))$ a *wasteless* temporal path, if for every $i = 1, 2, \dots, k-1$, we have that t_{i+1} is the first time after t_i that the edge e_{i+1} appears.

Let $u, v \in V$, and let $t \in \mathbb{N}$. Given that a temporal path starts within the period $[t, t + \Delta - 1]$, we denote with $A_t(u, v)$ the *arrival* of the fastest path in (G, λ) from u to v , and with $A_t(u, v, P)$, the *arrival* along path P in (G, λ) from u to v . Whenever $t = 1$, we may omit the index t , i.e., we may write $A(u, v, P) = A_1(u, v, P)$ and $A(u, v) = A_1(u, v)$.

Suppose now that we know the underlying path $P_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ of a fastest temporal path between vertices of interest u and v in G' . Let $v_i \in U$ with $u \neq v_i \neq v$ be a vertex of interest on the path $P_{u,v}$. Suppose that v_i is reached the fastest from u by a path $P = (u = u_1, u_2, \dots, u_{j-1}, v_i)$. We split the path $P_{u,v}$ into a path $Q = (u = v_1, v_2, \dots, v_i)$ and $R = (v_i, v_{i+1}, \dots, v_p = v)$ (for details see Fig. 9).

From the above, we get the following assumptions:

1. $d(u, v_i) = d(u, v_i, P) \leq d(u, v_i, Q)$, and
2. $d(u, v_p) = d(u, v_p, Q \cup R) \leq d(u, v_p, P \cup R)$.

In the remainder, we denote with δ_0 the difference $d(u, v_i, Q) - d(u, v_i) \geq 0$. Let $t_{v_2} \in [\Delta]$ be the label of the edge uv_2 , and denote by t_{u_2} the appearance of the edge uu_2 within the period $[t_{v_2}, t_{v_2} + \Delta - 1]$. Note that $1 \leq t_{v_2} \leq \Delta$ and that

$$t_{v_2} \leq t_{u_2} \leq t_{v_2} + \Delta - 1 \leq 2\Delta - 1. \quad (3)$$

From Assumption 1 we get

$$\delta_0 = d(u, v_i, Q) - d(u, v_i) = A_{t_{v_2}}(u, v_i, Q) - A_{t_{v_2}}(u, v_i, P) + (t_{u_2} - t_{v_2})$$

and thus

$$A_{t_{v_2}}(u, v_i, P) - A_{t_{v_2}}(u, v_i, Q) = t_{u_2} - (t_{v_2} + \delta_0). \quad (4)$$

We use all of the above discussion, to prove the following lemma.

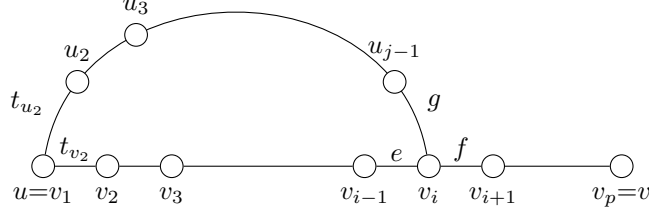


Figure 9: An example of the situation in Lemma 2, where we assume that a fastest temporal path from u to v is $P_{u,v} = (u = v_1, v_2, \dots, v_p)$, and a fastest temporal path from u to v_i is $P = (u, u_2, u_3, \dots, v_i)$. We denote with $Q = (u = v_1, v_2, \dots, v_i)$ and with $R = (v_i, v_{i+1}, \dots, v_p = v)$.

Lemma 3. *If $t_{u_2} \neq t_{v_2}$, then $\delta_0 \leq \Delta - 2$ and $t_{u_2} \geq t_{v_2} + \delta_0 + 1$.*

Proof. First assume that $\delta_0 \geq \Delta - 1$. Then, it follows by Eq. (4) that $A_{t_{v_2}}(u, v_i, P) - A_{t_{v_2}}(u, v_i, Q) \leq t_{u_2} - t_{v_2} - \Delta + 1 \leq 0$, and thus $A_{t_{v_2}}(u, v_i, P) \leq A_{t_{v_2}}(u, v_i, Q)$. Therefore, since we can traverse path P from u to v_i by departing at time $t_{u_2} \geq t_{v_2} + 1$ and by arriving no later than traversing path R , we have that $d(u, v_p, P \cup R) < d(u, v_p, Q \cup R)$, which is a contradiction to the second initial assumption. Therefore $\delta_0 \leq \Delta - 2$.

Now assume that $t_{v_2} + 1 \leq t_{u_2} \leq t_{v_2} + \delta_0$. Then, it follows by Eq. (4) that $A_{t_{v_2}}(u, v_i, P) \leq A_{t_{v_2}}(u, v_i, Q)$ which is, similarly to the previous case, a contradiction. Therefore $t_{u_2} \geq t_{v_2} + \delta_0 + 1$. \square

The next corollary follows immediately from Lemma 3.

Corollary 3. *If $t_{u_2} \neq t_{v_2}$, then $1 \leq A_{t_{v_2}}(u, v_i, P) - A_{t_{v_2}}(u, v_i, Q) \leq \Delta - 1 - \delta_0$.*

We are now ready to prove the following result.

Lemma 4. $d(u, v_{i-1}, P \cup \{v_i v_{i-1}\}) > d(u, v_{i-1}, Q \setminus \{v_i v_{i-1}\})$.

Proof. Let $e \in [\Delta]$ be the label of the edge $v_{i-1}v_i$, and let $f \in [e + 1, e + \Delta]$ be the time of the first appearance of the edge $v_i v_{i+1}$ after time e . Let $A_{t_{v_2}}(u, v_i, Q) = x\Delta + e$. Then $A_{t_{v_2}}(u, v_{i+1}, Q \cup \{v_i v_{i+1}\}) = x\Delta + f$. Furthermore let g be such that $A_{t_{v_2}}(u, v_i, P) = x\Delta + g$.

Case 1: $t_{u_2} \neq t_{v_2}$. Then Corollary 3 implies that $e + 1 \leq g \leq e + (\Delta - 1 - \delta_0)$. Assume that $g < f$. Then, we can traverse path P from u to v_i by departing at time $t_{u_2} \geq t_{v_2} + 1$ and by arriving at most at time $x\Delta + f - 1$, and thus $d(u, v_p, P \cup R) < d(u, v_p, Q \cup R)$, which is a contradiction to the second initial assumption. Therefore $g \geq f$. That is,

$$e + 1 \leq f \leq g \leq e + (\Delta - 1 - \delta_0).$$

Consider the path $P^* = P \cup \{v_i v_{i-1}\}$. Assume that we start traversing P^* at time t_{u_2} . Then we arrive at v_i at time $x\Delta + g$, and we continue by traversing edge $v_i v_{i-1}$ at time $(x + 1)\Delta + e$. That is, $d(u, v_{i-1}, P^*) = (x + 1)\Delta + e - t_{u_2} + 1$.

Now consider the path $Q^* = Q \setminus \{v_i v_{i-1}\}$. Let $h \in [1, \Delta]$ be such that $A_{t_{v_2}}(u, v_{i-1}, Q^*) = x\Delta + e - h$. That is, if we start traversing Q^* at time t_{v_2} , we arrive at v_{i-1} at time $x\Delta + e - h$, i.e., $d(u, v_{i-1}, Q^*) = x\Delta + e - h - t_{v_2} + 1$. Summarizing, we have:

$$\begin{aligned} d(u, v_{i-1}, P^*) - d(u, v_{i-1}, Q^*) &= \Delta + h - (t_{u_2} - t_{v_2}) \\ &\geq \Delta + 1 - (\Delta - 1) > 0, \end{aligned}$$

where the last inequality follows by (3). This proves the statement of the lemma.

Case 2: $t_{u_2} = t_{v_2}$. Then, it follows by Equation (4) that $A_{t_{v_2}}(u, v_i, P) = A_{t_{v_2}}(u, v_i, Q) - \delta_0 \leq A_{t_{v_2}}(u, v_i, Q)$. Therefore $g \leq e$. Similarly to Case 1 above, consider the paths $P^* = P \cup \{v_i v_{i-1}\}$ and $Q^* = Q \setminus \{v_i v_{i-1}\}$. Assume that we start traversing P^* at time $t_{u_2} = t_{v_2}$. Then we arrive at v_i at time $x\Delta + g$, and we continue by traversing edge $v_i v_{i-1}$, either at time $(x+1)\Delta + e$ (in the case where $g = e$) or at time $x\Delta + e$ (in the case where $g \neq e$). That is, $d(u, v_{i-1}, P^*) \geq x\Delta + e - t_{v_2} + 1$.

Similarly to Case 1, let $h \in [1, \Delta]$ be such that $A_{t_{v_2}}(u, v_{i-1}, Q^*) = x\Delta + e - h$. That is, if we start traversing Q^* at time t_{v_2} , we arrive at v_{i-1} at time $x\Delta + e - h$, i. e., $d(u, v_{i-1}, Q^*) = x\Delta + e - h - t_{v_1} + 1$. Summarizing, we have:

$$d(u, v_{i-1}, P^*) - d(u, v_{i-1}, Q^*) \geq h \geq 1,$$

which proves the statement of the lemma. \square

From the above it follows that if P is a fastest path from u to v , then all vertices of P , with the exception of vertices of interest $v_i \in P \setminus \{u, v\}$, are reached using the same path P . We use this fact in the following proof.

Proof of Lemma 2. For every vertex of interest $v_i \in U \cap (P \setminus \{u, v\})$ we have two options. First, when a fastest temporal path P' from u to v_i is a subpath of P . In this case, we determine the labeling of P' using Lemma 1. Second, when a fastest temporal path P' from u to v_i is not a subpath of P . In this case, we know exactly how to label all of the edges of P , with the exception of edges from $v_{i-1}v_i$, that are incident to v_i in P . \square

Lemma 5. *Let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ be a segment in G . If $S_{u,v}$ is of length at least 5 ($p > 5$) then it is impossible for an inner edge $f = v_i v_{i+1}$ from $S_{u,v} \setminus \{u, v\}$ (where f is an edge that is not incident to a vertex from U) to not be a part of any fastest temporal path, of length at least 2 between vertices in $S_{u,v}$. In other words, there must exist a pair $v_j, v_{j'} \in S_{u,v}$ s. t., a fastest temporal path from v_j to $v_{j'}$ passes through f . If $S_{u,v}$ is of length 4 then all fastest temporal paths of length 2 avoid the inner edge f if and only if f has the same label as both of the edges incident to it, while the label of the last remaining edge is determined with respect to $\lambda(f)$.*

Proof. For an easier understanding and better readability, we present the proof for $S_{u,v}$ of length 5. The case where $S_{u,v}$ is longer easily follows from the presented results.

Let $S_{u,v} = (u = v_1, v_2, v_3, v_4, v_5, v_6 = v)$. We distinguish two cases, first when $f = v_2 v_3$ (note that the case with $f = v_4 v_5$ is symmetrical), and second when $f = v_3 v_4$. Throughout the proof we denote with t_i the label of edge $v_i v_{i+1}$. Suppose for the sake of contradiction, that none of the fastest temporal paths between vertices of $S_{u,v}$ traverses the edge f .

Case 1: $f = v_2 v_3$. Let us observe the case of the fastest temporal paths between v_1 and v_3 . Denote with $Q = (v_1, v_2, v_3)$ and with $P' = (v_3, v_4, v_5, v_6)$. From our proposition, it follows that

- a fastest temporal path P^+ from v_1 to v_3 is of the following form $P^+ = v_1 \rightsquigarrow v_6 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3$, and
- a fastest temporal path P^- from v_3 to v_1 is of the following form $P^- = v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightsquigarrow v_1$.

It follows that $d(v_1, v_3, P^+) \leq d(v_1, v_3, Q)$, and $d(v_3, v_1, P^-) \leq d(v_3, v_1, Q)$. Note that $d(v_1, v_3, P^+) \geq 1 + d(v_6, v_3, P')$, and by the definition $d(v_6, v_3, P') = 1 + (t_4 - t_5)\Delta + (t_3 - t_4)\Delta$, where $(t_i - t_j)\Delta$ denotes the difference of the labels t_i, t_j of two consecutive edges modulo Δ and

we define $(0)_\Delta = \Delta$. Similarly holds for $d(v_3, v_1, P^-)$. Summing up both of the above equations we get

$$\begin{aligned}
d(v_1, v_3, P^+) + d(v_3, v_1, P^-) &\leq d(v_1, v_3, Q) + d(v_3, v_1, Q) \\
1 + d(v_6, v_3, P') + 1 + d(v_3, v_6, P') &\leq d(v_1, v_3, Q) + d(v_3, v_1, Q) \\
3 + (t_4 - t_5)_\Delta + (t_3 - t_4)_\Delta + 1 + (t_4 - t_3)_\Delta + (t_5 - t_4)_\Delta &\leq 1 + (t_2 - t_1)_\Delta + 1 + (t_1 - t_2)_\Delta \\
(t_4 - t_5)_\Delta + (t_5 - t_4)_\Delta + (t_4 - t_3)_\Delta + (t_3 - t_4)_\Delta + 2 &\leq (t_2 - t_1)_\Delta + (t_1 - t_2)_\Delta.
\end{aligned} \tag{5}$$

Note that if $t_i \neq t_j$ we get that the sum $(t_i - t_j)_\Delta + (t_j - t_i)_\Delta$ equals exactly Δ , and if $t_i = t_j$ the sum equals 2Δ . This follows from the definition of travel delays at vertices (see Observation 2). Therefore we get from Eq. (5), that the right part is at most 2Δ , while the left part is at least $2\Delta + 1$, for any relation of labels t_1, t_2, \dots, t_5 , which is a contradiction.

Case 2: $f = v_3v_4$. Here we consider the fastest paths between vertices v_2 and v_4 . By similar arguments as above we get

$$(t_5 - t_1)_\Delta + (t_4 - t_5)_\Delta + (t_5 - t_4)_\Delta + (t_1 - t_5)_\Delta + 2 \leq (t_3 - t_2)_\Delta + (t_2 - t_3)_\Delta,$$

which is impossible.

In the case when $S_{u,v}$ is longer, we would get an even bigger number on the left-hand side of Eq. (5), so we conclude that in all of the above cases, it cannot happen that all fastest paths of length 2, between vertices in $S_{u,v}$, avoid edge f .

Let us observe now the case when $S_{u,v} = (u = v_1, v_2, v_3, v_4, v_5 = v)$ is of length 4. Let $f = v_2v_3$ (the case with $f = v_3v_4$ is symmetrical). Suppose that the fastest temporal paths between v_1 and v_3 do not use the edge f . We denote with R^+ the fastest path from v_1 to v_3 , which is of the form $u \rightsquigarrow v \rightarrow v_4 \rightarrow v_3$, and similarly with R^- the fastest path from v_3 to v_1 , which is of the form $v_3 \rightarrow v_4 \rightarrow v \rightsquigarrow u$. We denote with $R' = (v_3, v_4, v_5)$ and with $S = (v_1, v_2, v_3)$. Again we get the following.

$$\begin{aligned}
d(v_1, v_3, R^+) + d(v_3, v_1, R^-) &\leq d(v_1, v_3, S) + d(v_3, v_1, S) \\
1 + d(v_5, v_3, R') + 1 + d(v_3, v_5, R') &\leq d(v_1, v_3, S) + d(v_3, v_1, S) \\
(t_3 - t_4)_\Delta + (t_4 - t_3)_\Delta + 2 &\leq (t_2 - t_1)_\Delta + (t_1 - t_2)_\Delta.
\end{aligned}$$

The only case when the equation has a valid solution is when $t_1 = t_2$ and $t_3 \neq t_4$, as in this case the left hand side evaluates to $\Delta + 2$, while the right side evaluates to 2Δ . Repeating the analysis for the fastest paths between v_2 and v_4 , we conclude that the only valid solution is when $t_2 = t_3$ and $t_1 \neq t_4$. Altogether, we get that f is not a part of any fastest path of length 2 in $S_{u,v}$ if and only if the label of edge f is the same as the labels on the edges incident to it, while the last remaining edge has a different label. Note now that a fastest temporal path from v_2 to v_4 must first use the edge uv_2 and finish with the edge v_4v_5 , and it has to be of duration D_{v_2, v_4} . Using Lemma 1 we determine the label of the edge v_4v_5 with respect to $\lambda(f)$. \square

We now present some properties involving the vertices from Z , that form the trees in $G[Z]$.

Lemma 6. *Let $v \in V(G')$ be the clip vertex of the tree T_v in $G[Z \cup \{v\}]$, and let $z \in N_{T_v}(v)$ be an arbitrary child of v in T_v . Among all neighbors of v in G' , let w be the one that is on the smallest duration away from z with respect to the values of D . In other words, $w \in N_{G'}(v)$ such that $D_{z,w} \leq D_{z,w'}$ for all $w' \in N_{G'}(v)$. Then, the path $P^* = (z, v, w)$ represents the unique fastest temporal path from z to w . Moreover, we can determine all labels of the tree T_v with respect to the label $\lambda(vw)$.*

Proof. Suppose for contradiction that there exists a path $P^{**} \neq P^*$ from z to w such that $d(P^{**}, \lambda) \leq d(P^*, \lambda)$. By the structure of G , it follows that P^{**} passes through the clip vertex v of T_v (as this is the only neighbor of z in G'), continues through a vertex $w' \in N_{G'}(v) \setminus \{w\}$, and through some other vertices u_1, u_2, \dots, u_j in G ($j \geq 0$) before finishing in w . Therefore, $P^{**} = (z, v, w', u_1, u_2, \dots, u_j, w)$. Now, since $D_{z,w} \leq D_{z,w'}$ by assumption, the first part of P^{**} from z to w' takes at least $D_{z,w'}$ time, and thus it takes at least $D_{z,w}$ time. Since $w \neq w'$, we need at least one more time step (one more edge) to traverse from w' to reach w . Therefore, $d(P^{**}, \lambda) \geq D_{z,w} + 1$ which implies that P^{**} is not a fastest temporal path from z to w . Therefore, the only fastest temporal path from z to w is $P^* = (z, v, w)$.

For the second part, knowing that the duration of P^* is $D_{z,w}$, we can determine the label of the edge zv with respect to the label $\lambda(vw)$ (see Observation 3). Furthermore, using the algorithm for trees (see Theorem 3), we can now determine all the labels on the edges of T_v with respect to the same label $\lambda(vw)$. \square

Lemma 7. *Let $x \in V(G')$ be the clip vertex of the tree T_x in $G[Z \cup \{x\}]$, where $x \notin U$. Let v_1 and v_2 be the two neighbors of x in G' . Then the labels of the tree T_x can be determined with respect to $\lambda(v_1x)$ and $\lambda(xv_2)$.*

Proof. First observe that since x is not a vertex of interest it must be a part of some segment $S_{u,w}$, where $u, v \in U$ and $x \neq u \neq v$. Therefore, x is of degree 2 in G' . Let $z \in V(T_x)$ be a child of x in T_x , i.e., a vertex in the first layer of the tree T_x . We observe the values D_{z,v_1}, D_{z,v_2} and distinguish the following cases.

First, $D_{z,v_1} = D_{z,v_2}$. Then, using Lemma 6 we conclude that the fastest temporal paths from z to v_1 and from z to v_2 are of length two. We know that these two paths consist of the edge zx and xv_1, xv_2 , respectively. This allows us to determine the label of the edge zx (and consequently all other edges of T_x) with respect to $\lambda(xv_1)$ and $\lambda(xv_2)$.

Second, $D_{z,v_1} \neq D_{z,v_2}$. Let us denote with $t_1 = \lambda(xv_1), t_2 = \lambda(xv_2)$ and $t_3 = \lambda(zx)$. W.l.o.g. suppose that $\min\{t_1, t_2, t_3\} = t_3$, and that $D_{z,v_1} > D_{z,v_2}$ (the other case is analogous). It follows that $t_1 > t_2$. We want to now prove that the inequality $D_{v_1,z} < D_{v_2,z}$ holds. Suppose for the contradiction that the inequality is false. Then $D_{v_2,z} < D_{v_1,z} \leq (\Delta + t_3 - t_1)$. This implies that a fastest temporal path from v_2 to z cannot use the path (v_1, x, z) , and is therefore of form (v_2, x, z) . By definition, the duration of this path is $D_{v_2,z} = \Delta + t_3 - t_2 + 1$. But since $t_1 > t_2$ it follows that $(\Delta + t_3 - t_2) + 1 > (\Delta + t_3 - t_1) + 1$. We also know that $D_{v_1,z} \leq (\Delta + t_3 - t_1) + 1$. This implies that $D_{v_2,z} > D_{v_1,z}$, a contradiction.

Knowing $D_{z,v_1} > D_{z,v_2}$ we can determine the label of edge zx (and consequently all other edges of T_x) with respect to $\lambda(xv_2)$, and similarly knowing $D_{v_1,z} < D_{v_2,z}$ we determine the label of edge zx (and all other edges of T_x) with respect to $\lambda(xv_1)$. \square

Remember, in the case where the clip vertex u of the tree T_u in $G[Z \cup \{u\}]$ is a vertex of interest, we split the vertices in the first layer of T_u into at most $|N_{G'}(u)|$ equivalence classes (as explained in Section 3.2.1). Let us now show the following important property of these equivalence classes.

Lemma 8. *Let $u \in V(G')$ be the clip vertex of the tree T_u in $G[Z \cup \{u\}]$, where $u \in U$, and let $z_1, z_2 \in V(T_u)$ be in the same equivalence class of the tree T_u . Then, for every vertex $y \neq u$ in $V(G')$, the set of the fastest paths from z_1 to y (resp. from y to z_1) on the edges in G' coincides with the set of the fastest paths from z_2 to y (resp. from y to z_2) on the edges in G' .*

Proof. Let $y \neq u$ be a vertex in $V(G')$. Denote with P_1 the underlying path of a fastest temporal path from z_1 to y , which consists of the edge z_1u and the path P from u to y . Similarly, let Q_2 be the underlying path of a fastest temporal path from z_2 to y , consisting of the edge z_2u

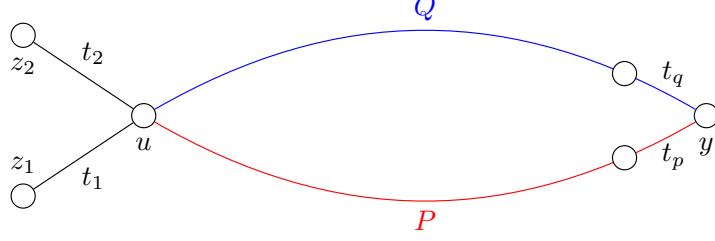


Figure 10: An example of the situation in Lemma 8. We have two paths Q (in blue) and P (in red) from u to y . We assume that a fastest temporal path from z_1 to y is $P_1 = \{z_1u\} \cup P$, and a fastest temporal path from z_2 to y is $Q_2 = \{z_2u\} \cup Q$. We denote with $Q_1 = \{z_1u\} \cup Q$ another temporal path from z_1 to y , with $P_2 = \{z_2u\} \cup P$ another temporal path from z_2 to y . The labels on the edges z_1u, z_2u are t_1, t_2 , respectively. Similarly, the labels of the last edges of paths P and Q are t_p, t_q , respectively.

together with the path Q from u to y . Define P_2 as the second path from z_2 to y that first uses the edge z_2u and then the path P . Similarly, Q_1 represents the second path from z_1 to y that first uses the edge z_1u and then the path Q . Our objective is to demonstrate that either $P = Q$ or that $d(P_1, \lambda) = d(Q_1, \lambda)$ and $d(P_2, \lambda) = d(Q_2, \lambda)$. This implies that z_1 and z_2 use temporal paths that coincide on the vertices of $V(G) \setminus V(T_u)$ to reach y . For an illustration see Fig. 10.

Let us set the label of the edge z_1u to t_1 , the label of z_2u to t_2 , the label of the last edge of the path P as t_p and the label of the last edge of the path Q as t_q . By the definition, since P_1 represents a fastest temporal path from z_1 to y we get that $D_{z_1,y} = t_p - t_1 + c_p\Delta$, where $c_p \in \mathbb{N}$. Similarly, for the path Q_2 it holds that $D_{z_2,y} = t_q - t_2 + c_q\Delta$ with $c_q \in \mathbb{N}$. Note that the difference between the first label of P (resp. Q) with t_1 and t_2 is smaller than Δ , or the difference (with at least one t_1, t_2) is Δ if and only if the first label of P and the first label of Q are the same.

We want to first show that $c_p = c_q$. Let us assume, for the sake of contradiction, that this is not the case, and suppose that $c_p > c_q$ (the case with $c_q > c_p$ is analogous). Then $c_p \leq c_q + 1$. Now, since z_1 and z_2 are in the same equivalence class and by the definition of the duration of a temporal path we get that $d(P_2, \lambda) = t_p - t_2 + c_p\Delta \leq t_p - t_2 + (c_q - 1)\Delta$. Because Q_2 is the fastest path from z_2 to y we have also that $d(P_2, \lambda) \geq D_{z_2,y}$, which gives us $t_p - t_2 + (c_q - 1)\Delta \geq t_q - t_2 + c_q\Delta$. This is equivalent to $t_p \geq t_q + \Delta$, but since $t_p, t_q \in [\Delta]$ this cannot happen. Therefore, we conclude that $c_p = c_q$.

Now, we want to show also, that $t_p = t_q$. Let us assume, for the sake of contradiction, that this is not the case, and suppose that $t_p > t_q$ (the case with $t_q > t_p$ is analogous). Then the duration of the path Q_1 is $d(Q_1, \lambda) = t_q - t_1 + c_q\Delta$ since $c_q = c_p$. Above we proved that $c_p = c_q$. We also know that $d(Q_1, \lambda) \geq d(P_1, \lambda)$ as P_1 is the fastest path from z_1 to y . All of this results in $t_q - t_1 + c_p\Delta \geq t_p - t_1 + c_p\Delta$ implying $t_q \geq t_p$, a contradiction. Therefore, $t_p = t_q$.

We proved that P and Q are the same or, if they are different, then P_1 and Q_1 are of the same duration and are both fastest paths from z_1 to y (the same holds for z_2).

The proof of the statement for the fastest temporal paths in the other direction, namely starting at y and reaching z_1 and z_2 , is done analogously. \square

Observation 4. Let $v \in V(G')$ be the clip vertex of the tree T_v in $G[Z \cup \{v\}]$, $z \in N_{T_v}(v)$ be a child of v in T_v , and let z' be a descendant of z in T_v . Let $x \in V(G) \setminus V(T_v)$ be an arbitrary vertex. Denote by P_z the underlying path of a fastest temporal path from z to x (resp. from x to z), and denote by Q the (unique) path between z and z' in T_v . Then, the symmetric difference between P_z and Q is the underlying path of a fastest path from z' to x (resp. from x to z').

The correctness of the above observation is a consequence of Lemma 8 and of the fact that P_z and $P_{z'}$ leave the tree T_v using the same edge zv .

3.2.4. Adding constraints and variables to the ILP

We start by analyzing the case where we want to determine the labels on fastest temporal paths between vertices of interest. We proceed in the following way. Let $u, v \in U$ be two vertices of interest and let $P_{u,v}$ be a fastest temporal path from u to v . If $P_{u,v}$ is a segment we determine all the labels of edges of $P_{u,v}$, with respect to the label of the first edge (see Lemma 1). In the case when $P_{u,v}$ is a sequence of ℓ segments, we determine all but $\ell - 1$ labels of edges of $P_{u,v}$, with respect to the label of the first edge (see Lemma 2). We call these $\ell - 1$ edges, *partially determined* edges. After repeating this step for all pairs of vertices in U , the edges of fastest temporal paths from u to v , where $u, v \in U$, are determined with respect to the label of the first edge of each path, or are partially determined. If a fastest temporal path between two vertices $u, v \in U$ is just an edge e , then we treat it as being determined, since it gets assigned a label $\lambda(e)$ with respect to itself. All other edges in G' are called the *not yet determined* edges. Note that the not yet determined edges are exactly the ones that are not a part of any fastest temporal path between any two vertices in U .

Now we want to relate the not-yet-determined segments with the determined ones. Let $S_{u,v}$ and $S_{w,z}$ be two segments. At the beginning, we have guessed the fastest path from v_i to all vertices in $S_{w,z}$ (see guess G-9). We did this by determining which vertices z_j, z_{j+1} in $S_{w,z}$ are furthest away from v_i (remember we can have the case when $z_j = z_{j+1}$), and then we guessed how the path from v_i leaves the segment $S_{u,v}$ (i.e., either through the vertex u or v), and then how it reaches z_j (in the case when $z_j \neq z_{j+1}$ there is a unique way, when $z_j = z_{j+1}$ we determined which of the vertices w or z is on the fastest path). W.l.o.g. assume that we have guessed that the fastest path from v_i to z_j passes through w and z_{j-1} . Then a fastest temporal path from v_i to z_{j+1} passes through z . And all fastest temporal paths from v_i to any $z_{j'} \in S_{w,z}$ use all of the edges in $S_{w,z}$ with the exception of the edge $z_j z_{j+1}$. Using this information and Observation 3, we can determine the labels on all edges, with respect to the first or last label from the segment $S_{u,v}$, with the exception of the edge $z_j z_{j+1}$. Therefore, all edges of $S_{w,z}$ but $z_j z_{j+1}$ become determined. Since we repeat that procedure for all pairs of segments, we get that for a fixed segment $S_{w,z}$ we end up with a not yet determined edge $z_j z_{j+1}$ if and only if this is a not yet determined edge in relation to every other segment $S_{u,v}$ and its fixed vertex v_i . We repeat this procedure for all pairs of segments. Each specific calculation takes linear time. Since there are $O(k^2)$ segments, the whole calculation takes $O(k^4)$ time.

From the above procedure (where we were determining labels of edges of segments with each other) we conclude that all of the edges $e_i = v_i v_{i+1}$ of a segment $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ are in one of the following relations. First, where all of the edges are determined with respect to each other. Second, where there are some edges e_1, e_2, \dots, e_{i-1} , whose label is determined with respect to the label $\lambda(e_1)$, there is an edge $f = e_i = v_i v_{i+1}$ which is not yet determined, and then there follow the edges $e_{i+1}, e_{i+2}, \dots, e_{p-1}$, whose labels are determined with respect to $\lambda(e_{p-1})$. Third, where the first e_1, \dots, e_{i-1} edges are determined with respect to the $\lambda(e_1)$ and all of the remaining edges $e_i, e_{i+1}, \dots, e_{p-1}$ are determined with respect to the $\lambda(e_{p-1})$. We want to now determine all of the edges in such segment $S_{u,v}$ with respect to just one edge (either the first or the last one). In the second case, we use the fact that at least one of the temporal paths between v_{i-1} and v_{i+1} has to pass through f , to determine $\lambda(f)$ with respect to $\lambda(e_{i-1})$ (and consequently $\lambda(e_1)$), and similarly, one of the temporal paths between v_i and v_{i+2} has to pass through f , which determines $\lambda(f)$ with respect to $\lambda(e_{i+1})$ (and consequently $\lambda(e_{p-1})$). In the third case, knowing the temporal paths between v_{i-1} to v_{i+1} results in determining the label of $\lambda(e_{i-1})$ with $\lambda(e_i)$, which consequently relates labels of all of the edges of the segment against each other. To determine the desired paths we proceed as follows.

- G-11. Let $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ be a segment of length at least 4. If there is a not yet determined edge $v_i v_{i+1} = f$ in $S_{u,v}$ then we guess which of the fastest temporal paths: from

v_{i-1} to v_{i+1} , from v_{i+1} to v_{i-1} , from v_i to v_{i+2} , from v_{i+2} to v_i pass through the edge f . If there are two incident edges $e = v_{i-1}v_i$ and $f = v_iv_{i+1}$ in $S_{u,v}$, that are determined with respect to $\lambda(v_1v_2)$ and $\lambda(v_{p-1}v_p)$, respectively then we guess which of the fastest temporal paths: from v_{i-1} to v_{i+1} , from v_{i+1} to v_{i-1} pass through the edges e, f .

We create $O(1)$ guesses for every such segment $S_{u,v}$, and $O(k^2)$ new guesses in total, as there are at most $O(k^2)$ segments.

Note that the condition for segment length of at least four comes from Lemma 5. We now conclude the following.

Corollary 4. *Let $S_{u,v}$ be an arbitrary segment in G' . If $S_{u,v}$ is of length 3 or 2 then it has at most 3 or 2 not yet determined edges, respectively. If $S_{u,v}$ is of length at least 4 then the labels of all its edges are determined with respect to the first edge.*

At this point G' is a graph, where each edge e has a value for its label $\lambda(e)$ that depends on (i.e., is a function of) some other label $\lambda(f)$ of edge f , or it depends on no other label. We now describe how we create variables and start building our ILP instances. For every edge e in G' that is incident to a vertex of interest, we create a variable x_e that can have values from $\{1, 2, \dots, \Delta\}$. Besides that, we create one variable for each edge that is still not yet determined on a segment. Since each vertex of interest is incident to at most k edges in G' , and each segment has at most one extra not yet determined edge, we create $O(k^2)$ variables. At the end, we create our final guess.

G-12. We guess the permutation of all $O(k^2)$ variables. So, for any two variables x_e and x_f , we know if $x_e < x_f$ or $x_e = x_f$, or $x_e > x_f$. This results in $O(k^2!) = k^{O(k^2)}$ guesses and consequently each of the ILP instances we created up to now is further split into $k^{O(k^2)}$ new ones.

We have now finished creating all ILP instances. From Section 3.2.2 we know the structure of all guessed paths, to which we have just added also the knowledge of permutation of all variables. We proceed with adding constraints to each of our ILP instances. First, we add all constraints for the labels of edges that we have determined up to now. We then continue to iterate through all pairs of vertices and start adding equality (resp. inequality) constraints for the fastest (resp. not necessarily fastest) temporal paths between them.

We now describe how we add constraints to a path. Whenever we say that a duration of a path gives an equality or inequality constraint, we mean the following. Let $P = (u = v_1, v_2, \dots, v_p = v)$ be the underlying path of a fastest temporal path from u to v , and let $Q = (u = z_1, z_2, \dots, z_r = v)$ be the underlying path of another temporal path from u to v . Then we know that $d(P, \lambda) = D_{u,v}$ and $d(Q, \lambda) \geq D_{u,v}$. Using Observation 1 we create an *equality constraint* for P of the form

$$D_{u,v} = \sum_{i=2}^{p-1} (\lambda(v_iv_{i+1}) - \lambda(v_{i-1}v_i))_{\Delta} + 1, \quad (6)$$

and an *inequality constraint* for Q

$$D_{u,v} \leq \sum_{i=2}^{r-1} (\lambda(z_iz_{i+1}) - \lambda(z_{i-1}z_i))_{\Delta} + 1. \quad (7)$$

In both cases we implicitly assume that if the difference of $(\lambda(z_iz_{i+1}) - \lambda(z_{i-1}z_i))$ is negative, for some i , we add the value Δ to it (i.e., we consider the difference modulo Δ), therefore we

have the sign Δ around the brackets. Note that we can determine if the difference between two consecutive labels is positive or negative. In the case when two consecutive labels are determined with respect to the same label $\lambda(e)$ the difference between them is easy to determine. If consecutive labels are not determined with respect to the same label, both labels are considered undetermined and are assigned a variable for which we know in what kind of relation they are (see guess G-12). Therefore, we know when Δ has to be added, which implies that Eqs. (6) and (7) are calculated correctly for all paths.

We iterate through all pairs of vertices x, y and make sure that a fastest temporal path from x to y produces the equality constrain Eq. (6), and all other temporal paths from x to y produce the inequality constraint Eq. (7). For each pair, we argue how we determine these paths.

Fastest paths between $u, v \in U$. Let $u, v \in U$, i.e., both u, v are vertices of interest. For the path from u to v (resp. from v to u) in G' , which we guessed that it coincides with the fastest in G-1, we introduce an equality constraint. We then iterate over all other paths from u to v (resp. from v to u) in G' , and for each one we introduce an inequality constraint. There are $k^{O(k)}$ possible paths from u to v in G' . Therefore we add $k^{O(k)}$ inequality constraints for the pair u, v .

Fastest paths from $u \in U$ to $x \in V(G') \setminus U$. From the guesses G-8 and G-10 we know the fastest temporal paths from u to all vertices in a segment $S_{w,v}$. In this case, we create an equality constraint for the fastest path and we iterate through all other paths, for which we introduce the inequality constraints. There are $k^{O(k)}$ possible paths of the form $u \rightsquigarrow w$ (resp. $u \rightsquigarrow v$), and a unique way how to extend these paths from w (resp. v) to reach x in $S_{w,v}$. Therefore we add $k^{O(k)}$ inequality constraints for the pair u, x .

Fastest paths from $x \in V(G') \setminus U$ to $u \in U$. Let x be a vertex in the segment $S_{w,z} = (w = z_1, z_2, \dots, z_r = z)$, and let $u \in U$. If $S_{w,z}$ is of length 3 or less, then we already know a fastest temporal path from every vertex in the segment to u (since $S_{w,z}$ has at most 2 inner vertices, we determined the fastest temporal paths from them to u in guess G-4).

Assume that $S_{w,z}$ is of length at least 4. From Corollary 4 we know that the labels of all the edges in $S_{w,z}$ are determined with respect to the label of the first edge. Moreover, this gives us the knowledge of the exact differences among two consecutive edge labels, which is enough to uniquely determine travel delays at all of the inner vertices $z_i \in S_{w,z}$ (see Definition 4).

From the matrix D we can easily determine the two vertices $z_i, z_{i+1} \in S_{w,z} \setminus \{w, z\}$ for which a fastest temporal path from z_i to u has the biggest duration. Let us denote with P^+ a fastest temporal path of the form $z_2 \rightarrow z \rightsquigarrow u$, and with P^- a fastest temporal path of the form $z_{r-1} \rightarrow w \rightsquigarrow u$ (we know these paths from guess G-8). It follows that all vertices z_j in $S_{w,z} \setminus \{z_i, z_{i+1}\}$ that are closer to w than z_i, z_{i+1} reach u the fastest using the path $(z_j \rightarrow z_{j-1} \rightarrow \dots \rightarrow z_2) \cup P^+$ and all the vertices z_j in $S_{w,z} \setminus \{z_i, z_{i+1}\}$ that are closer to z than z_i, z_{i+1} reach u the fastest using the path $(z_j \rightarrow z_{j+1} \rightarrow \dots \rightarrow z_{r-1}) \cup P^-$. Since the first part of the above path is unique, and we know that the second part is the fastest, it follows that these paths indeed represent the fastest temporal paths to u . What remains to determine is the fastest temporal paths from z_i, z_{i+1} to u . We distinguish the following two options.

- (i) $z_i \neq z_{i+1}$. Then a fastest temporal path from z_i to u is $(z_i \rightarrow z_{i-1} \rightarrow \dots \rightarrow z_2) \cup P^+$, and a fastest temporal path from z_{i+1} to u is $(z_{i+1} \rightarrow z_{i+2} \rightarrow \dots \rightarrow z_{r-1}) \cup P^-$.
- (ii) $z_i = z_{i+1}$, i.e., let z_i be the unique vertex, that is furthest away from u in $S_{w,z}$. In this case, we have to determine if a fastest temporal path from z_i to u , travels first through vertex z_{i-1} (and then through w), or it travels first through z_{i+1} (and then through z). Since

we know the values $D_{z_{i-1},u}$, $D_{z_{i+1},u}$, and we know the value of the waiting time $\tau_{v_{i-1}}^{v_i, v_{i-2}}$ at vertex v_{i-1} when traveling from v_i to v_{i-2} , we can uniquely determine the desired path. We set $c = D_{z_{i-1},u} + \tau_{v_{i-1}}^{v_i, v_{i-2}}$ and compare c to the value $D_{z_i,u}$. If $c < D_{z_i,u}$ we conclude that our ILP has no solution and we stop with calculations, if $c = D_{z_i,u}$ then a fastest temporal path from z_i to u is of the form $(z_i \rightarrow z_{i-1} \rightarrow \dots \rightarrow z_2) \cup P^+$, if $c > D_{z_i,u}$ then a fastest temporal path from z_i to u is of the form $(z_i \rightarrow z_{i+1} \rightarrow \dots \rightarrow z_{r-1}) \cup P^-$.

Once a fastest temporal path from c to u is determined, we introduce an equality constraint for it. For each of the other $k^{O(k)}$ paths from x to u (which correspond to all paths of the form $w \rightsquigarrow u$ and $z \rightsquigarrow u$, together with the unique subpath on $S_{w,z}$), we introduce an inequality constraint. Therefore we add $k^{O(k)}$ inequality constraints for the pair x, u .

Fastest paths between $x, y \in V(G') \setminus U$. Let $x, y \in V(G') \setminus U$. There are two options.

- (i) Vertices x, y are in the same segment $S_{u,v} = (u, v_1, v_2, \dots, v_p, v)$. If the length of $S_{u,v}$ is less than 4 then we know what is the fastest path between vertices, as $x, y \in U^*$. Suppose now that $S_{u,v}$ is of length at least 4 and assume that x is closer to u in $S_{u,v}$ than y . Then we have two options; either the path from x to y travels only through the edges of $S_{u,v}$, denote such path as $P_{x,y}$, or it is of the form $x \rightarrow v_1 \rightarrow u \rightsquigarrow v \rightarrow v_p \rightarrow y$, denote it as $P_{x,y}^*$. Note that we can determine $P_{x,y}^*$ as it is a concatenation of a unique path from x to v_2 , together with the fastest path from v_2 to v_p , that travels through u and v (we know this path from G-7), and the unique path from v_p to y . Because of Corollary 4 we can determine $c = d(P_{x,y}, \lambda)$. If $c > D_{x,y}$ we set the fastest path to be $P_{x,y}^*$, if $c = D_{x,y}$ then the fastest path is $P_{x,y}$, and if $c < D_{x,y}$ we conclude that our ILP has no solution and we stop with calculations.
- (ii) Vertices x and y are in different segments. Let x be a vertex in the segment $S_{u,v} = (u = v_1, v_2, \dots, v_p = v)$ and let y be a vertex in the segment $S_{w,z} = (w = z_1, z_2, z_3, \dots, z_r = z)$. By checking the durations of the fastest paths from x to every vertex in $S_{w,z} \setminus \{w, z\}$ we can determine the vertex $z_i \in S_{w,z}$, for which the duration from x is the biggest. Note that if there are two such vertices z_i and z_{i+1} , we know exactly how all fastest temporal paths enter $S_{w,z}$ (we use similar arguing as in case (i) from above, where we were determining the fastest path from $x \in V(G')$ to $u \in U$). This implies that the fastest temporal paths from x to all vertices z_2, z_3, \dots, z_{i-1} (resp. $z_{i+1}, z_{i+2}, \dots, z_{r-1}$) pass through w (resp. z). Now we determine the vertex $v_j \in S_{u,v} \setminus \{u, v\}$, for which the value of the durations of the fastest paths from it to the vertex y is the biggest. Again, if there are two such vertices v_j and v_{j+1} we know exactly how the fastest temporal paths, starting in these two vertices, leave the segment $S_{u,v}$. We use similar arguing as in case (i) from above, when we were determining the fastest path from $x \in V(G')$ to $u \in U$. Knowing the vertex v_j implies that the fastest temporal paths from the vertices v_2, v_3, \dots, v_{j-1} (resp. $v_{j+1}, v_{j+2}, \dots, v_{p-1}$) to the vertex y passes through u (resp. v). Since we know the following fastest temporal paths (see guess G-7) $z_2 \rightarrow w \rightsquigarrow u \rightarrow v_2$, $z_2 \rightarrow w \rightsquigarrow v \rightarrow v_{p-1}$, $z_{r-1} \rightarrow z \rightsquigarrow v \rightarrow v_{p-1}$ and $z_{r-1} \rightarrow z \rightsquigarrow v \rightarrow v_{p-1}$, we can uniquely determine all fastest temporal paths from $x \neq v_j$ to any $y \in S_{u,v} \setminus \{z_i\}$.

In case when $x = v_j$ and $y = z_i$ and the segments are of length at least 4, we can uniquely determine the fastest path from v_j to z_i , using similar arguing as in case (ii) from above, when we were determining the fastest path from $x \in V(G')$ to $u \in U$. If at least one of the segments is of length 3 or less, we can again uniquely determine the fastest path from v_j to z_i , using the same approach, and the knowledge of fastest paths to (or from) all vertices of the segment of length 3 (as we guessed them in guess G-7).

Once the fastest path is determined we introduce the equality constraint for it and iterate through all other paths, for which we introduce inequality constraints. To enumerate all these non-fastest temporal paths, we just consider all possible paths $u \rightsquigarrow w$, where u and w are the vertices of interest that are the endpoints of segments to which x and y belong; once the correct segment is reached, there is a unique path to the desired vertex x (resp. y). Therefore we introduce $k^{O(k)}$ inequality constraints for each pair of vertices x, y .

Fastest paths for vertices in Z . All of the above is enough to determine the labeling λ of G' . We have to extend the labeling to consider also the vertices of $Z = V(G) \setminus V(G')$ that we initially removed from G .

Recall that $G[Z]$ consists of disjoint trees and that each of these trees has a unique neighbor (clip vertex) v in G' . We then define the tree T_v in $G[Z \cup \{v\}]$ as the collection of trees from $G[Z]$ with a clip vertex v , together with the root v . Determining the fastest temporal paths between any two vertices in the same tree is a straightforward process (see Theorem 3), therefore we exclude this case from our upcoming analysis. From Observation 4 it follows that knowing temporal paths between any $y \in V(G')$ and all vertices in the first layer of the tree T_v (i.e., children of the root v), it is enough to determine the fastest temporal paths between y and all other vertices in T_v . Therefore, in the upcoming analysis, we focus only on the vertices in the first layer of each tree T_v . During the process of determining the fastest paths from and to the vertices in Z , we use the fact that we have already identified the fastest paths among all vertices in G' .

We split our analysis into two cases. First, when the clip vertex v of tree T_v is not a vertex of interest, and second when the clip vertex is also a vertex of interest in G' . In the first case, we use the fact that v has only two edges e, f incident to it in G' , and that we can determine all the labels of the tree edges with respect to $\lambda(e), \lambda(f)$ (see Lemma 7). This results to be enough for us to determine the fastest temporal paths among any vertex r in the first layer of the tree T_v and an arbitrary vertex in $V(G) \setminus V(T_v)$. In the second case, we cannot determine the labeling of the tree with respect to the labels of all edges incident to the clip vertex. Therefore, we split the vertices in the first layer of T_v into equivalence classes, and use the fact that the fastest temporal paths between two vertices in the same equivalence class coincide on the edges outside of T_v .

Fastest paths from $r \in Z$ to $y \in U \cup U^*$. Let $x \in V(G')$ be the clip vertex of the tree T_x in $G[Z \cup \{x\}]$ with $r \in V(T_x)$ be a vertex in the first layer of T_x . We distinguish the following two cases.

- (i) The clip vertex $x = u \in U$ is a vertex of interest. In this case, we can w.l.o.g. assume that r is a representative vertex in its equivalence class among the first layer vertices of T_u . From the guesses G-5 and G-6 we know a fastest temporal path from r to y .
- (ii) The clip vertex $x \in U$ is not a vertex of interest. Then $x = v_j$ is a part of some segment $S_{u,v} = \{u = v_1, v_2, \dots, v_p = v\}$, where $j \neq 1 \neq p$. Using Lemma 7 we can determine all the edge labels of T_x with respect to the label $\lambda(v_{j-1}v_j)$ and with respect to the label $\lambda(v_jv_{j+1})$. Using the calculations of fastest temporal paths among vertices in G' and the performed guesses we know the exact structure (i.e., the sequence of vertices and edges) of the following paths:
 - path P_{xy}^* which is a fastest temporal path from the vertex x to the vertex y ,
 - path P_{xy}^u which is a fastest temporal path from the vertex x to the vertex y , that passes through the vertex u ,

- path P_{xy}^v which is a fastest temporal path from the vertex x to the vertex y , that passes through the vertex v .

Note that P_{xy}^* is either equal to the path P_{xy}^u or to the path P_{xy}^v . More precisely, from the guesses performed we know the structure of the fastest path from v_2 through u , which then continues to any other vertex of interest, and any other neighbor of the vertex of interest (see guesses G-7 and G-8). This path can then be easily (uniquely) extended to start from $x = v_i$, as there is a unique (temporal) path starting at x and finishing in u or v .

Suppose now that $P_{xy}^* = P_{xy}^u$. Since the labels of T_x are determined with respect to $\lambda(v_{i-1}x)$ we can calculate the value $c = D_{x,y} + |\lambda(v_{i-1}x) - \lambda(rx)|$. We then compare c to $D_{r,y}$ and get one of the following three options. First $c = D_{r,y}$, in this case, a fastest temporal path from r to y uses first the edge rx and then continues to y using the edges and vertices of P_{xy}^* . Second $c > D_{r,y}$, in this case, a fastest temporal path from r to y uses first the edge rx and then continues to y using the vertices and edges of P_{xy}^v . Third $c < D_{r,y}$, in this case, we stop the calculation and return false, as it cannot happen that a temporal path has a smaller duration than the corresponding value in the matrix D .

In both cases, we introduce an equality constraint for the determined fastest temporal path and inequality constraints for all the other $k^{O(k)}$ paths.

Fastest paths from $r \in Z$ to $y \in V(G) \setminus (U \cup U^*)$. The proof in this case is similar to the one above. We still split the analysis into two parts, one where the clip vertex x of a tree T_x that includes r is in U and one where it is not in U . The difference is that in some cases we need to also extend the ending part of the path (which can be done uniquely, using the same arguments as in the above analysis).

Once we determine a fastest temporal path from r to y we introduce an equality constraint for it, and for all other $k^{O(k)}$ paths we introduce inequality constraints.

The procedure produces one equality constraint (for the fastest path) and $k^{O(k)}$ inequality constraints.

Fastest paths from $y \in V(G)$ to $r \in Z$. The process of determining fastest temporal paths from any vertex in the graph G to a vertex r that is a vertex in the first layer of a tree $T_x \in G[Z \cup \{x\}]$, where $x \in V(G')$, is similar to the one above, but performed in the opposite direction.

3.2.5. Solving ILP instances

All of the above finishes our construction of ILP instances. We have created $f(k)$ instances (where f is a double exponential function), each with $O(k^2)$ variables and $O(n^2)g(k)$ constraints (again, g is a double exponential function). We now solve each ILP instance I , using results from Lenstra [51], in the FPT time, with respect to k . If none of the ILP instances gives a positive solution, then there exists no labeling λ of G that would realize the matrix D (i.e., for any pair of vertices $u, v \in V(G)$ the duration of a fastest temporal path from u to v has to be $D_{u,v}$). If there is at least one I that has a valid solution, we use this solution and produce our labeling λ , for which (G, λ) realizes the matrix D . We have proven in the previous subsections that this is true since each ILP instance corresponds to a specific configuration of fastest temporal paths in the graph (i.e., considering all ILP instances is equivalent to exhaustively searching through all possible temporal paths between vertices). Besides that, in each ILP instance we add also the constraints for durations of all temporal paths between each pair of vertices. This results in setting the duration of a fastest path from a vertex $u \in V(G)$ to a vertex $v \in V(G)$

as $D_{u,v}$, and the duration of all other temporal paths from u to v , to be greater or equal to $D_{u,v}$, for all pairs of vertices u, v . Therefore, if there is an instance with a positive solution, then this instance gives rise to the desired labeling, as it satisfies all of the constraints. For the other direction, we can observe that if there is a labeling λ meeting all duration requirements specified by D , then this labeling produces a specific configuration of fastest temporal paths. Since we consider all configurations, one of the produced ILP instances will correspond to the configuration implicitly defined by λ , and hence our algorithm finds a solution.

To create the labeling λ from a solution X , of a positive ILP instance, we use the following procedure. First we label each edge e , that corresponds to the variable x_e by assigning the value $\lambda(e) = x_e$. We then continue to set the labels of all other edges. We know that the labels of all of the remaining edges depend on the label of (at least one) of the edges that were determined in previous step. Therefore, we easily calculate the desired labels for all remaining edges.

4. Conclusion

We have introduced a natural and canonical temporal version of the graph realization problem with respect to distance requirements, called SIMPLE PERIODIC TEMPORAL GRAPH REALIZATION. We have shown that the problem is NP-hard in general and polynomial-time solvable if the underlying graph is a tree. Building upon those results, we have investigated its parameterized computational complexity with respect to structural parameters of the underlying graph that measure “tree-likeness”. For those parameters, we essentially gave a tight classification between parameters that allow for tractability (in the FPT sense) and parameters that presumably do not. We showed that our problem is W[1]-hard when parameterized by the feedback vertex number of the underlying graph, and that it is in FPT when parameterized by the feedback edge number of the underlying graph. Note that most other common parameters that measure tree-likeness (such as the treewidth) are smaller than the vertex cover number.

We believe that our work spawns several interesting future research directions and builds a base upon which further temporal graph realization problems can be investigated.

Further parameterizations. There are several structural parameters that can be considered to obtain tractability which are either larger than or incomparable to the feedback vertex number.

- The *vertex cover number* measures the distance to an independent set, on which we trivially only have no-instances of our problem. We believe this is a promising parameter to obtain tractability.
- The *tree-depth* measures “star-likeness” of a graph and is incomparable to both the feedback vertex number and the feedback edge number. We leave the parameterized complexity of our problem with respect to this parameter open.
- Parameters that measure “path-likeness” such as the *pathwidth* or the *vertex deletion distance to disjoint paths* are also natural candidates to investigate.

Furthermore, we can consider combining a structural parameter with Δ . Our NP-hardness reduction (Theorem 1) produces instances with constant Δ , so as a single parameter Δ cannot yield fixed-parameter tractability. However, in our parameterized hardness reduction (Theorem 2) the value for Δ in the produced instance is large. This implies that our result does not rule out e.g. fixed-parameter tractability for the combination of the treewidth and Δ as a parameter. We believe that investigating such parameter combinations is a promising future research direction.

Further problem variants. There are many natural variants of our problem that are well-motivated and warrant consideration. In the following, we give three specific examples. We believe that one of the most natural generalizations of our problem is to allow more than one label per edge in every Δ -period. This problem variant has been studied for the first time very recently by Erlebach et al. [26]. A well-motivated variant (especially from the network design perspective) of our problem is to consider the entries of the duration matrix D as upper bounds on the duration of fastest paths rather than exact durations. This problem variant has very recently been studied by Mertzios et al. [59]. Furthermore, it would be interesting to also study non-strict temporal paths, as opposed to the strict temporal paths that are considered in the present paper. These three problem variants are the most immediate extensions/variations of the problem studied in this paper. More generally, temporal graph realization problems (not necessarily periodic) can be naturally defined also with respect to other temporal graph objects (instead of fastest temporal paths and their durations) such as having sufficiently small separators [69], or sufficiently large temporal matching [57] etc.

References

- [1] 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*, 61:907–944, 2017. 4
- [2] E. C. Akrida, G. B. Mertzios, P. G. Spirakis, and C. Raptopoulos. The temporal explorer who returns to the base. *Journal of Computer and System Sciences*, 120:179–193, 2021. 4
- [3] E. Arrighi, N. Grüttemeier, N. Morawietz, F. Sommer, and P. Wolf. Multi-parameter analysis of finding minors and subgraphs in edge-periodic temporal graphs. In *Proceedings of the 48th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 283–297, 2023. 2, 3
- [4] J. Augustine, K. Choudhary, A. Cohen, D. Peleg, S. Sivasubramaniam, and S. Sourav. Distributed graph realizations. *IEEE Transactions on Parallel and Distributed Systems*, 33(6):1321–1337, 2022. 2
- [5] A. Bar-Noy, K. Choudhary, D. Peleg, and D. Rawitz. Efficiently realizing interval sequences. *SIAM Journal on Discrete Mathematics*, 34(4):2318–2337, 2020. 2
- [6] A. Bar-Noy, K. Choudhary, D. Peleg, and D. Rawitz. Graph realizations: Maximum degree in vertex neighborhoods. In *Proceedings of the 17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 10:1–10:17, 2020. 2
- [7] A. Bar-Noy, D. Peleg, M. Perry, and D. Rawitz. Composed degree-distance realizations of graphs. In *Proceedings of the 32nd International Workshop on Combinatorial Algorithms (IWOCA)*, pages 63–77, 2021. 2
- [8] A. Bar-Noy, D. Peleg, M. Perry, and D. Rawitz. Graph realization of distance sets. In *Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 13:1–13:14, 2022. 2
- [9] M. Behzad and J. E. Simpson. Eccentric sequences and eccentric sets in graphs. *Discrete Mathematics*, 16(3):187–193, 1976. 2
- [10] R. E. Bixby and D. K. Wagner. An almost linear-time algorithm for graph realization. *Mathematics of Operations Research*, 13(1):99–123, 1988. 2

- [11] B.-M. Bui-Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003. 5
- [12] A. Casteigts, T. Corsini, and W. Sarkar. Invited paper: Simple, strict, proper, happy: A study of reachability in temporal graphs. In *Proceedings of the 24th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 3–18, 2022. 4
- [13] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012. 3
- [14] A. Casteigts, A.-S. Himmel, H. Molter, and P. Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021. 4, 31
- [15] W.-K. Chen. On the realization of a (p, s) -digraph with prescribed degrees. *Journal of the Franklin Institute*, 281(5):406–422, 1966. 2
- [16] F. Chung, M. Garrett, R. Graham, and D. Shallcross. Distance realization problems with applications to internet tomography. *Journal of Computer and System Sciences*, 63(3):432–448, 2001. 2
- [17] J. C. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Information Processing Letters*, 30(4):215–220, 1989. 2
- [18] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshantov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. 5
- [19] A. Deligkas and I. Potapov. Optimizing reachability sets in temporal graphs by delaying. *Information and Computation*, 285:104890, 2022. 4
- [20] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, London, 2013. 5
- [21] 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. 4
- [22] 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. 4
- [23] 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)*, pages 30:1–30:19, 2023. 4
- [24] P. Erdős and T. Gallai. Graphs with prescribed degrees of vertices. *Mat. Lapok*, 11:264–274, 1960. 2
- [25] T. Erlebach, M. Hoffmann, and F. Kammer. On temporal graph exploration. *Journal of Computer and System Sciences*, 119:1–18, 2021. 4
- [26] T. Erlebach, N. Morawietz, and P. Wolf. Parameterized algorithms for multi-label periodic temporal graph realization. In *Proceedings of the 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND)*, pages 14:1–14:16, 2024. 4, 52

- [27] T. Erlebach and J. T. Spooner. A game of cops and robbers on graphs with periodic edge-connectivity. In *Proceedings of the 46th International Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, pages 64–75, 2020. 3
- [28] 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. 4, 10
- [29] M. R. Fellows, B. M. P. Jansen, and F. A. Rosamond. Towards fully multivariate algorithms: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013. 3
- [30] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer, 2006. 5
- [31] T. Fluschnik, H. Molter, R. Niedermeier, M. Renken, and P. Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020. 4
- [32] A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal on Discrete Mathematics*, 5(1):25–53, 1992. 2
- [33] A. Frank. Connectivity augmentation problems in network design. *Mathematical Programming: State of the Art 1994*, 1994. 2
- [34] H. Frank and W. Chou. Connectivity considerations in the design of survivable networks. *IEEE Transactions on Circuit Theory*, 17(4):486–490, 1970. 2
- [35] E. Füchsle, H. Molter, R. Niedermeier, and M. Renken. Delay-robust routes in temporal graphs. In *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 30:1–30:15, 2022. 4
- [36] E. Füchsle, H. Molter, R. Niedermeier, and M. Renken. Temporal connectivity: Coping with foreseen and unforeseen delays. In *Proceedings of the 1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND)*, pages 17:1–17:17, 2022. 4
- [37] D. Fulkerson. Zero-one matrices with zero trace. *Pacific Journal of Mathematics*, 10(3):831–836, 1960. 2
- [38] P. A. Golovach and G. B. Mertzios. Graph editing to a given degree sequence. *Theoretical Computer Science*, 665:1–12, 2017. 2
- [39] M. C. Golumbic and A. N. Trenk. *Tolerance Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2004. 2
- [40] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961. 2
- [41] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. *Handbooks in Operations Research and Management Science*, 7:617–672, 1995. 2
- [42] J. Guo, F. Hüffner, and R. Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC)*, pages 162–173, 2004. 3

- [43] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962. 2
- [44] S. L. Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quarterly of applied mathematics*, 22(4):305–317, 1965. 2
- [45] P. Hell and D. Kirkpatrick. Linear-time certifying algorithms for near-graphical sequences. *Discrete Mathematics*, 309(18):5703–5713, 2009. 2
- [46] 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. 2
- [47] N. Klobas, G. B. Mertzios, H. Molter, R. Niedermeier, and P. Zschoche. Interference-free walks in time: Temporally disjoint paths. *Autonomous Agents and Multi-Agent Systems*, 37(1):1, 2023. 4
- [48] 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)*, pages 62:1–62:15, 2022. 4
- [49] N. Klobas, G. B. Mertzios, H. Molter, and P. G. Spirakis. Temporal graph realization from fastest paths. In *Proceedings of the 3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND)*, pages 16:1–16:18, 2024. Best student paper award. 1
- [50] F. Kuhn and R. Oshman. Dynamic networks: Models and algorithms. *SIGACT News*, 42(1):82–96, March 2011. 5
- [51] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983. 4, 32, 36, 50
- [52] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014. 3
- [53] L. Lesniak. Eccentric sequences in graphs. *Periodica Mathematica Hungarica*, 6:287–293, 1975. 2
- [54] R. M. McConnell and J. P. Spinrad. Construction of probe interval models. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 866–875, 2002. 2
- [55] F. McMorris, C. Wang, and P. Zhang. On probe interval graphs. *Discrete Applied Mathematics*, 88(1):315–324, 1998. Computational Molecular Biology DAM - CMB Series. 2
- [56] G. B. Mertzios, O. Michail, and P. G. Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019. 4
- [57] 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. 52
- [58] 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)*, pages 75:1–75:18, 2021. 4

- [59] G. B. Mertzios, H. Molter, and P. G. Spirakis. Realizing temporal transportation trees. *CoRR*, abs/2403.18513, 2024. 52
- [60] 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. 4
- [61] N. Morawietz, C. Rehs, and M. Weller. A timecop’s work is harder than you think. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 71–1, 2020. 3
- [62] N. Morawietz and P. Wolf. A timecop’s chase around the table. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2021. 3
- [63] A. Patrinos and S. L. Hakimi. The distance matrix of a graph and its tree realization. *Quarterly of Applied Mathematics*, 30:255–269, 1972. 2
- [64] E. Rubei. Weighted graphs with distances in given ranges. *Journal of Classification*, 33:282—297, 2016. 2
- [65] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann. Tracking human mobility using wifi signals. *PloS one*, 10(7):e0130824, 2015. 3
- [66] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978. 6
- [67] H. Tamura, M. Sengoku, S. Shinoda, and T. Abe. Realization of a network from the upper and lower bounds of the distances (or capacities) between vertices. In *Proceedings of the 1993 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2545—2548, 1993. 2
- [68] H. Wu, J. Cheng, Y. Ke, S. Huang, Y. Huang, and H. Wu. Efficient algorithms for temporal path computation. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2927–2942, 2016. 5
- [69] P. Zschoche, T. Fluschnik, H. Molter, and R. Niedermeier. The complexity of finding separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020. 4, 52