Sliding into the Future: Investigating Sliding Windows in Temporal Graphs

Nina Klobas
Department of Computer Science, Durham University, UK

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

Paul G. Spirakis
Department of Computer Science, University of Liverpool, UK

Abstract

Graphs are fundamental tools for modelling relations among objects in various scientific fields. However, traditional static graphs have limitations when it comes to capturing the dynamic nature of real-world systems. To overcome this limitation, *temporal graphs* have been introduced as a framework to model graphs that change over time. In temporal graphs the edges among vertices appear and disappear at specific time steps, reflecting the temporal dynamics of the observed system, which allows us to analyse time dependent patterns and processes. In this paper we focus on the research related to *sliding time windows* in temporal graphs. Sliding time windows offer a way to analyse specific time intervals within the lifespan of a temporal graph. By sliding the window along the timeline, we can examine the graph's characteristics and properties within different time periods.

This paper provides an overview of the research on sliding time windows in temporal graphs. Although progress has been made in this field, there are still many interesting questions and challenges to be explored. We discuss some of the open problems and highlight their potential for future research.

2012 ACM Subject Classification Mathematics of computing → Graph theory; General and reference → Surveys and overviews

Keywords and phrases Temporal Graphs, Sliding Time Windows

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.5

Category Invited Talk

Funding *George B. Mertzios*: Supported by the EPSRC grant EP/P020372/1.
Paul G. Spirakis: Supported by the EPSRC grant EP/P02002X/1.

1 Introduction

Graphs are used to model (binary) relations among different objects. They consist of a set of vertices, where two of them are connected together with an edge. They have become a fundamental tool for modelling diverse systems and real-world problems, steaming through the wide range of scientific fields. Let us mention just a few of them. In Social sciences they can be used to model different interactions among people (for example friendships, communications, etc.), in Chemistry they can model chemical compounds where the vertices represent different atoms of the compound and edges correspond to the chemical bonds among them, in molecular Biology they can model physical interactions between proteins, gene co-expression or biochemical reactions, in Physics they model interactions among particles, where nodes represent interactions where particles are created or destroyed and edges are particles traveling between the interactions. Having such a varied application and use, it is not surprising that the graph theory has been the subject of extensive research over the past centuries.

When studying real-life problems, it becomes evident that this 'simple' graph model is often insufficient. Many problems exhibit dynamic behavior, where the connections or interactions among their vertices change over time. For instance, in transportation networks, specific roads may be closed during certain intervals. In social networks, individuals may only interact at particular times of the day or month. Similarly, in information and communication networks, information or messages are transmitted from a source to a recipient through a set of connections at specific times. These graph models share a common attribute: their underlying graph topology or network structure undergoes discrete changes over time. This observation naturally gives rise to the concept of temporal graphs, which provide a straightforward and intuitive model for representing graphs that change over time, called *temporal graphs*.

▶ **Definition 1** (temporal graph [24]). *A* temporal graph $\mathcal{G}$ *is a pair* (G, λ)*, where* $G = (V, E)$ *is an underlying (static) graph and* $\lambda : E \to 2^{\mathbb{N}}$ *is a* time labeling *function which assigns to every edge of* G *a set of discrete time labels.*

Due to their relevance and applicability in many areas, temporal graphs have been studied from various perspectives and under different names such as *dynamic* [9,19], *evolving* [7,12,16], *time-varying* [1,17,36], and *graphs over time* [27].

In most applications of temporal graphs, information can naturally only move along edges in a way that respects the ordering of their timestamps (i.e. time labels), that is, information can only flow along sequences of edges whose time labels are increasing (or non-decreasing). Motivated by this fact, most studies on temporal graphs have focused on "path-related" problems, such as e.g. temporal analogues of distance, diameter, reachability, exploration, and centrality [2,3,10,14,15,22,25,26,30,34,40]. In these problems, the most fundamental notion is that of a *temporal path* from a vertex u to a vertex v, which is a path from u to v such that the time labels of the time labels of the edges are increasing (or at least non-decreasing) in the direction from u to v. To complement this direction, several attempts have been recently made to define meaningful "non-path" temporal graph problems which appropriately model specific applications. Some examples include temporal cliques, cluster editing, temporal vertex cover, temporal graph coloring, temporally transitive orientations of temporal graphs [4,6,11,18,21,23,32,33,37,39].

One of the main goals in temporal graphs' research is to lift (algorithmic) graph theory models and results to a temporal/dynamic domain, in order to model natural, real world situations which are subject to discrete changes over time. The main challenge in this front is to find appropriate natural extensions and definitions of such problems. For instance, in static graphs, a *shortest path* between two vertices is a path connecting these two vertices with the smallest number of edges. On the other hand, in temporal graphs, there are at least three, equally natural, different analogues of a shortest path. First, a *shortest* temporal path from u to v is one that contains the smallest number of edges. Second, a *foremost* temporal path from u to v is one that arrives at v with the smallest time-stamp. Third, a *fastest* temporal path from u to v is one that has the smallest duration. These three types of temporal paths are illustrated in Figure 1.

What is common to most of the path-related problems is that their extension from static to temporal graphs often follows easily and quite naturally from their static counterparts. For example, requiring a graph to be (temporally) connected results in requiring the existence of a (temporal) path among each pair of vertices. In the case of non-path related problems, the exact definition and its application is not so straightforward. Let us consider the case of cliques. Defining cliques in a temporal graph as the set of vertices that interact at least once in the lifetime of the graph would be a bit counter intuitive, as two vertices may just interact

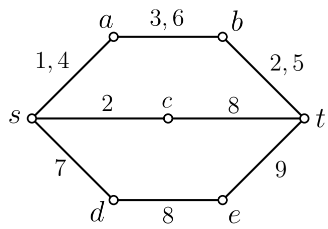

Figure 1 In this temporal graph, the shortest path from s to t is (s, c, t) as it contains two edges; the foremost temporal path is (s, a, b, t) as it arrives at time 5; the fastest temporal path is (s, d, e, t) a it has duration $9 - 7 + 1 = 3$.

at the first time step and never again. To help with this problem, Viard et al. [37] introduced the idea of the *sliding time window* of some size Δ, where they define a temporal clique as a set of vertices where in all Δ consecutive time steps each pair of vertices interacts at least once. There is a natural motivation for this problem, namely to be able to find the contact patterns among high-school students. Following the idea of Viard et al. [37], many other problems on temporal graph were defined wiusing sliding time windows. In this paper we present an overview of works on sliding windows in temporal graphs and at the end provide some open problems and further ideas with potential research topics.

2 Preliminaries and Notations

In the literature there are many (slightly) different notations and terminologies used for certain structures in temporal graph. For the purpose of this paper, we fix the following notation and definitions.

Given a (static) graph $G = (V, E)$ with vertices in V and edges in E, an edge between two vertices u and v is denoted by uv, and in this case u and v are said to be *adjacent* in G. For every $i, j \in \mathbb{N}$, where $i \leq j$, we let $[i, j] = \{i, i + 1, \ldots, j\}$ and $[j] = [1, j]$. Throughout the paper we consider temporal graphs whose underlying graphs are finite and whose time labeling functions only map to finite sets. This implies that there is some $t \in \mathbb{N}$ such that, for every $t' > t$, no edge of G is active at t' in (G, λ). We denote the smallest such t by T, i. e., $T = \max\{t \in \lambda(e) \mid e \in E\}$, and call T the *lifetime* of (G, λ). Unless otherwise specified, n denotes the number of vertices in the underlying graph G, and T denotes the lifetime of the temporal graph $\mathcal{G}$. We refer to each integer $t \in [T]$ as a *time step* of (G, λ). The *instance* (or *snapshot*) of (G, λ) *at time* t is the static graph $G_t = (V, E_t)$, where $E_t = \{e \in E : t \in \lambda(e)\}$. Note that the *size* of a temporal graph $\mathcal{G}$ is $|\mathcal{G}| := |V| + \sum_{t=1}^{T} |E_t|$.

For every $t = 1, \ldots, T - \Delta + 1$, let $W_t = [t, t + \Delta - 1]$ be the Δ-*time window* that starts at time t. For every $v \in V$ and every time step t, we denote the *appearance of vertex* v *at time* t by the pair (v, t) and the *edge appearance* (or *time-edge*) of e at time t by (e, t). For $t \in \lambda(e)$ we also say that e is *active* at time t in (G, λ). That is, for every edge $e \in E$, $\lambda(e)$ denotes the set of time steps at which e is *active*.

A *temporal vertex subset* of (G, λ) is a set of vertex appearances in (G, λ), i.e. a set of the form $S \subseteq \{(v, t) \mid v \in V, t \in [T]\}$. For a temporal vertex subset S and some Δ-time window W_i within the lifetime T of (G, λ), we denote by $S[W_i] = \{(v, t) \in S \mid t \in W_i\}$ the subset of all vertex appearances in S in the Δ-time window W_i. For a Δ-time window W_i within the lifetime of a temporal graph (G, λ), we denote by $E[W_i] = \{e \in E \mid \lambda(e) \cap W_i \neq \emptyset\}$ the set of all edges which appear at some time step within W_i.

3 Known Results on Sliding Windows

In this section we present some of the known results on temporal graphs using sliding windows. As we discussed, the aim is to find a suitable definition for well motivated graph problems, that take in consideration also the changes that appear over time.

3.1 Temporal Cliques

In a (static) graph $G = (V, E)$, a clique $C \subseteq V$ is a collection of vertices, where every two of them are connected. We say that a clique C is maximal, if there exists no other vertex in $V \setminus C$ that is connected to all of the vertices in C. There are many applications of (maximal) cliques for modeling real-world problems. For example, in their work Creamer et al. [13] calculate hierarchical structures in complex (communication) networks using cliques, and in [35] Samudrala and Moult use cliques in the context of protein structure modeling.

Viard et al. [37] extended the notion of cliques to temporal graphs. Their work was motivated by the contact patterns among French high-school students. They studied the dataset with real-world contacts between individuals, captured with sensors. Where an edge e at time t was formed between two subjects if they were close enough to each other at time t for the detection to happen. The aim is to determine groups of students that were interacting more often. The obstacle in this case is how to naturally define such groups. If two students interacted only once and then never again, their interaction should not be considered as "valuable" as in the case when students interact more often, over certain period of time. With this in mind, the authors present the following natural definition of a Δ-clique.

▶ **Definition 2.** *A Δ-clique C in a temporal graph $\mathcal{G} = (G, \lambda)$ with a life-time T, is a pair (X, I), where X is a subset of vertices of G and $I \subseteq [T]$, such that for every two vertices $u, v \in X$ there is a time-edge (uv, t) in $\mathcal{G}$ in every Δ-time window $W_i \in I$.*

Intuitively, among each pair of vertices in X there is a time-edge every Δ time steps in the time interval I. The significance of the parameter Δ is that it measures the level of interaction in Δ-cliques. A small value of Δ means that the interaction among vertices has to occur more often compared to the case of large Δ values. The selection of Δ depends on the data set and the purpose of the analysis.

The authors provide an algorithm that in $O(2^n n^2 \overline{m}^3 + 2^n n^3 \overline{m}^2)$ time computes all maximal Δ-cliques of a temporal graph (G, λ), where $n = V(G)$ and $\overline{m} = \sum_{e \in E(G)} \lambda(e)$. This result was further improved by Himmel et. al. [23] by providing an adaptation of the Bron-Kerbosch algorithm for enumerating maximal cliques, where they improve the running time to $O(2^n T m)$, where $m = |E(G)|$.

Cliques may not be always practical for modelling real-world situations as they can be too restrictive, for example some edges may not exist due to measurement errors or other reasons specific to the application. To overcome this issue, various relaxations of the clique concept have been developed. One popular approach is the use of k-plexes, a degree-based relaxation of cliques that requires every vertex to be connected to all but at most $k - 1$ vertices in the k-plex, excluding itself. Extending this idea to temporal graphs, Bentert et al. [6] introduce the study of $\Delta - k$-plexes, where they relax the condition of Δ-clique by allowing each vertex to have up to $k - 1$ missing connections to other vertices in each Δ consecutive time steps. They adapt the algorithm for Δ-cliques to enumerate them, and provide some heuristic speed-up techniques that are useful when dealing with practical scenarios.

3.2 Temporal Vertex Cover

The vertex cover problem on a static graph G asks for a set of vertices S in $V(G)$, of a minimum size, such that each edge $e \in E(G)$ has at least one endpoint in S (i. e., is covered by at least one vertex in the vertex cover). To extend the idea to temporal graphs one needs to first find a relevant and well motivated definition. For example, requiring that each edge is covered whenever it appears (i. e., there is a vertex cover in every snapshot of the temporal graph), may be a bit too restrictive. A well known motivation behind the vertex cover on static graphs is a problem of placing security guards throughout the airport, where corridors represent edges and two corridors meet in a vertex. Then a vertex cover is a collection of corridor intersections, where we place security guards such that the airport is fully observed by the security. Suppose now that during the day, for some reason, certain corridors are not in use (some gates may be open only during specific times). And suppose now also that a criminal needs a specific amount of time, without any supervision, to execute an illegal activity. Now, to prevent all such acts, we do not need to fully monitor each sector of the airport all the time, but we just have to make sure we check each part often enough. With this in mind, Akrida et al. [4] introduced the notion of sliding window temporal vertex cover.

▶ **Definition 3.** *A Δ-sliding window temporal vertex cover $S \subseteq V(G) \times [T]$ (or Δ-TVC for short) in a temporal graph (G, λ), with a lifetime T, is a collection of vertex appearances, such that each edge $e \in E(G)$ is covered in every Δ-time window $W_i \subseteq [T]$, if it appears.*

When determining Δ-TVC of a given temporal graph, one wants to always find the one of minimum size. In their work Akrida et al. [4] first prove that a relaxed version of the problem (where $\Delta = T$, i. e., each edge has to be covered at least once in the whole lifetime T of the graph) is NP-hard already for the temporal graphs where the underlying graph is a star. For this sub-problem they prove also that the optimal solution cannot be obtained in $O(2^{\epsilon T})$ time (for some small ϵ), assuming the Strong Exponential Time Hypothesis (SETH), as well as that it does not admit a Polynomial-Time Approximation Scheme (PTAS). For the general problem they provide an exact dynamic algorithm running in $O(T\Delta(n+m) \cdot 2^{n\Delta(\Delta+1)})$ time on arbitrary temporal graphs, which cannot admit much more improvements (as it is almost at the lower complexity bound). They complement this result by providing an algorithm that, for graphs where each snapshot has a vertex cover number bounded by k, runs in an FPT time, when parameterized by Δ. They investigate also the problem's approximability and prove that Δ-TVC does not admit a PTAS, even when $\Delta = 2$, maximum degree of the underlying graph is 3 and every connected component of each snapshot is of size at most 7. In addition, they augment this result by providing approximation algorithms with ratios (i) $\ln n + \ln \Delta + \frac{1}{2}$, (ii) $2k$, where k is the maximum number of appearances of an edge in a sliding window, (iii) d, where d is the maximum vertex degree in every snapshot.

The study of Δ-TVC problem was then further extended by Hamm et. al. [21]. The researchers studied the Δ-sliding window vertex cover problem on sparse temporal graphs. They proved that the problem is NP-hard when $\Delta \geq 2$ and the underlying graph G of the temporal graph (G, λ) is a path or a cycle. On the other hand, they developed a polynomial-time algorithm for solving T-TVC on paths and cycles, where T is the lifetime of the temporal graph. This raises the interesting question of whether there exists a boundary value for Δ that distinguishes between the tractable and intractable categories on paths, thus determining the complete dichotomy of the problem. Moreover, for any $\Delta \geq 2$ they augmented these results with a PTAS for Δ-TVC on paths and cycles, which complements the hardness result. In addition, the authors presented three algorithms to counter the hardness of the Δ-TVC problem for arbitrary (non-restricted) temporal graphs. The first algorithm is an exact algorithm for Δ-TVC with an exponential running time dependency on the number of edges in the underlying graph. Using this algorithm, they developed a polynomial-time $(d-1)$-approximation algorithm for any $d \geq 3$, where d is the maximum vertex degree in any time step (which improved on d-approximation algorithm from Akrida et al.). Finally, the authors presented a simple fixed-parameter tractable algorithm with respect to the size of an optimum solution.

3.3 Temporal Coloring

In a static graph G a coloring problem asks for a minimum number of colors associated to vertices, such that two endpoints of each edge are not assigned the same color. A classical motivation behind this problem is allocating radio frequencies to radio towers at specific locations. Here the idea is to allocate different frequencies to towers that are located close enough to cause an overlap in transmission. In this case each tower is represented as a node of the graph, where two of them are connected if the towers are positioned so close that they interfere with each other, and each frequency represents a different color. Now, coloring the graph properly results in an assignment of frequencies, that causes no interference.

Let us consider a bit more evolved scenario, where instead of static radio towers, we observe mobile agents. Here every agent broadcasts information over a specific communication channel while it listens on all others. Therefore, when two agents are in close proximity, they exchange information only if they broadcast on different channels. We assume that agents can switch channels at any time. To ensure maximum information exchange, it is essential to find a schedule of assigning broadcasting channels to the agents over time that minimizes the number of required channels. This should allow each pair of agents to communicate at least once within every small time window when they are close to each other.

Following this motivation Mertzios et al. [33] introduce the study of temporal coloring using sliding windows. Where one wants to determine the coloring of vertex appearances, using the smallest possible number of colors, such that each edge is properly colored (incident vertices are of different color) at least once in every Δ consecutive time steps, if the edge appears. For a formal definition see the following.

▶ **Definition 4.** *A Δ-sliding window temporal coloring (or Δ-TC for short) in a temporal graph (G, λ), with a lifetime T, is a function $\phi : V(G) \times [T] \to \mathbb{N}$, that assigns one color $\phi(v, t)$ to each vertex appearance, such that for every Δ-time window $W_i \subseteq [T]$, and every edge $e \in E[W_t]$ there is at least one time step $t \in W_i$, where e appears and its two endpoints u, v are colored using different colors, i. e., (e, t) is a time-edge in (G, λ) and $\phi(u, t) \neq \phi(v, t)$.*

Mertzios et al. [32] start by studying a subcase of the problem, when $\Delta = T$. In this case the objective is to ensure that each edge is properly colored (its endpoints are of different color) at least once in the whole life time T of the temporal graph. Surprisingly, even the restricted subcase turns out to be NP-hard, already when one is asking if 2 colors are enough to color it properly. This presents a stark contrast to the static case, where identifying if a graph is 2-colorable (bipartite) can be accomplished in linear time. On the positive side they show that the this subcase admits a polynomial kernel, when parameterized by the number of vertices in the input temporal graph. For the general case they prove that the problem is NP-hard, and provide two algorithms for it. One is an exponential-time algorithm, that asymptotically matches the running time lower bound (assuming ETH), and the second one is a linear time FPT algorithm, with respect to the number n of vertices.

In addition to the above mentioned work, some other variations of coloring temporal graphs have been explored (e.g. example [18, 29, 39]), however these studies do not use the approach with sliding windows.

3.4 Temporal Matching

Given a static graph G the problem of (maximum) matching asks for a (maximum) set of pairwise independent edges, that is, edges that share no endpoints. This problem has numerous applications in fields such as scheduling and planning, chemistry modeling, job allocation, and more. Once the time-dimension is added to the graph model, there can be different ways to carry over the definition to temporal graphs.

Following the idea of a sliding time window, Mertzios et al. [31] introduced the problem of Δ-Maximum Temporal Matching (Δ-TM), where one wants to determine a maximum set of time-edges that are pairwise Δ-independent. Two time-edges $(e, t), (f, t')$ are Δ-independent if (i) $e \cap f = \emptyset$, or (ii) $e \cap f \neq \emptyset$ and $|t - t'| \geq \Delta$. In other words, for any feasible solution of Δ-TM, it is not possible to match a vertex more than once within any time interval of duration Δ. This condition can represent scenarios where a short "recovery" period is needed for every vertex that participates in the matching, such as a brief period of rest after engaging in an energy-demanding activity.

In contrast to the Edmonds' polynomial-time algorithm for finding a maximum matching in static graphs, Mertzios et al. [31] prove that Δ-TM does not even admit an approximation algorithm, meaning it is APX hard, already in the case when $\Delta = 2$ and the lifetime T of the temporal graph is 3. In addition, they show that the problem remains NP-hard even if the underlying graph, of the input temporal graph, is just a path. On the positive side, they provide an approximation algorithm for any constant Δ, which achieves an approximation ration of $\frac{1}{2} + \epsilon$, where $\epsilon = \frac{1}{2(2\Delta-1)}$. Besides that, they show that a problem admits two FPT algorithms, one when it is parameterized by the solution size, and the second one, when it is parameterized by the combined parameter Δ and the size of a maximum matching of the underlying graph.

It is worth mentioning that another related variant of Maximum Temporal Matching has been studied (see Baste et al. [5]). In this model the authors do not use the Δ-time windows, but instead require an edge to appear at least Δ consecutive time steps, in order to be eligible for a matching. A temporal matching then consists of independent edge time-blocks of length at least Δ.

4 Further Work

In the previous section we presented some already completed works on temporal graphs, that use the idea of sliding time windows. In this section we focus on problems that, to the best of our knowledge, have not yet been investigated using the sliding windows, and give rise to some interesting research questions.

4.1 Dominating Set

In a static graph G, a dominating set is a subset of vertices $D \subseteq V(G)$, such that each vertex $V(G)$ is either in D or has a neighbor in D. The *Dominating set problem* asks for a dominating set of G of minimum size. One of the applications of the dominating set is in routing protocols for ad hoc wireless networks. The fundamental concept behind this approach involves identifying a dominating set within a network of devices and using these dominating nodes for message routing. More specifically, when a user u wants to transmit a message to a user v, the routing process consists of determining the shortest path between the dominating neighbors of user u and user v. By ensuring that all devices admit at least one dominating neighbor, this method guarantees the delivery of messages.

In the word where these agents become mobile (i. e., they travel around the space), one can model this problem using temporal graph, where the aim is to find a *temporal dominating set*. Similarly as in other cases, we do not necessarily want to find a dominating set in every time step (as this would be to costly), therefore an approach with sliding time windows would be of use. We propose the following definition.

▶ **Definition 5.** *A Δ-sliding window temporal dominating set (Δ-SWDS) is a subset of vertex appearances $D \subseteq V(G) \times [T]$, of a temporal graph (G, λ), with the lifetime T, such that for any vertex appearance (v, t) the following holds:*
1. *$(v, t') \in D$, where $|t - t'| \leq \Delta$ or*
2. *$(u, t') \in D$, where u is a neighbor of v in G and $|t - t'| \leq \Delta$.*

Intuitively, any vertex of the underlying temporal graph is at any time step t either at most Δ time-units away from being in D, or it has a neighbor that is at most Δ time-units away from being in D. Since the Dominating set problem is already NP-hard on static graphs, it remains hard also for temporal graphs. So the interesting research question for the Δ-SWDS would be if there exist any exact algorithms for it, i. e., some FPT algorithms, or maybe some approximation algorithms.

It is important to mention that there already exist some variations of the dominating set problems on temporal graphs. Casteigts and Flocchini [8] propose three different definitions of dominating sets on temporal graphs, namely temporal dominating set, evolving dominating set and permanent dominating set. In the temporal and evolving dominating set, one wants to determine the smallest set of vertices D, such that, in the temporal case, each vertex is dominated in at least one time step, and in the evolving case, each vertex is dominated in every time step. While the evolving dominating set D consists of vertex appearances, such that all vertex appearances are dominated in each time step. More specifically, in the first two cases, once a vertex is selected to be in D it is in D for all lifetime of the graph, while in the last case one vertex can be in D only at specific times. Some research has been done for aforementioned problems. For interested readers, we recommend exploring the following works [20, 28, 38], among others.

4.2 Edge Cover

The *minimum edge covering problem* on a static graph G asks for a minimum set $E_C \subseteq E(G)$ of edges such that every vertex in $V(G)$ is incident to at least one edge in E_C. Calculating a minimum edge cover can be done in polynomial time, by finding a maximum matching and then extending it greedily until all vertices are covered.

For the version of the edge covering problem on temporal graphs we propose the following definition.

▶ **Definition 6.** *A Δ-sliding window temporal edge covering (Δ-SWEC) is a subset of edge appearances $EC \subseteq E(G) \times [T]$, of a temporal graph (G, λ), with the lifetime T, such that every vertex appearance (v, t) is incident to at least one time-edge from the selected set $EC \subseteq E \times T$, in every time window $t \in W'_t$.*

Since many problems become significantly more challenging when dealing with temporal graphs, it would be really interesting to explore whether the same holds true for the Δ-SWEC problem. Applying the exact approach used for static graphs may not yield direct results, as it requires to first find a (suitable definition of a) temporal maximum matching. It is worth noting that in Section 3.4 we presented a Δ-Maximum Temporal Matching, which turns out to be NP-hard.

4.3 Periodic connectivity

We say that a temporal graph (G, λ) is *temporally connected* if there exists a temporal path among each pair of vertices. Some results regarding the connectivity of temporal graphs have already been established, for example [3, 26, 30]. However, what if we introduce additional constraints and require that each vertex can reach any other vertex within every Δ time-window? In such cases, we refer to the temporal graph (G, λ) as being Δ-*temporally connected*. It would be interesting to study, for example, what is the minimum number of labels needed to label a given graph G in such a way that ensures Δ-temporal connectivity of (G, λ)? We can further restrict this problem by allowing only limited number k of labels to be added per each edge.

4.4 (Temporal) Graph Classes

Based on the properties of the studied graphs, we can assign them into different graph classes. For instance there are graphs that are k-colorable (can be properly colored using k colors), k-regular (each vertex is of degree k), or planar (can be drawn on a plane without any edges crossing), among others.

To extend the concept of graph classes to the temporal setting with sliding windows, we propose introducing temporal graph classes. One such class could be the Δ *sliding window k-colorable temporal graphs*, which refers to temporal graphs that can be temporally colored using k colors. Another class would be the Δ *sliding window k-regular temporal graphs*, where each vertex admits exactly k different neighbors in every Δ time-window, or perhaps each vertex v admits exactly k different neighbors in a time step $t' \in W_t$ for every time window W_t. Similarly, we can define Δ *sliding window planar temporal graphs* as temporal graphs that are planar in some $t' \in W_t$ for every time-window W_t. Further refinement of these classes is possible by imposing additional restrictions. For example, we can consider temporal graphs that are 3-*colorable in every* 5-*time window*. In such graphs, every vertex appearance (v, t') is assigned one of three colors, ensuring that within each sliding window W_t of size 5, there is at least one time step where the edge e, that appears in W_t is properly colored.

Overall, these extensions allow for the classification of temporal graphs based on their temporal characteristics, enabling the exploration of various graph classes in the context of sliding windows.

5 Conclusion

The study of temporal graphs has emerged as an important area of research with significant implications for understanding and analyzing dynamic systems. In this paper, we have presented a short overview of the works on sliding windows in temporal graphs. The concept of a sliding time window allows us to focus on specific temporal intervals within the lifetime of a temporal graph, providing valuable insights into the changing behavior and patterns of interactions. Give that this research field is fairly young, there are still many intriguing questions and challenges to be addressed. We presented some of them here and hope that this work inspires further exploration and investigation into these intriguing problems.

References

1. Eric Aaron, Danny Krizanc, and Elliot Meyerson. DMVP: foremost waypoint coverage of time-varying graphs. In *Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 29–41, 2014.
2. Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. Ephemeral networks with random availability of links: The case of fast networks. *Journal of Parallel and Distributed Computing*, 87:109–120, 2016.
3. Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017.
4. Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020.
5. Julien Baste, Binh-Minh Bui-Xuan, and Antoine Roux. Temporal matching. *Theor. Comput. Sci.*, 806:184–196, 2020.
6. Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, and René Saitenmacher. Listing all maximal k-plexes in temporal graphs. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 41–46, 2018.
7. Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285, 2003.
8. Arnaud Casteigts and Paola Flocchini. Deterministic algorithms in dynamic networks: Problems, analysis, and algorithmic tools. Technical report, Defence R&D Canada, 2013.
9. Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
10. Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021.
11. Jiehua Chen, Hendrik Molter, Manuel Sorge, and Ondrej Suchý. Cluster editing in multi-layer and temporal graphs. In *Proceedings of the 29th International Symposium on Algorithms and Computation (ISAAC)*, volume 123, pages 24:1–24:13, 2018.
12. Andrea E. F. Clementi, Claudio Macci, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Flooding time of edge-markovian evolving graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1694–1712, 2010.
13. Germán Creamer, Ryan Rowe, Shlomo Hershkop, and Salvatore J. Stolfo. Segmentation and automated social hierarchy detection through email network analysis. In *Advances in Web Mining and Web Usage Analysis, 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007*, Lecture Notes in Computer Science, pages 40–58, 2007.
14. Jessica A. Enright, Kitty Meeks, George B. Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. *Journal of Computer and System Sciences*, 119:60–77, 2021.
15. Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On temporal graph exploration. *Journal of Computer and System Sciences*, 119:1–18, 2021.
16. Afonso Ferreira. Building a reference combinatorial model for MANETs. *IEEE Network*, 18(5):24–29, 2004.
17. Paola Flocchini, Bernard Mans, and Nicola Santoro. Exploration of periodically varying graphs. In *Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC)*, pages 534–543, 2009.
18. Subhankar Ghosal and Sasthi C Ghosh. Channel assignment in mobile networks based on geometric prediction and random coloring. In *Proceedings of the 40th IEEE Conference on Local Computer Networks (LCN)*, pages 237–240, 2015.

19 George Giakkoupis, Thomas Sauerwald, and Alexandre Stauffer. Randomized rumor spreading in dynamic graphs. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 495–507, 2014.

20 Leonidas J. Guibas, Nikola Milosavljevic, and Arik Motskin. Connected dominating sets on dynamic geometric graphs. *Comput. Geom.*, 46(2):160–172, 2013.

21 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, February 22 - March 1, 2022*, pages 10193–10201, 2022.

22 Klaus Heeger, Danny Hermelin, George B. Mertzios, Hendrik Molter, Rolf Niedermeier, and Dvir Shabtay. Equitable scheduling on a single machine. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 11818–11825. AAAI Press, 2021.

23 Anne-Sophie Himmel, Hendrik Molter, Rolf Niedermeier, and Manuel Sorge. Adapting the bron-kerbosch algorithm for enumerating maximal cliques in temporal graphs. *Social Network Analysis and Mining*, 7(1):35:1–35:16, 2017.

24 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

25 Nina Klobas, George B. Mertzios, Hendrik Molter, Rolf Niedermeier, and Philipp Zschoche. Interference-free walks in time: temporally disjoint paths. *Auton. Agents Multi Agent Syst.*, 37(1), 2023.

26 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, August 22-26, 2022, Vienna, Austria*, volume 241, pages 62:1–62:15, 2022.

27 Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007.

28 Subhrangsu Mandal and Arobinda Gupta. Approximation algorithms for permanent dominating set problem on dynamic networks. In *Distributed Computing and Internet Technology - 14th International Conference, ICDCIT 2018, Bhubaneswar, India, January 11-13, 2018, Proceedings*, volume 10722 of *Lecture Notes in Computer Science*, pages 265–279. Springer, 2018.

29 Andrea Marino and Ana Silva. Coloring temporal graphs. *Journal of Computer and System Sciences*, 123:171–185, 2022.

30 George B. Mertzios, Othon Michail, and Paul G. Spirakis. Temporal network optimization subject to connectivity constraints. *Algorithmica*, 81(4):1416–1449, 2019.

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

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

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

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

35 Ram Samudrala and John Moult. A graph-theoretic algorithm for comparative modeling of protein structure11edited by f. cohen. *Journal of Molecular Biology*, 279(1):287–302, 1998.

36 John Kit Tang, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. Characterising temporal distance and reachability in mobile and online social networks. *Computer Communication Review*, 40(1):118–124, 2010.

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

38 John Whitbeck, Marcelo Dias de Amorim, Vania Conan, and Jean-Loup Guillaume. Temporal reachability graphs. In *The 18th Annual International Conference on Mobile Computing and Networking, Mobicom'12, Istanbul, Turkey, August 22-26, 2012*, pages 377–388. ACM, 2012.

39 Feng Yu, Amotz Bar-Noy, Prithwish Basu, and Ram Ramanathan. Algorithms for channel assignment in mobile wireless networks using temporal coloring. In *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems (MSWiM)*, pages 49–58, 2013.

40 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *Journal of Computer and System Sciences*, 107:72–92, 2020.