Parameterized Algorithms for Multi-Label Periodic Temporal Graph Realization

Thomas Erlebach
Department of Computer Science, Durham University, Durham, UK
Nils Morawietz
Friedrich Schiller University Jena, Institute of Computer Science, Jena, Germany
Petra Wolf
LaBRI, CNRS, Université de Bordeaux, Bordeaux INP, France

Abstract

In the periodic temporal graph realization problem introduced by Klobas et al. [SAND '24] one is given a period Δ and an $n \times n$ matrix D of desired fastest travel times, and the task is to decide if there is a simple periodic temporal graph with period Δ such that the fastest travel time between any pair of vertices matches the one specified by D. We generalize the problem from simple temporal graphs to temporal graphs where each edge can appear up to ℓ times in each period, for some given integer ℓ. For the resulting problem Multi-Label Periodic TGR, we show that it is fixed-parameter tractable for parameter n and for parameter $\mathrm{vc} + \Delta$, where vc is the vertex cover number of the underlying graph. We also show the existence of a polynomial kernel for parameter $\mathrm{nu} + d_{\max}$, where nu is the number of non-universal vertices of the underlying graph and $d_{\max}$ is the largest entry of D. Furthermore, we show that the problem is *NP*-hard for each $\ell \geq 5$, even if the underlying graph is a tree, a case that was known to be solvable in polynomial time if the task is to construct a simple periodic temporal graph, that is, if $\ell = 1$.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Fixed-Parameter Tractability, Almost-Clique, Kernelization, Dynamic Network, Temporal Graph

Digital Object Identifier 10.4230/LIPIcs.SAND.2024.12

Funding *Thomas Erlebach*: Supported by EPSRC grants EP/S033483/2 and EP/T01461X/1.
Petra Wolf: Supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).

1 Introduction

Graph realization problems are problems where one is given information about a certain property of a graph, such as the matrix of shortest-path distances or the degree sequence, and wants to decide whether there exists a graph for which that property matches the given information (and to find such a graph, if it exists). A wide range of graph realization problems have been studied for static graphs for many years, with the work by Erdős and Gallai [9] on realizing a given degree sequence and the work by Hakimi and Yau [10] on realizing a given distance matrix by an edge-weighted graph being two particularly early examples. In recent years, temporal graphs, i.e., graphs whose edge set may change in each time step, have received substantial attention. In temporal graphs, one usually considers paths that traverse at most one edge in each time step (and we also do so in this paper), although non-strict paths where several edges can be traversed in the same time step have also been studied. Many classical graph problem have been adapted and studied in the temporal graph setting (see [14] for an introduction to temporal graphs and [4] for a broader overview of different classes of time-varying graphs). Therefore, considering graph realization problems in the temporal graph setting is a natural and timely direction. Very recently, Klobas et al. [12] have started this line of research and introduced the following periodic temporal

3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2024).
Editors: Arnaud Casteigts and Fabian Kuhn; Article No. 12; pp. 12:1–12:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

graph realization problem (PERIODIC TGR): Given a period Δ and an $n \times n$ integer matrix D that specifies the desired fastest travel times for each pair of vertices, find a simple periodic temporal graph $\mathcal{G}$ with period Δ such that the fastest travel times in $\mathcal{G}$ match those given by D (or decide that no such temporal graph exists). Here, a periodic temporal graph is *simple* if each edge of the underlying graph appears exactly once in each period.

Klobas et al. [12] noted that the consideration of periodic temporal graphs can be motivated by, for example, railway networks, satellite networks, or social networks. They showed that PERIODIC TGR is *NP*-hard for any $\Delta \geq 3$ and also W[1]-hard when parameterized by the feedback vertex number of the underlying graph. (The latter result also applies to the non-periodic version of the problem if the distance matrix can have entries equal to ∞.) Here, the underlying graph is the graph containing edges between all vertex pairs that have distance 1 according to D. They showed that the problem can be solved in polynomial time if the underlying graph is a tree or a cycle, and that it is fixed-parameter tractable (FPT) when parameterized by the feedback edge number of the underlying graph. Finally, they raised a number of interesting questions for future research, including the investigation of PERIODIC TGR with the vertex cover number of the underlying graph as parameter and with parameter combinations that include a structural parameter and the period Δ.

Our contribution. In this paper, we follow up on the work by Klobas et al. [12]. Furthermore, we generalize PERIODIC TGR from simple periodic temporal graphs to ℓ-label periodic temporal graphs, i.e., to periodic temporal graphs where each edge of the underlying graph is allowed to appear up to ℓ times in each period. This problem (MULTI-LABEL PERIODIC TGR) is defined as follows: Given a period Δ and an $n \times n$ integer matrix D that specifies the desired fastest travel times for each pair of vertices and a positive integer ℓ, find an ℓ-label periodic temporal graph $\mathcal{G}$ with period Δ such that the fastest travel times in $\mathcal{G}$ match those given by D (or decide that no such temporal graph exists). Clearly, PERIODIC TGR is the special case of MULTI-LABEL PERIODIC TGR where $\ell = 1$. We also consider the non-periodic version (MULTI-LABEL TGR). While Klobas et al. mainly considered parameters that relate to how close the underlying graph is to being a tree, we explore also the opposite end of the spectrum and consider a parameter that measures how close the underlying graph is to being a clique, namely the number of non-universal vertices. As the problem is trivial if the underlying graph is a clique, parameters that measure closeness to a clique are interesting candidates for obtaining FPT algorithms.

We obtain the following main results:

- The known *NP*-hardness proof for PERIODIC TGR [12] only applies to MULTI-LABEL PERIODIC TGR with $\ell = 1$ and leaves open the complexity for $\ell \geq 2$. We show that MULTI-LABEL PERIODIC TGR is *NP*-hard for every $\ell \geq 1$ even if the largest entry in D is 3. For $\ell \geq 3$, we show *NP*-hardness even if the underlying graph has a size-1 feedback vertex set.
- In contrast to the known result that PERIODIC TGR can be solved in polynomial time for trees [12], we show that MULTI-LABEL PERIODIC TGR is *NP*-hard for any $\ell \geq 5$ even if the underlying graph is a star.
- Both MULTI-LABEL PERIODIC TGR and MULTI-LABEL TGR are FPT for parameter n. Here, n is the number of vertices. (For PERIODIC TGR, this result is implied by the FPT algorithm for parameter feedback edge number by Klobas et al. [12], but our algorithm is conceptually simpler and can handle the multi-label problem variants.)
- MULTI-LABEL PERIODIC TGR is FPT for parameter $\mathrm{vc} + \Delta$, where vc is the vertex cover number of the underlying graph.

- Multi-Label TGR can be solved in $O(pn^4)$ time if D has no entries equal to ∞, $\ell \geq n^2$, and the underlying graph is such that for each pair (u, v) of vertices the number of u-v paths is at most p. For trees and cycles, we have $p \leq 2$, and hence the algorithm runs in polynomial time.
- Multi-Label Periodic TGR admits a polynomial kernel for parameter $\mathrm{nu} + d_{\max}$ and is hence FPT for that parameter, where nu is the number of non-universal vertices of the underlying graph and $d_{\max}$ is the largest entry of D.

The remainder of the paper is structured as follows. After discussing further related work below, we give formal definitions and present preliminary results in Section 2. Our hardness results are presented in Section 3, and our algorithmic results in Section 4. Section 5 gives conclusions and open problems.

Proofs of statements marked with ($\star$) are deferred to the full version.

Related work. For a general introduction to temporal graphs, we refer to the article by Michail and Spirakis [14]. The only previous work dealing with the problems we consider (but only for the case of simple temporal graphs) is the recent work by Klobas et al. [12], which has already been discussed above. Other settings where the task is to assign time labels to the edges of a graph in order to create a temporal graph have also been studied, but mainly with the goal of ensuring certain temporal connectivity properties rather than realizing pre-specified journey durations. For example, Akrida et al. [1] studied the problem of assigning (multiple) labels to the edges of a given graph in such a way that the resulting temporal graph is temporally connected (i.e., there exist u-v journeys for all pairs (u, v) of vertices), with the objective of minimizing the total number of labels used. They showed that $O(n)$ labels suffice. Klobas et al. [11] showed that the problem can be solved optimally in polynomial time but becomes *NP*-hard if restrictions are placed on the lifetime of the temporal graph or if connectivity needs to be established only for a subset of the vertices. Mertzios et al. [13] studied variations of the problem where the goal is to minimize the maximum number of labels assigned to any single edge, termed the *temporality* of the temporal graph. Note that the parameter ℓ that we consider in this paper corresponds to the temporality. Enright et al. [8] considered the problem of reordering the snapshots of a given temporal graph in order to minimize reachability.

2 Preliminaries

For details about parameterized complexity we refer to the standard monographs [5, 7].

For any integers i, j with $i \leq j$ we write $[i, j]$ for the set $\{i, i+1, i+2, \ldots, j\}$. We use standard notation for (static) graphs (see, e.g., [6]). For a graph $G = (V, E)$ and a vertex v of G, we denote by $N_G(v)$ the *neighbors* of v in G and define $N_G[v] := N_G(v) \cup \{v\}$. If the graph is clear from the context, we may omit the subscript. We write uv to denote an edge $\{u, v\}$ in an undirected graph.

A *temporal graph* is a graph that evolves over discrete time steps and whose vertex set remains the same while the edge set may be different in each time step. Two standard ways to represent a temporal graph $\mathcal{G}$ with lifetime L are as follows: The first representation uses a pair $(G = (V, E), \lambda)$, where $G = (V, E)$ is an undirected graph and $\lambda : E \to (2^{[1,L]} \setminus \{\emptyset\})$ is a function that assigns to each $e \in E$ the non-empty set of time steps during which e is present. The graph G is called the *underlying graph* of $\mathcal{G}$. We also call λ a *multi-labeling* to emphasize that an edge can receive more than one label. The second representation uses a sequence $(G_1, G_2, \ldots, G_L)$ of *snapshots* or *layers*, where $G_i = (V, E_i)$, for $1 \leq i \leq L$, is the graph on vertex set V that contains all edges that are present in time step i. The underlying graph is then the graph $G = (V, E)$ with $E = \bigcup_{i\in[1,L]} E_i$. The two representations are mathematically equivalent, and each can be transformed into the other in a straightforward way. For implementation purposes, we assume in this paper that temporal graphs are represented in the form (G, λ), but for ease of exposition we will also often use terminology that refers to the representation using explicit snapshots. We use the convention that $n = |V|$ and $m = |E|$ throughout.

If a temporal graph $\mathcal{G}$ with lifetime L is considered as a non-periodic graph, one assumes that the graph ceases to exist once time step L has passed. If $\mathcal{G}$ is considered as a periodic graph (with period L), then it is assumed that the snapshots repeat after L time steps, i.e., $G_{i+zL} = G_i$ for all $i \in [1, L]$ and all positive integers z. A temporal graph with lifetime or period L is called *simple* if every edge of the underlying graph appears in only one snapshot among the first L time steps, i.e., if $|\lambda(e)| = 1$ for all $e \in E$. For simple temporal graphs, we also write $\lambda(e) = t$ instead of $\lambda(e) = \{t\}$ if $t \in [1, L]$ is the time step in which edge e appears. For periodic temporal graphs, we usually denote the period by Δ instead of L.

If an edge e is present in time step t of a temporal graph $\mathcal{G}$, we say that (e, t) is a *time-edge* of $\mathcal{G}$. A *u-v journey* (or *u-v temporal path*) in $\mathcal{G}$ is a sequence $((e_1, t_1), (e_2, t_2), \ldots, (e_r, t_r))$ of time-edges such that $t_i < t_{i+1}$ for $1 \le i < r$ and $(e_1, e_2, \ldots, e_r)$ is a u-v path in the underlying graph of $\mathcal{G}$. The journey *starts* or *begins* at u in time step t_1, *reaches* or *arrives at* v in time step t_r, and has *duration* or *travel time* $t_r - t_1 + 1$. For vertices u, v and any time step t, an *earliest-arrival u-v journey at time t* is a u-v journey that begins at u at some time $\ge t$ and minimizes the time when it arrives at v. A *fastest u-v journey* is a u-v journey of minimum duration, and the duration of that journey is referred to as *the fastest travel time from u to v*.

A *distance matrix* D is an $n \times n$ matrix whose values are non-negative integers or ∞. If all values are non-negative integers, we say that D is *finite-valued*. The rows (and columns) of D correspond to n vertices, and we use V to denote the set of these n vertices. For two vertices $u, v \in V$, we use D_{uv} to denote the entry in row u and column v of D, and that entry specifies the desired fastest travel time from u to v. We say that a temporal graph $\mathcal{G}$ with vertex set V *realizes* D if, for any two vertices $u, v \in V$, the duration of a fastest u-v journey in $\mathcal{G}$ is equal to D_{uv}. (If $D_{uv} = \infty$, this means that $\mathcal{G}$ does not contain any u-v journey.) We can assume that $D_{uv} = 0$ if and only if $u = v$, as otherwise there cannot exist a temporal graph that realizes D. Furthermore, we can also assume for any pair (u, v) with $u \ne v$ that $D_{uv} = 1$ if and only if $D_{vu} = 1$, as a journey with duration one uses a single time-edge and thus is also a journey in the opposite direction. We only consider distance matrices that satisfy these assumptions throughout this paper. The graph $G = (V, E)$ that contains precisely those edges uv for which $D_{uv} = D_{vu} = 1$ is called the *underlying graph induced by D* as any temporal graph that realizes D must have underlying graph G.

As mentioned in the introduction, Klobas et al. [12] introduced the problem of constructing, for a given $n \times n$ distance matrix D and period Δ, a periodic simple temporal graph with n vertices and period Δ that realizes D. We generalize this problem by allowing multiple labels per edge, with an input parameter ℓ specifying how many labels an edge can receive at most:

Multi-Label Periodic TGR
Input: An integer ℓ, an $n \times n$ distance matrix D, and a period Δ.
Question: Is there a periodic temporal graph $\mathcal{G}$ with period Δ that realizes D and in which no edge receives more than ℓ labels?

For this problem we assume that D is finite-valued, as otherwise the problem could be split into independent subproblems on temporally connected components. Note that, contrary to the case of non-periodic temporal graphs, the temporal reachability relation in periodic temporal graphs is symmetric and transitive.

Furthermore, we also consider the non-periodic variant of the problem:

Multi-Label TGR
Input: An integer ℓ and an $n \times n$ distance matrix D.
Question: Is there a non-periodic temporal graph $\mathcal{G}$ (with arbitrary lifetime) that realizes D and in which no edge receives more than ℓ labels?

For the non-periodic variant, we may allow the distance matrix to contain entries equal to ∞. We use $d_{\max} = \max\{D_{uv} \mid u, v \in V, D_{uv} \neq \infty\}$ to refer to the largest finite entry of D.

Basic observations. In the following, we present some basic observations about the problems under consideration. First, we observe that each yes-instance of Multi-Label Periodic TGR can be realized with at most n^2 labels per edge.

▶ **Lemma 1** ($\star$). *Let I be an instance of Multi-Label Periodic TGR or Multi-Label TGR with $\ell \geq n^2$. Then reducing ℓ to n^2 yields an equivalent instance.*

The argument to show Lemma 1 is that a solution only needs to realize one fastest u-v journey for each of the $n(n-1) < n^2$ vertex pairs (u, v), and for each such u-v journey it suffices to assign at most one additional label to every edge. Thus, it can never be necessary to assign more than n^2 labels to an edge.

Hence, in the following we assume that for each instance of Multi-Label Periodic TGR under consideration, $\ell \leq n^2$. Moreover, we can further assume that $\ell \leq \Delta$, since no edge can receive more than Δ labels.

Next, we observe that a yes-instance can be realized by using time labels of value at most $\ell \cdot d_{\max} \cdot n^2$.

▶ **Lemma 2** ($\star$). *Let $I := (\ell, D)$ ($I := (\ell, D, \Delta)$) be a yes-instance of Multi-Label TGR (Multi-Label Periodic TGR). There is a solution for I with largest time label at most $\ell \cdot d_{\max} \cdot m \leq \ell \cdot d_{\max} \cdot n^2$.*

The proof of Lemma 2 considers gaps (sequences of edgeless snapshots) between non-empty snapshots. For Multi-Label TGR it is clear that gaps of length greater than $d_{\max} - 1$ are never necessary and can be reduced by removing empty snapshots in the gap. As there are at most $m\ell$ snapshots with at least one edge, the result follows. For Multi-Label Periodic TGR, if there is a gap that is longer than $d_{\max} - 1$, we can perform a cyclic shift of the time labels so that the longest gap appears in the final steps of the period. All gaps before that final gap can then be reduced to size at most $d_{\max} - 1$ in the same way as in the non-periodic case, showing the lemma.

Note that for Multi-Label Periodic TGR, we can thus reduce Δ to at most $\ell \cdot n^2 \cdot d_{\max} \geq \ell \cdot m \cdot d_{\max} + d_{\max}$ if the period Δ is larger than $\ell \cdot m \cdot d_{\max} + d_{\max}$.

▶ **Corollary 3.** *For an instance (ℓ, D, Δ) of Multi-Label Periodic TGR with $\Delta > \ell \cdot n^2 \cdot d_{\max}$, the instance $(\ell, D, \ell \cdot n^2 \cdot d_{\max})$ is an equivalent instance of Multi-Label Periodic TGR.*

Note that this also implies the existence of polynomial kernels for Multi-Label Periodic TGR of size $\mathcal{O}(\ell \cdot n^2 \cdot d_{\max}) \subseteq \mathcal{O}(n^4 \cdot d_{\max})$, since ℓ can be reduced to n^2 and Δ can be reduced to $\ell \cdot n^2 \cdot d_{\max}$.

3 NP-Hardness for Multi-Label Periodic TGR on restricted instances

In this section, we present three hardness results for Multi-Label Periodic TGR on very restricted instances. Recall that Multi-Label Periodic TGR for $\ell = 1$ is known to be *NP*-hard for each $\Delta \geq 3$ and that for $\ell = 1$, Multi-Label Periodic TGR and Multi-Label TGR are known to be *NP*-hard and W[1]-hard when parameterized by the feedback vertex set number [12]. All three of our hardness results are obtained by reductions from a restricted version of Vertex Cover. First, we show that for each $\ell \geq 1$, Multi-Label Periodic TGR is *NP*-hard even if $d_{\max} = 3$. Afterwards, we show that for each $\ell \geq 5$, Multi-Label Periodic TGR is *NP*-hard even on stars, which stands in stark contrast to the fact that for $\ell = 1$ the problem can be solved in polynomial time on trees [12]. Finally, we show hardness for $\ell \in \{3, 4\}$ on graphs that are very close to trees, that is, on graphs with a feedback vertex set of size 1.

We start by showing that for each $\ell \geq 1$, Multi-Label Periodic TGR is *NP*-hard even if $d_{\max} = 3$ and the underlying graph is a dense split graph, i.e., a graph where the non-universal vertices form an independent set.

▶ **Theorem 4.** *For each $\ell \geq 1$, Multi-Label Periodic TGR is NP-hard even if the underlying graph is a dense split graph, $\Delta = 6$, and $d_{\max} = 3$.*

Proof. We reduce from Vertex Cover which is known to be *NP*-hard even if the input graph has maximum degree 3, contains no cycle of length three or four, and no two vertices of degree 3 are adjacent [15].

Vertex Cover
Input: A graph $G = (V, E)$ and an integer k.
Question: Is there a *vertex cover* of size at most k for G, that is, a set of vertices S of size at most k, such that each edge of E is incident with at least one vertex of S?

Let $\ell \geq 1$. (Our reduction does not actually depend on ℓ; it thus shows *NP*-hardness for all values of ℓ simultaneously.)

Let $I := (G = (V, E), k)$ be an instance of Vertex Cover with the above restrictions and let $n := |V|$ with $n \geq 13$. Clearly, we can assume $k < n$ as I is trivially a yes-instance otherwise. Without loss of generality, we assume that G contains four isolated edges x_1x_2, y_1y_2, z_1z_2, and w_1w_2. (We can ensure this property for any graph by adding four isolated edges and increasing k by four.) These four edges will be helpful to prove that D can be realized with only one label per edge, if I is a yes-instance of Vertex Cover.

We construct an instance $I' := (\ell, D, \Delta)$ of Multi-Label Periodic TGR as follows: We set $\Delta = 6$. The underlying graph $G' := (V', E')$ of I' is set to be a dense split graph with $V' = V \cup S$, where V is an independent set and $S = \{s_1, s_2, \ldots, s_k\}$ is the vertex set of a clique of size k. The edge set of G' is therefore $E' = E'_1 \cup E'_2$, where $E'_1 = \{vs \mid v \in V, s \in S\}$ and $E'_2 = \{s_is_j \mid 1 \leq i < j \leq k\}$. Next, we describe the distance matrix D. As always, we have $D_{uu} = 0$ for all $u \in V'$ and $D_{uv} = D_{vu} = 1$ for all $uv \in E'$. Orient G by picking for each edge $uv \in E$ a direction (u, v) arbitrarily, and denote the resulting set of arcs by A. We assume that the arcs corresponding to the four isolated edges mentioned above are (x_2, x_1), (y_2, y_1), (z_2, z_1), and (w_2, w_1). For every $(u, v) \in A$, set $D_{uv} = 2$ and $D_{vu} = 3$. For every pair (u, v) of vertices that do not form an edge in E, set $D_{uv} = D_{vu} = 3$. This completes the construction of I'. We show that I admits a vertex cover of size at most k if and only if I' is a yes-instance of Multi-Label Periodic TGR.

($\Rightarrow$) Let $C = \{c_1, c_2, \ldots, c_k\}$ be a vertex cover of size k for I. (If I has a vertex cover smaller than k, we can add arbitrary vertices to it until $|C| = k$.) In the following, we describe a labeling of the edges of G' that realizes D. This labeling is visualized in Figure 1.

Without loss of generality, assume $c_{k-3} = x_1$, $c_{k-2} = y_1$, $c_{k-1} = z_1$, and $c_k = w_1$. For $i = 1, 2, \dots, k-2$, assign one time label to each edge in E'_1 that is incident with s_i in G' as follows:

- For each incoming arc (u, c_i) of c_i in A, set $\lambda(us_i) := 1$.
- Set $\lambda(c_i s_i) := 2$.
- For each outgoing arc (c_i, v) of c_i in A, set $\lambda(vs_i) := 3$.
- For each other vertex $r \in V \setminus N_G[c_i]$, set $\lambda(rs_i) := 5$.

Note that these labels generate journeys of duration 2 exactly for the incoming and outgoing arcs of c_i in A.

For $i = k-1$ and $i = k$, we do essentially the same, but all assigned time labels are shifted by two time steps. Recall that $c_{k-1} = z_1$ ($c_k = w_1$) and that z_1 (w_1) has no outgoing neighbor and only one incoming neighbor, namely z_2 (w_2). The time labels are assigned as follows:

- Set $\lambda(z_2 s_{k-1}) := \lambda(w_2 s_k) := 5$ and $\lambda(s_{k-1} z_1) := \lambda(s_k w_1) := 6$.
- For each vertex $r \in V \setminus \{z_1, z_2\}$, set $\lambda(rs_{k-1}) = 3$, and
- For each vertex $r \in V \setminus \{w_1, w_2\}$, set $\lambda(rs_k) = 3$.

Note that these labels again generate journeys of duration 2 exactly for the arcs incident with z_1 and w_1, that is, arcs (z_2, z_1) and (w_2, w_1). As C is a vertex cover of G, every edge in E is an incoming or outgoing arc in A of at least one vertex in C. Hence, it is clear that a u-v journey of duration 2 is created for all pairs (u, v) with $D_{uv} = 2$.

Finally, set $\lambda(e) := 4$ for all $e \in E'_2$.

We claim that λ is a solution to I'. First, note that each edge $e \in E'$ receives only a single label. We have already shown above that the journeys of duration 2 that are created are exactly those for all vertex pairs (u, v) with $D_{uv} = 2$. Furthermore, we can show that a journey of duration 3 is generated for each vertex pair $(u, v) \in V \times V$ as follows: Choose $i = k-1$ if $u \notin \{z_1, z_2\}$ and $i = k$ otherwise. Since $\{z_1, z_2\}$ and $\{w_1, w_2\}$ are disjoint, this implies that at time step 3 the edge us_i exists. Similarly, choose $j = k-3$ if $v \notin \{x_1, x_2\}$ and $j = k-2$ otherwise. Since $\{x_1, x_2\}$ and $\{y_1, y_2\}$ are disjoint, this implies that at time step 5 the edge $s_j v$ exists. The u-v journey of duration 3 is then as follows: Take edge us_i at time 3, edge $s_i s_j$ at time 4, and edge $s_j v$ at time 5. Thus, λ solves I', and hence I' is a yes-instance of MULTI-LABEL PERIODIC TGR.

($\Leftarrow$) Assume that λ is a solution that realizes D and maps each $e \in E'$ to a subset of $[1, 6]$. For each $s_i \in S$, let N_i denote the set of edges in E'_1 that are incident with s_i. Note that for every distance $D_{uv} = 2$, there is an $i \in [1, k]$, such that a u-v journey with duration 2 is realized by edges of N_i, since such a journey must pass from u to s_i and then from that s_i to v.

▷ Claim 5. Let P_i be the set of vertex pairs $(u, v) \in V \times V$ for which λ realizes a u-v journey of duration 2 using edges of N_i. Then there exists a vertex u_i that all vertex pairs in P_i have in common.

Proof. We show that it is impossible that λ realizes journeys of duration two for two disjoint vertex pairs (u, v) and (a, b) in $V \times V$ using the edges of N_i. This implies that any two vertex pairs in P_i share a common vertex. As E does not contain cycles of length 3, this implies further that there exists one vertex that is common to all vertex pairs in P_i.

Assume for a contradiction that λ realizes journeys of duration 2 for two disjoint vertex pairs (u, v) and (a, b) in $V \times V$ using edges of N_i. Assume without loss of generality that us_i is present at time 1 and vs_i at time 2. Let j be such that as_i is present at time j and bs_i at time $j + 1$, where we use the convention that $6 + 1 = 1$. Also, let z be a vertex in V

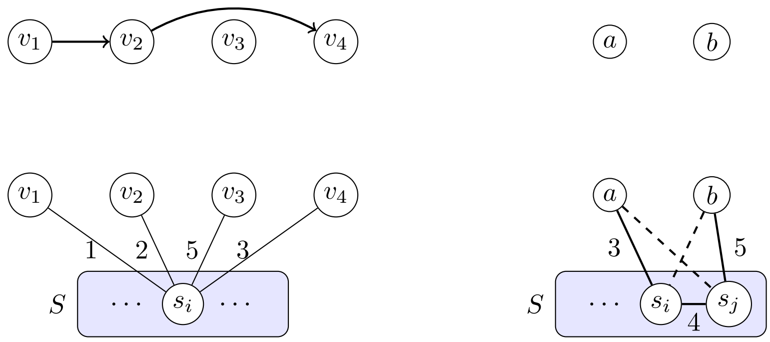

Figure 1 An example how the distances between different vertices are realized in the constructed instance of Multi-Label Periodic TGR in the proof of Theorem 4. Top left shows an input instance of Vertex Cover excluding the additional isolated arcs (w_2, w_1), (x_2, x_1), (y_2, y_1), and (z_2, z_1). Bottom left shows how the distances representing the arcs incident with vertex v_2 can be realized if vertex v_2 is selected to be the ith vertex of the vertex cover. Top right represents any two distinct vertices a and b of G and bottom right shows how an a-b journey of duration 3 can be realized by using two of the four vertices of $\{s_{k-3}, s_{k-2}, s_{k-1}, s_k\} \subseteq S$. Here, $i = k-1$ if $a \notin \{z_1, z_2\}$ and $i = k$ otherwise, and $j = k-3$ if $b \notin \{x_1, x_2\}$ and $j = k-2$ otherwise. The labels of the dashed edges are not depicted, since they depend on a, b, i, and j.

that is adjacent to no vertex in $\{u, v, a, b\}$ in G (such a vertex must exists as we assume that $n \geq 13$, G has maximum degree 3, and G contains the edges uv and ab). Let k be a time step in which the edge $s_i z$ is active. Note that $k \notin \{6, 1, 2, 3\}$ as otherwise z has a journey of duration 2 from or to u or v, a contradiction to λ being a solution while z is adjacent to neither u nor v. Hence, $k \in \{4, 5\}$. By symmetry, we can assume $k = 4$.

Now we show that we obtain a contradiction no matter what the value of j is:

- $j = 1$: We have journeys of duration 2 from u to v, u to b, a to v and a to b, implying that G must contain the 4-cycle $uvab$, a contradiction.
- $j = 2$: We have journeys of duration 2 from u to v, u to a, a to b, and v to b, implying a 4-cycle $uvba$, a contradiction.
- $j = 6$: We have journeys of duration 2 from a to b, a to u, b to v, and u to v, implying a 4-cycle $abvu$, a contradiction.
- $j = 3$: We have an a-z journey of duration 2, a contradiction to z not being adjacent to a.
- $j = 4$: We have a z-b journey of duration 2, a contradiction to z not being adjacent to b.
- $j = 5$: We have a z-a journey of duration 2, a contradiction to z not being adjacent to a.

As all cases lead to a contradiction, the assumption that λ realizes journeys of duration 2 for two disjoint vertex pairs (u, v) and (a, b) cannot hold. ◁

By Claim 5, all vertex pairs for which λ realizes a journey of duration 2 using the edges of N_i have a common vertex u_i. As a journey of duration 2 must be realized for every edge uv of G (either from u to v or from v to u, depending on how the edge has been oriented), the set $U = \{u_1, u_2, \ldots, u_k\}$ is a vertex cover of G. Furthermore, $|U| \leq k$, and hence I is a yes-instance of Vertex Cover. ◀

Note that in contrast to hardness for $d_{\max} = 3$ that we have just shown, for each $\ell \geq 2$, it is not difficult to see that Multi-Label Periodic TGR can be solved in polynomial time if $d_{\max} \leq 2$: If $\Delta = 1$, the problem is polynomial time solvable, since there is only a single temporal multi-labeling. If $d_{\max} = 1$, the underlying graph is a clique and the distance matrix is realizable by any periodic multi-labeling. If $d_{\max} = 2$, then the instance is a trivial no-instance, if the underlying graph has diameter larger than 2. Otherwise, if the underlying graph has diameter at most two, we can label each edge with label 1 and 2 (since $\ell \geq 2$) and so ensure a path between any two vertices of duration 2 that starts at time step 1.

▶ **Observation 6.** *For each $\ell \geq 2$ Multi-Label Periodic TGR can be solved in polynomial time if $d_{\max} \leq 2$.*

We now shift to considering the structure of the realized graph. Next, we show that for $\ell \geq 5$, Multi-Label Periodic TGR is *NP*-hard even on stars. This implies that Multi-Label Periodic TGR is *NP*-hard even if $\ell + \mathrm{vc} \in \mathcal{O}(1)$, and so FPT algorithms for parameter $\ell + \mathrm{vc}$ are impossible, unless $P = NP$.

▶ **Theorem 7** ($\star$)**.** *For each $\ell \geq 5$, Multi-Label Periodic TGR is NP-hard even if the underlying graph is a star.*

Proof sketch. Let $\ell \geq 5$. We again reduce from Vertex Cover where the input graph contains no cycle of length three or four, the input graph has a maximum degree of 3, and no two vertices of degree 3 are adjacent.

Let $I := (G = (V, E), k)$ be an instance of Vertex Cover with the above restrictions and let $n := |V|$ be larger than 10. We construct an instance $I' := (\ell, D, \Delta)$ of Multi-Label Periodic TGR as follows: The underlying graph $G' := (V', E')$ of I' is a star with center c and leaf set $V \cup \{v^*, w^*\}$. We set $D_{w^*v^*} := D_{v^*w^*} := n^2$ and for each vertex $v \in V$, we set $D_{vv^*} := D_{v^*v} := D_{vw^*} := D_{w^*v} := n^2$. For each two distinct vertices u and v of V, we set $D_{uv} := D_{vu} := 2$ if uv is an edge of E, and $D_{uv} := D_{vu} := n^2$, otherwise. Finally, we set $\Delta := (k + 2) \cdot (n^2 + 1) = k \cdot n^2 + 2n^2 + k + 2$. This completes the construction of I'.

Note that each temporal path between any two vertices u and v of G' distinct from the center vertex c is of the form ucv. Since each vertex of $V' \setminus \{c\}$ has only one incident edge in G', we may say in the following that for a temporal multi-labeling, a vertex $v \in V' \setminus \{c\}$ is *active* in time step i, if the edge cv exists in time step i.

Observe that for each vertex $u \in V$, journeys of travel time 2 from u to all its neighbors in G and vice versa can be realized in three consecutive time steps: u is active in the first and the third of these time steps and all vertices of $N_G(u)$ are active in the second time step. Hence, the journeys of duration 2 from u to all its neighbors in G start in the first time step and end in the second time step, and the journeys of duration 2 from all neighbors of u in G to u start in the second time step and end in the third time step. If G admits a vertex cover of size k, all required journeys of travel time 2 can thus be realized in k such groups of three consecutive time steps, with a separation of $n^2 - 2$ edgeless time steps between them (to avoid creating journeys of travel time shorter than n^2 between pairs of vertices that are independent in G). In addition to these $k(n^2 + 1)$ time steps, a further $2(n^2 + 1)$ time steps can be used to realize all the required journeys of travel time n^2. For the other direction, we can show that any feasible realization must have a similar structure, implying the existence of a vertex cover of size at most k. The detailed proof of correctness is deferred to the full version. ◀

Since the hardness result above only holds for $\ell \geq 5$, we note that we can also show that for $\ell \geq 3$, the problem is still *NP*-hard even on graphs with a size-1 feedback vertex set.

▶ **Theorem 8** ($\star$)**.** *For each $\ell \geq 3$, Multi-Label Periodic TGR is NP-hard even if the underlying graph has diameter two and a feedback vertex set of size one.*

4 Parameterized algorithms

In this section, we present FPT algorithms for Multi-Label Periodic TGR and Multi-Label TGR for several parameter combinations. First, we give FPT algorithms for both problems parameterized by n in Section 4.1. Section 4.2 presents our FPT algorithm for Multi-Label Periodic TGR parameterized by $\mathrm{vc} + \Delta$. Section 4.3 discusses our $O(pn^4)$-time algorithm for instances of Multi-Label TGR with finite-valued D, $\ell \geq n^2$, and underlying graphs that have at most p different u-v paths for each vertex pair (u, v). Finally, the polynomial kernel for parameter $d_{\max} + \mathrm{nu}$ is shown in Section 4.4, where nu denotes the number of non-universal vertices in the underlying graph.

4.1 Parameterization by the number of vertices

In this section, we present an FPT algorithm for parameter n for Multi-Label Periodic TGR. The key idea is to enumerate all possibilities for the sequence of snapshots that contain at least one edge (and some extra information that specifies for each pair (u, v) a snapshot in which a u-v journey of shortest duration begins). For each possibility, we use an integer linear program (ILP) to decide whether it is possible to assign these snapshots to time steps in such a way that the resulting periodic temporal graph realizes D. The approach extends to Multi-Label TGR as well.

▶ **Theorem 9.** *Multi-Label Periodic TGR can be solved in $n^{\mathcal{O}(\ell \cdot n^2)} \cdot |I|^{\mathcal{O}(1)}$ time and $n^{\mathcal{O}(n^4)} \cdot |I|^{\mathcal{O}(1)}$ time, where $|I|$ denotes the encoding length of the instance.*

Proof. Let an instance $I = (\ell, D, \Delta)$ of Multi-Label Periodic TGR be given. Recall that we can assume that $\ell \leq n^2$ due to Lemma 1. Let $K = \ell m$ and $T = \min\{K, \Delta\}$. Note that $T \leq \ell m \in \mathcal{O}(n^2 \cdot n^2) \subseteq n^{\mathcal{O}(1)}$. We observe that there are at most K non-empty snapshots in any realization, as each of the m edges can occur in at most ℓ snapshots. Furthermore, it is clear that there are at most Δ non-empty snapshots, so the number of non-empty snapshots is at most T. The number of sequences of at most T non-empty snapshots in which each of the m edges occurs in at most ℓ snapshots (and in at least one snapshot) can be bounded by $T^{\ell m} = T^K$, as each such sequence can be encoded by assigning to each of ℓm edge copies (with ℓ copies of each edge) a number in $[1, T]$ that identifies the snapshot in which it occurs. (If an edge occurs fewer than ℓ times, this can be captured by assigning some of its copies the same number.) Thus, we can enumerate all such sequences in $T^{\mathcal{O}(K)} \subseteq n^{\mathcal{O}(\ell \cdot n^2)}$ time.

For each such sequence $\mathcal{S}$ of non-empty snapshots, we enumerate all possibilities of assigning to each vertex pair (u, v) with $D_{uv} > 1$ a number s_{uv} in $[1, |\mathcal{S}|]$ that identifies the snapshot in which the journey from u to v that realizes the duration D_{uv} starts. (Note that if $s_{uv} = i$, this means that the journey starts in the i-th non-empty snapshot. That snapshot will be present in some time step t_i in $[1, \Delta]$ that has not yet been determined.) The number of possibilities to be enumerated is bounded by $T^{n^2} \subseteq n^{\mathcal{O}(n^2)}$.

Intuitively, we want to proceed along the following lines: For each combination of a sequence $\mathcal{S}$ of snapshots and an assignment of values s_{uv} to vertex pairs (u, v) with $D_{uv} > 1$, we want to use an ILP to check whether we can assign the i-th snapshot of $\mathcal{S}$ to a time step t_i, for all i, in such a way that the resulting periodic temporal graph realizes D. To be able to formulate the constraints of the ILP, we use an auxiliary temporal graph, without gaps between the snapshots of $\mathcal{S}$, to determine for each pair (u, v) of vertices and each starting snapshot i the snapshot at which v can first be reached if starting at u in snapshot i. The constraints of the ILP can then express that the gaps inserted between the snapshots must be such that (1) the duration of the u-v journey starting in snapshot s_{uv} is equal to D_{uv}, and (2) the duration of the u-v journey starting in any other snapshot is at least D_{uv}. The variables of the ILP represent the time steps to which the snapshots get assigned.

Formally, we process each combination of a sequence $\mathcal{S}$ of snapshots and assignment of values s_{uv} to vertex pairs (u, v) with $D_{uv} > 1$ as follows. Let $L = |\mathcal{S}|$ and $\mathcal{S} = (S_1, S_2, \ldots, S_L)$. Build a periodic temporal graph $\mathcal{G}_\mathcal{S}$ with period L such that the edges present in time steps $i + zL$ for all integers $z \geq 0$ are those of S_i, for $1 \leq i \leq L$. For every $(u, v) \in V \times V$ with $D_{uv} > 1$ and every $i \in [1, L]$, we denote by $q(u, v, i)$ the tuple (j, z), such that each fastest u-v journey in $\mathcal{G}_\mathcal{S}$ starting at time i ends in time step $j + zL$, where $1 \leq j \leq L$. Note that computing $q(u, v, i)$ can be done in polynomial time, since finding a path of shortest duration from u to v starting at time step i can be done in polynomial time on non-periodic temporal graphs [3, 17], and we can simply unroll $\mathcal{G}_\mathcal{S}$ n times to obtain a non-periodic temporal graph on which the shortest duration of any journey between u and v remains the same. This is discussed in detail for a more general class of periodic temporal graphs in [2, Remark 1]. Note that in a periodic temporal graph, for any pair (u, v) of vertices for which a u-v journey exists, there exists a u-v journey of shortest duration that starts in a snapshot of the first period. Therefore, it suffices to consider only start times $i \in [1, L]$ for u-v journeys in $\mathcal{G}_\mathcal{S}$.

We want to determine whether there exist values t_i for $1 \leq i \leq L$ with $1 \leq t_1 < t_2 < \cdots < t_L \leq \Delta$ such that the periodic temporal graph $\mathcal{G}$ defined as follows realizes D:

- for each $i \in [1, L]$, S_i is the set of edges of $\mathcal{G}$ that are present in time step t_i, and
- no edge is present in any of the time steps in $[1, \Delta] \setminus \{t_i \mid 1 \leq i \leq L\}$.

Observe that $q(u, v, i) = (j, z)$ if and only if the earliest-arrival journey from u to v in $\mathcal{G}$ starting at time t_i reaches v in time step $t_j + z\Delta$ (and thus has duration $\delta(u, v, t_i) = t_j + z\Delta - t_i + 1$). This is because $\mathcal{G}_\mathcal{S}$ can be obtained from $\mathcal{G}$ by removing all empty snapshots.

The temporal graph $\mathcal{G}$ realizes D if, for all (u, v) with $D(u, v) > 1$, $\delta(u, v, t_i) = D_{uv}$ for at least one t_i and $\delta(u, v, t_i) \geq D_{uv}$ for all t_i. The purpose of the value s_{uv} that we have enumerated is to give a value of i with the property that $\delta(u, v, t_i) = D_{uv}$. We can then formulate the following ILP with variables $t_1, t_2, \ldots, t_L$ to check whether there exist values of these variables such that $\mathcal{G}$ realizes D:

$$\begin{aligned}
& t_j + z\Delta - t_i + 1 = D_{uv} && \forall (u, v) \text{ with } D_{uv} > 1,\ i = s_{uv},\ q(u, v, i) = (j, z) \\
& t_j + z\Delta - t_i + 1 \geq D_{uv} && \forall (u, v) \text{ with } D_{uv} > 1,\ \forall i \neq s_{uv},\ q(u, v, i) = (j, z) \\
& t_1 \geq 1 && \\
& t_i - t_{i-1} \geq 1 && \forall i \text{ with } 2 \leq i \leq L \\
& t_L \leq \Delta &&
\end{aligned} \tag{ILP}$$

Note that there is no objective function as we only want to check feasibility, i.e., check whether there exist values $t_1, \ldots, t_L$ that satisfy the constraints. The first two constraints express that the earliest-arrival path from u to v starting in time step $t_{s_{uv}}$ has duration D_{uv} and the earliest-arrival paths from u to v starting in any other time step have duration at least D_{uv}. The last three constraints ensure that $1 \leq t_1 < t_2 < \cdots < t_L \leq \Delta$. Thus, a feasible solution of the ILP gives a periodic temporal graph that is a solution to the given instance I of Multi-Label Periodic TGR.

As the ILP has $L \leq T$ variables, we can solve each such ILP instance in $(\log L)^{\mathcal{O}(L)} \cdot |I|^{\mathcal{O}(1)} \subseteq (\log n)^{\mathcal{O}(\ell \cdot n^2)} \cdot |I|^{\mathcal{O}(1)}$ time [16]. We solve the ILP once for each combination of a sequence $\mathcal{S}$ of non-empty snapshots and an assignment of values s_{uv} to all pairs (u, v) with $D_{uv} > 1$. Thus, we solve $n^{\mathcal{O}(\ell \cdot n^2)}$ different ILPs. The resulting overall running-time is then $n^{\mathcal{O}(\ell \cdot n^2)} \cdot |I|^{\mathcal{O}(1)}$. The claimed running-time follows because we can assume $\ell \leq n^2$. The algorithm is correct because, if I is a yes-instance, one of the enumerated combinations of a sequence of snapshots and an assignment of values s_{uv} corresponds to a realization of D, and for that combination a realization of D will be obtained from the solution of the ILP. ◀

For the special case $\ell = 1$ of Multi-Label Periodic TGR, our FPT algorithm of Theorem 9 has a running time of $n^{\mathcal{O}(n^2)} \cdot |I|^{\mathcal{O}(1)}$ and is a conceptually simpler FPT algorithm for Periodic TGR than the FPT algorithm for n that is implied by the algorithm for Periodic TGR parameterized by the feedback edge number of the underlying graph from [12]. The main advantage of our approach is that it extends to arbitrary ℓ. Furthermore, our approach also works for the non-periodic version.

▶ **Corollary 10** ($\star$). *Multi-Label TGR can be solved in $n^{\mathcal{O}(\ell \cdot n^2)} \cdot |I|^{\mathcal{O}(1)}$ time and $n^{\mathcal{O}(n^4)} \cdot |I|^{\mathcal{O}(1)}$ time, where $|I|$ denotes the encoding length of the instance.*

4.2 Parameterization by the vertex cover number plus the period

In this section we give an FPT algorithm for Multi-Label Periodic TGR parameterized by $\mathrm{vc} + \Delta$. The key idea of our approach is to show that any given instance can be reduced to one where the number of vertices in the independent set that have the same neighbors can be bounded by a function of the parameter. It then suffices to apply the FPT algorithm for parameter n to this reduced instance.

▶ **Theorem 11.** *There is an FPT algorithm for Multi-Label Periodic TGR parameterized by* $\mathrm{vc} + \Delta$.

Recall that ℓ is upper bounded by Δ. Hence, to show Theorem 11, it is sufficient to present an FPT algorithm for parameter $\mathrm{vc} + \Delta + \ell$. Let (ℓ, D, Δ) be the given instance of Multi-Label Periodic TGR. Recall that vc denotes the size of a minimum vertex cover of the underlying graph $G = (V, E)$. Observe that the number of possible label sets assigned to any particular edge $e \in E$ can be bounded by Δ^ℓ: Each of the up to ℓ labels assigned to the edge is a value in $[1, \Delta]$, and combinations where fewer than ℓ labels are assigned to the edge can be modeled as assigning the same label several times.

We call two vertices $u, v \in V$ *distance twins* if they have the same distance to every vertex in $V \setminus \{u, v\}$ and to each other. This means that their rows in the distance matrix D are identical, and their columns in D are identical, up to the obvious difference in the intersection of their rows with their columns: $D_{uu} = 0$, $D_{uv} = D_{vu}$, and $D_{vv} = 0$. Note that the distance twin relation is an equivalence relation on V.

Let C be a vertex cover of G of size vc, and let $I = V \setminus C$ be the independent set that is the complement of C. Partition I into *neighborhood classes* $\mathcal{I} = \{I_1, I_2, \dots, I_t\}$ based on adjacency to C, i.e., two vertices $u, v \in I$ are in the same class I_j if and only if $N(u) = N(v)$. Note that, $t \leq 2^{\mathrm{vc}}$.

Consider one part I_j of the partition $\mathcal{I}$. For each vertex $u \in I_j$, there are at most $\Delta^{\mathrm{vc} \cdot \ell}$ different ways of assigning label sets to the (at most vc) edges incident with u. For any fixed labeling λ of E, call two vertices $u, v \in I_j$ *label twins* if for every vertex $w \in N(u)$, $\lambda(uw) = \lambda(vw)$. The label twin relation partitions the set I_j into equivalence classes that we call *label classes*. Note that there can be at most $\Delta^{\mathrm{vc} \cdot \ell}$ label classes for I_j.

Observe that all vertices u, v in a label class have the same distance to every vertex in $V \setminus \{u, v\}$ and to each other. This means that, if λ realizes D, then u and v must be distance twins. The distance twin relation partitions I_j into *distance classes*. If I_j contains more than $\Delta^{\mathrm{vc} \cdot \ell}$ distance classes, the given instance is a no-instance, because at most $\Delta^{\mathrm{vc} \cdot \ell}$ different distance classes can be realized by the at most $\Delta^{\mathrm{vc} \cdot \ell}$ different label classes.

▶ **Observation 12.** *If a neighborhood class contains more than $\Delta^{\mathrm{vc} \cdot \ell}$ distance classes, then the considered instance is a trivial no-instance of Multi-Label Periodic TGR.*

If I_j contains at most $\Delta^{\text{vc}\cdot\ell}$ different distance classes, it suffices to keep $2\Delta^{\text{vc}\cdot\ell}$ of the vertices in each such class, while any additional vertices of the class can be deleted (i.e., the corresponding rows and columns of there vertices can be removed from D).

▶ **Lemma 13** ($\star$). *Let I_j be a neighborhood class and let F denote a distance class of I_j of size more than $2\Delta^{\text{vc}\cdot\ell}$. Then removing one vertex of F from G yields an equivalent instance.*

With this statement at hand, we are now able to present the algorithm for Multi-Label Periodic TGR.

Proof of Theorem 11. For each neighborhood class I_j of $\mathcal{I}$ and each distance class F of I_j with $|F| > 2\Delta^{\text{vc}\cdot\ell}$, remove $|F| - 2\Delta^{\text{vc}\cdot\ell}$ arbitrary vertices from F. Due to Lemma 13, this yields an equivalent instance, where each distance class contains at most $2\Delta^{\text{vc}\cdot\ell}$ vertices. If the resulting instance contains at most $\text{vc} + 2^{\text{vc}} \cdot 2\Delta^{2\text{vc}\cdot\ell}$ vertices, the FPT algorithm is obtained by applying the algorithm behind Theorem 9. Otherwise, if the resulting instance contains more than $\text{vc}+2^{\text{vc}}\cdot 2\Delta^{2\text{vc}\cdot\ell}$ vertices, there is a neighborhood class I_j such that I_j has more than $\Delta^{\text{vc}\cdot\ell}$ distance classes. This is correct, since the new instance contains exactly vc vertex cover vertices, at most 2^{vc} neighborhood classes, and at most $2\Delta^{\text{vc}\cdot\ell}$ vertices in each distance class. Due to Observation 12, we can thus correctly output that the instance under consideration is a trivial no-instance of Multi-Label Periodic TGR. ◀

Recall that Multi-Label Periodic TGR is *NP*-hard even if $\ell = 1$ and $\Delta = 3$ [12] and that Theorem 7 shows that Multi-Label Periodic TGR is *NP*-hard even if $\ell = 5$ and $\text{vc} = 1$. Hence, neither of the considered parameters can be omitted to still obtain an FPT algorithm for Multi-Label Periodic TGR. Still, the question remains whether there is an FPT algorithm parameterized by vc alone for the case $\ell = 1$ (Periodic TGR), or if one can replace Δ in the combined parameter by some potentially smaller parameter. In particular, the parameterized complexity of Multi-Label Periodic TGR when parameterized by $\text{vc} + d_{\max} + \ell$ is open.

4.3 Efficient algorithm for Multi-Label TGR on graphs with few paths

While we have shown Multi-Label Periodic TGR to be *NP*-hard for any $\ell \geq 5$ even if the underlying graph is a star, we show in this section that Multi-Label TGR can be solved in polynomial time if the underlying graph is a tree and $\ell \geq n(n-1)$ and D is finite-valued. In fact, our result is more general and solves the problem in polynomial time whenever the number of different u-v paths in the underlying graph can be bounded by a polynomial, for each vertex pair (u, v).

As a subproblem we consider the problem Path Realization that is defined as follows: Given a distance matrix D, a pair (u, v) with $u \neq v$, and a u-v path $P = (u_0 = u, u_1, \ldots, u_r = v)$ in the underlying graph, decide if one can assign one time label to each edge on P in such a way that, in the temporal graph on P with those time labels, the u-v journey has duration D_{uv}, while any u_i-u_j journey with $0 \leq i < j \leq r$ has duration at least $D_{u_i u_j}$. Here, we use the convention that the duration of a u_i-u_j journey is ∞ if there is no u_i-u_j journey in the temporal graph on P with the time labels assigned.

Intuitively, the purpose of solving Path Realization is to decide whether it is possible to assign time labels to the edges of P in such a way that a u-v journey of duration D_{uv} is realized while no u'-v' journey that is too short (i.e., has duration strictly less than $D_{u'v'}$) is created. The key ingredient of the proof is the following lemma that shows that Path Realization can be solved in polynomial time. For the main result of this section we only need to be able to solve Path Realization for finite-valued distance matrices, but since our approach can also handle entries equal to ∞, we present the lemma for this more general case.

▶ **Lemma 14** ($\star$). *Path Realization can be solved in $O(r^2)$ time.*

We say that the underlying graph $G = (V, E)$ is *p-path-diverse* if the number of u-v paths in G is bounded by p for each pair (u, v) of vertices in V.

▶ **Theorem 15** ($\star$). *Multi-Label TGR can be be solved in $O(pn^4)$ time if the underlying graph is p-path-diverse and $\ell \geq n(n-1)$ and D is finite-valued.*

The maximum number of u-v paths for any vertex pair (u, v) can be bounded by $n!$, so the running-time of the algorithm of Theorem 15 is bounded by $O(n! \cdot n^4)$. Thus, the algorithm is an FPT algorithm for Multi-Label TGR parameterized by n that is simpler and more efficient than that of Theorem 9 in Section 4.1, but only works for instances with a finite-valued distance matrix and $\ell \geq n(n-1)$.

As trees are 1-path-diverse and cycles are 2-path-diverse, we obtain the following corollary.

▶ **Corollary 16.** *Multi-Label TGR can be be solved in $O(n^4)$ time if the underlying graph is a tree or a cycle, $\ell \geq n(n-1)$, and D is finite-valued.*

4.4 A polynomial kernel

In this section, we present a kernel for Multi-Label Periodic TGR for the combined parameter $d_{\max} + \mathrm{nu}$, where $\mathrm{nu} := |\{v \in V \mid N[v] \neq V\}|$ denotes the number of non-universal vertices of the underlying graph. Note that nu is never larger than the number of entries of D of value larger than 1, since for each non-edge $\{u, v\}$ of G, $D_{uv} > 1$ and $D_{vu} > 1$. Hence, the kernel we present also implies a kernel for Multi-Label Periodic TGR for the parameter combination $d_{\max} + |\{D_{uv} \mid u, v \in V, D_{uv} > 1\}|$. We also show that this kernel transfers to Multi-Label TGR for finite-valued distance matrices.

▶ **Theorem 17** ($\star$). *Multi-Label Periodic TGR admits a kernel of size $\mathcal{O}(\min\{\ell \cdot \mathrm{nu}^4 \cdot d_{\max}, \mathrm{nu}^8 \cdot d_{\max}\})$. More precisely, this kernel has $\mathcal{O}(\mathrm{nu}^2 \cdot d_{\max})$ vertices and a period of $\mathcal{O}(\min\{\ell \cdot \mathrm{nu}^4 \cdot d_{\max}, \mathrm{nu}^8 \cdot d_{\max}\})$ and does not increase the value of ℓ.*

Proof. Let $I := (\ell, D, \Delta)$ be an instance of Multi-Label Periodic TGR where $d_{\max}$ denotes the largest non-infinite entry of D and where $G = (V, E)$ is the underlying graph implied by the distance matrix D. Moreover, let X denote the set of all vertices of G that are not universal and let $\mathrm{nu} := |X|$.

If $n \in \mathcal{O}(\mathrm{nu}^2)$, then D contains $\mathcal{O}(\mathrm{nu}^4)$ entries of size at most $d_{\max}$ each. Moreover, due to Lemma 2, we can reduce Δ to $\mathcal{O}(\ell \cdot \mathrm{nu}^4 \cdot d_{\max}) \subseteq \mathcal{O}(\mathrm{nu}^8 \cdot d_{\max})$. This then implies a polynomial kernel of the desired size. Hence, in the following, we assume that $n > \mathrm{nu}^2$. Note that this implies that there is at least one universal vertex v^* in G. Hence, for any two vertices u and w of G distinct from v^*, there is the path uv^*w of length two in G. Consequently if I is a yes-instance of Periodic TGR, the largest possible time that can be realized between any two vertices of G is $\Delta + 1$, since traversing the path uv^*w takes time at most $\Delta + 1$. In other words, I is a trivial no-instance of Periodic TGR if D contains an entry larger than $\Delta + 1$. In the following, we thus assume that $d_{\max} \leq \Delta + 1$. We distinguish two cases.

Case 1: $\Delta \geq 3d_{\max}$. We show in this case that I is a trivial yes-instance of Multi-Label Periodic TGR even when assigning only a single label to each edge. This proof is deferred to the full version.

Case 2: $\Delta < 3d_{\max}$. Let R_X denote the set of directed pairs of vertices of X for which the distance is not trivially realized, that is, $R_X := \{(u,v) \in X \times X \mid u \neq v, D_{uv} > 1\}$. If G contains no more than $\mathrm{nu}^2 \cdot d_{\max}$ universal vertices, then $n \in \mathcal{O}(\mathrm{nu}^2 \cdot d_{\max})$ and the polynomial kernel follows directly, since $\ell \leq \Delta \in \mathcal{O}(d_{\max})$ and D contains $\mathcal{O}(\mathrm{nu}^2)$ entries of value larger than 1. Otherwise, if G contains at least $\mathrm{nu}^2 \cdot d_{\max}$ universal vertices, we remove an arbitrary set Z of universal vertices from G such that $\mathrm{nu}^2 \cdot d_{\max}$ universal vertices remain. This then gives the kernel of desired size due to the above argumentation. In the full version, we show that the so obtained instance $I' := (\ell, D', \Delta)$ of Multi-Label Periodic TGR is a yes-instance if and only if I is a yes-instance of Multi-Label Periodic TGR. ◀

Note that this implies the following polynomial kernel for Periodic TGR.

▶ **Corollary 18.** *Periodic TGR admits a polynomial kernel of size $\mathcal{O}(\mathrm{nu}^4 \cdot d_{\max}^2)$. More precisely, this kernel has $\mathcal{O}(\mathrm{nu}^2 \cdot d_{\max})$ vertices and a period of $\mathcal{O}(\mathrm{nu}^4 \cdot d_{\max})$.*

Moreover, we can derive the following result for Multi-Label TGR on finite-valued distance matrices.

▶ **Corollary 19** ($\star$)**.** *On finite-valued distance matrices, Multi-Label TGR is FPT when parameterized by* nu *and admits a kernel of size $\mathcal{O}(\mathrm{nu}^4 + \mathrm{nu}^2 \cdot d_{\max})$. More precisely, this kernel has $\mathcal{O}(\mathrm{nu}^2)$ vertices and does not increase the value of ℓ.*

5 Conclusion and open questions

In this paper, we have studied multi-label versions of the temporal realization problem introduced by Klobas et al. [12] and presented various hardness results and FPT algorithms for different parameters or parameter combinations. There are a number of interesting directions for future work. While our hardness results exclude FPT algorithms for Multi-Label Periodic TGR parameterized by the vertex cover number alone (unless $P = NP$), the question whether such an FPT algorithm exists for Periodic TGR remains open. With respect to the polynomial kernel of size $\mathcal{O}(\mathrm{nu}^4 \cdot d_{\max}^2)$ that we have obtained, an interesting question is whether a kernel whose size is a polynomial of nu alone exists. To answer this question, one first has to analyze whether the problem admits a polynomial kernel for parameter n alone. A question in relation to our FPT algorithms for parameter n is whether the subproblem that we solve using ILP can be solved more efficiently using a combinatorial algorithm.

Furthermore, it would be interesting to analyze the computational complexity of Multi-Label Periodic TGR on stars and trees for $\ell \in \{2,3,4\}$. Klobas et al. [12] have shown that the problem on trees is polynomial for $\ell = 1$, while we have shown that it is *NP*-hard on stars for $\ell \geq 5$, so the status for $\ell \in \{2,3,4\}$ is open for stars and trees.

For Multi-Label TGR, *NP*-hardness has so far only been shown in the case that the distance matrix can have entries equal to ∞ and $\ell = 1$ [12]. It would be interesting to analyze the complexity for finite-valued distance matrices and for $\ell > 1$. For the problem variant where a maximum allowed label L is specified as part of the input (i.e., a bound on the lifetime of the temporal graph that can be built to realize D), our *NP*-hardness proofs of Section 3 should translate. We expect that our FPT algorithms for parameter n and for parameter $\mathrm{vc} + \Delta$ (which then becomes $\mathrm{vc} + L$) can also be adapted to that case even if the distance matrix contains entries of value ∞.

Given that we have shown that Multi-Label Periodic TGR is *NP*-hard even if $d_{\max} = 3$ for every $\ell \geq 1$, and that it can be solved in polynomial time if $d_{\max} \leq 2$ for all $\ell \geq 2$, it would be interesting to settle the complexity of the problem for $\ell = 1$ and $d_{\max} = 2$.

References

1 Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory Comput. Syst.*, 61(3):907–944, 2017. doi:10.1007/S00224-017-9757-X.

2 Emmanuel Arrighi, Niels Grüttemeier, Nils Morawietz, Frank Sommer, and Petra Wolf. Multi-parameter analysis of finding minors and subgraphs in edge periodic temporal graphs. *CoRR*, abs/2203.07401, 2022. doi:10.48550/ARXIV.2203.07401.

3 Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *Int. J. Found. Comput. Sci.*, 14(2):267–285, 2003. doi:10.1142/S0129054103001728.

4 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *Int. J. Parallel Emergent Distributed Syst.*, 27(5):387–408, 2012. doi:10.1080/17445760.2012.668546.

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

6 Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2000.

7 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

8 Jessica A. Enright, Kitty Meeks, and Fiona Skerman. Assigning times to minimise reachability in temporal graphs. *J. Comput. Syst. Sci.*, 115:169–186, 2021. doi:10.1016/J.JCSS.2020.08.001.

9 Paul Erdős and Tibor Gallai. Graphs with points of prescribed degree (in Hungarian). *Mat. Lapok*, 11:264–274, 1961.

10 S. L. Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quarterly of Applied Mathematics*, 22(4):305–317, 1965.

11 Nina Klobas, George B. Mertzios, Hendrik Molter, and Paul G. Spirakis. The complexity of computing optimum labelings for temporal connectivity. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, volume 241 of *LIPIcs*, pages 62:1–62:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.MFCS.2022.62.

12 Nina Klobas, George B. Mertzios, Hendrik Molter, and Paul G. Spirakis. Temporal graph realization from fastest paths. In *3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2024)*, volume 292 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SAND.2024.16.

13 George B. Mertzios, Othon Michail, and Paul G. Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019. doi:10.1007/S00453-018-0478-6.

14 Othon Michail and Paul G. Spirakis. Elements of the theory of dynamic networks. *Commun. ACM*, 61(2):72, 2018. doi:10.1145/3156693.

15 Owen J. Murphy. Computing independent sets in graphs with large girth. *Discret. Appl. Math.*, 35(2):167–170, 1992. doi:10.1016/0166-218X(92)90041-8.

16 Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 974–988. IEEE, 2023. doi:10.1109/FOCS57990.2023.00060.

17 Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *Proc. VLDB Endow.*, 7(9):721–732, 2014. doi:10.14778/2732939.2732945.