Two moves per time step make a difference

Two Moves per Time Step Make a Difference

Thomas Erlebach
Department of Informatics, University of Leicester, Leicester, England
te17@leicester.ac.uk

Frank Kammer
THM, University of Applied Sciences Mittelhessen, Giessen, Germany
frank.kammer@mni.thm.de

Kelin Luo
School of Management, Xi'an Jiaotong University, Xianning West Road, Xi'an, China
luokelin@stu.xjtu.edu.cn

Andrej Sajenko
THM, University of Applied Sciences Mittelhessen, Giessen, Germany
andrej.sajenko@mni.thm.de

Jakob T. Spooner
Department of Informatics, University of Leicester, Leicester, England
jts21@leicester.ac.uk

Abstract

A temporal graph is a graph whose edge set can change over time. We only require that the edge set in each time step forms a connected graph. The temporal exploration problem asks for a temporal walk that starts at a given vertex, moves over at most one edge in each time step, visits all vertices, and reaches the last unvisited vertex as early as possible. We show in this paper that every temporal graph with n vertices can be explored in $O(n^{1.75})$ time steps provided that either the degree of the graph is bounded in each step or the temporal walk is allowed to make two moves per step. This result is interesting because it breaks the lower bound of $\Omega(n^2)$ steps that holds for the worst-case exploration time if only one move per time step is allowed and the graph in each step can have arbitrary degree. We complement this main result by a logarithmic inapproximability result and a proof that for sparse temporal graphs (i.e., temporal graphs with $O(n)$ edges in the underlying graph) making $O(1)$ moves per time step can improve the worst-case exploration time at most by a constant factor.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Temporal Graph Exploration, Algorithmic Graph Theory, NP-Complete Problem

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.141

Category Track C: Foundations of Networks and Multi-Agent Systems: Models, Algorithms and Information Management

Funding *Kelin Luo*: This work was partially supported by the China Postdoctoral Science Foundation (Grant No. 2016M592811), and the China Scholarship Council (Grant No. 201706280058).
Andrej Sajenko: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 379157101.

Acknowledgements We would like to thank an anonymous reviewer for a helpful suggestion regarding the presentation of the proof of one of the lemmas.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 141; pp. 141:1–141:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Temporal graphs, or time-varying graphs, are graphs whose edge set changes over time. Due to the prevalence of dynamic networks whose links change over time in many application settings (e.g., wireless mobile networks, transportation networks, social networks), the study of algorithmic aspects of temporal graphs has received increasing attention recently [14]. A temporal graph $\mathcal{G}$ with lifetime τ is a sequence of graphs $G_i = (V, E_i)$, for $i = 1, \ldots, \tau$, that all have the same vertex set, but possibly different edge sets. A particular problem of interest is the *temporal exploration problem* (TEXP) where an agent starts at a given vertex $s \in V$ and aims to visit all vertices in V as quickly as possible (i.e., minimizing the time step in which the last unvisited vertex is reached) while making one move (either stay at the current vertex or move to a neighboring vertex in the current graph) in each time step. The existence of such an exploration is guaranteed if the graph G_i in each time step is a connected graph [13], and so it has become customary to study TEXP for temporal graphs with this property [6, 7, 13]. We make the same assumption throughout this paper. The number of vertices of the temporal graph under consideration is denoted by n.

It is known that every temporal graph (under the assumption that the graph in each step is connected) can be explored in $O(n^2)$ steps and that there are temporal graphs that require $\Omega(n^2)$ steps to be explored [6]. The construction of temporal graphs that require $\Omega(n^2)$ steps for an exploration from [6] produces temporal graphs for which the graph in each step has a vertex of high degree (the graph in each step is a star, so one vertex has degree $n-1$) and in which the edge set changes in every step (the center of the star changes in each step). This poses the natural question whether a better upper bound on the worst-case exploration time holds if any of these properties is avoided, i.e., if the graph in each step has bounded degree or if the edge set of the graph changes only in every other step (which is, up to a factor of two, equivalent to allowing the agent to make two moves instead of one in each step). A first step towards answering this question was made in [7], where it was shown that $O(\log \Delta \cdot \frac{n^2}{\log n})$ steps[1] suffice for the exploration if the graph in each step has maximum degree at most Δ. For constant Δ, this proves that $O(\frac{n^2}{\log n})$ steps suffice.

In this paper, we present a substantial further improvement by showing that $O(n^{1.75})$ steps suffice for an exploration if either the graph in each step has bounded degree, or if the graph in each step has arbitrary degree but the agent can make two moves in each step (or, equivalently, if the agent can make one move per step but the graph changes only in every other step). Surprisingly, the improvement for both cases follows from the same analysis: The key insight is that the better bound on the number of time steps required for an exploration can be proved if the graph in each time step admits a spanning tree of bounded degree. The existence of such a spanning tree is obvious if the graph itself is connected and has bounded degree, and it follows in the model with two moves per step because the square of any connected graph has a spanning tree of bounded degree.

To complement our positive result, we also show that letting the agent make c moves per step, for any constant c, cannot improve the worst-case exploration time for any family of temporal graphs where the underlying graph (the union of the graphs in all steps) has $O(n)$ edges by more than a constant factor. Furthermore, we show that it is *NP*-hard to approximate the exploration time with approximation ratio better than $c \log n$ for some constant c, both for the case of bounded degree and the case of two moves per step.

[1] The conference paper [7] proves a weaker bound of $O(\Delta \log \Delta \cdot \frac{n^2}{\log n})$, but a simple change in one calculation shows that the factor Δ can be avoided.

The remainder of the paper is organized as follows. Section 1.1 discusses related work. Preliminaries are given in Section 2. The main results, showing that exploration in $O(n^{1.75})$ time steps is possible for the case of bounded degree and the case of two moves per step, are presented in Sections 3 and 4, respectively. The result that bounds the improvement obtainable by using c moves per step for sparse temporal graphs and the inapproximability results appear in Sections 5 and 6, respectively. Section 7 concludes the paper.

1.1 Related Work

Brodén et al. [3] studied a temporal analogue of the traveling salesperson problem (TSP) in which the graph is a complete graph in every step and the cost of every edge is either 1 or 2 in each time step, with each edge being allowed to change its cost at most k times over the graph's lifetime. They provided an approximation algorithm with approximation ratio $2 - \frac{2}{3k}$. Michail and Spirakis [13] studied this model as well, showing the general problem to be APX-hard and presenting a $(1.7 + \varepsilon)$-approximation algorithm. They also considered the temporal exploration problem and showed that it is NP-hard to decide if a temporal graph can be explored if no restrictions are placed on the graph in each step. They therefore suggested making the assumption that the graph is connected in each step, which has turned out to be a very useful model to study. They proved that it is NP-hard to approximate the temporal exploration problem under this assumption with ratio $2 - \varepsilon$. Erlebach et al. [6] strengthened this result and proved that approximation with ratio $n^{1-\varepsilon}$ for any $\varepsilon > 0$ is NP-hard. Moreover, they constructed a concrete family of temporal graphs for which exploration takes $\Omega(n^2)$ time. They also presented further results for special graph classes, including upper bounds of $O(n^{1.5}k^2 \log n)$ steps for underlying graphs of treewidth k and $O(n \log^3 n)$ steps for the case that the underlying graph is a $2 \times n$ grid. For the case that the underlying graph is a planar graph of maximum degree 4 and the graph in each step is a path, they proved that $\Omega(n \log n)$ steps can be necessary in the worst case. The temporal exploration problem for the special case where the underlying graph is a ring has been studied for the setting of T-interval-connectivity (the intersection of the graphs of any T consecutive time steps is connected) by Ilcinkas and Wade [11]. Decentralized algorithms for the exploration of temporal rings have been studied by Di Luna et al. [5]. Temporal exploration for the case where the underlying graph is a cactus has been studied by Ilcinkas et al. [10].

For surveys of other work on algorithmic aspects and different models of temporal or time-varying graphs we refer to [4, 12]. Examples of recent work include results on the design of temporal networks [1], on temporal (s, t)-separation problems [8, 15], and on temporal vertex cover with sliding time windows [2].

2 Preliminaries

▶ **Definition 2.1** (Temporal Graph). *We represent a temporal graph $\mathcal{G}$ with underlying graph $G = (V, E)$ using an ordered sequence of static graphs: $\mathcal{G} = \langle G_1, G_2, \ldots, G_\tau \rangle$. The subscripts $i \in 1, 2, \ldots, \tau$ indexing the graphs in the sequence are the discrete time steps 1 to τ, where τ is known as the lifetime of G. Each G_i represents the structure of G in time step i. More precisely, $G_i = (V, E_i)$ is a subgraph of G with $V(G_i) = V(G)$ and $E_i \subseteq E$ for all $1 \leq i \leq \tau$.*

▶ **Definition 2.2** (Temporal walk). *A temporal walk W through a temporal graph $\mathcal{G}$ is given as an alternating sequence of vertices and edge-time pairs*

$$W = v_0, (e_0, i_0), v_1, (e_1, i_1), v_2, \ldots, v_{k-1}, (e_{k-1}, i_{k-1}), v_k$$

that starts at vertex v_0 and ends at vertex v_k. Additionally, we require that $i_0 < i_1 < \ldots < i_{k-1}$, so that an agent following W can traverse at most one edge per time step. Each edge-time pair (e_j, i_j) denotes the traversal of edge $e_j = \{v_j, v_{j+1}\}$ at time step i_j. For such a traversal to be possible, e_j must be present in graph G_{i_j}, i.e., $e_j \in E_{i_j}$. We say that the walk W departs at time i_0 (or, at a time $i' \leq i_0$ if we imagine the walk to wait at v_0 from time i' to time i_0), and arrives at time $i_{k-1} + 1$.

We often view a temporal walk W, defined as above, as describing the movement of an agent that is initially located at v_0 and can, in each step, either stay at its current vertex or move to a neighboring vertex in the current graph.

▶ **Lemma 2.3** (Reachability Lemma [6]). *Let $\mathcal{G}$ be a temporal graph with vertex set V, and assume that $\mathcal{G}$ is connected in each step. Then an agent situated at any vertex $u \in V$ at any time $t \leq \tau - n$ can reach any other vertex $v \in V$ in at most $|V| - 1 = n - 1$ steps, i.e., by time step $t + n - 1$.*

▶ **Problem** (Temporal Exploration). *An instance of the* Temporal Exploration *(TEXP) problem is given by a pair $(\mathcal{G}, s)$, where $\mathcal{G} = \langle G_1, G_2, \ldots, G_\tau \rangle$ is an arbitrary temporal graph with lifetime $\tau \geq |V(G)|^2 = n^2$ (in order to ensure that there exist feasible solutions for any instance), and $s \in V(\mathcal{G})$ is a start vertex. The problem then asks for a temporal walk W that departs from vertex s and visits all vertices of $\mathcal{G}$ and minimizes the arrival time at the last unvisited vertex. We make the extra assumption that the graph is connected in each step; without this it could happen that there exists no valid exploration schedule.*

We end the section with an auxiliary lemma used in the next section.

▶ **Lemma 2.4.** *Let $T = (V, E)$ be a tree with maximum degree Δ and $U \subseteq V$. Then $\lfloor |U|/(\Delta + 1) \rfloor$ pairs of vertices in U can be found such that the paths between all pairs are pairwise vertex-disjoint.*

Proof. We root T at an arbitrary vertex and explain how to iteratively select pairs $\{u, v\} \subseteq U$. The procedure will ensure that, whenever we select a pair of vertices in U, there are at most $\Delta - 1$ other vertices in U that can no longer be paired up and have to be removed from U.

Let $u, v \in U$ be two vertices such that their lowest common ancestor $w = \mathrm{lca}(u, v)$ has the largest possible depth. Note that we may have $w = v$ or $w = u$. Since T has maximum degree Δ, w can have at most Δ children. There can be at most $\Delta + 1$ vertices from U in the subtree rooted at w: Each subtree rooted at a child of w contains at most one vertex from U (by our choice of w), and w itself may be in U. Select the pair $\{u, v\}$ and remove u, v and all other vertices that are in the subtree rooted at w from U, a total of at most $\Delta + 1$ vertices. It is easy to see that the paths between the pairs selected by this procedure are vertex-disjoint. ◀

3 Exploration of Degree-Bounded Temporal Graphs

In this section, our goal is to construct a temporal walk using $O(n^{1+\alpha})$ time steps on a temporal n-vertex graph $\mathcal{G}$ for some α with $0 < \alpha < 1$. We assume that the maximum degree of the graph G_i in each step is bounded by a constant $\Delta \in \mathbb{N}$. While constructing the temporal walk, we distinguish between vertices that already belong to our constructed temporal walk, which we call *seen vertices*, and the remaining vertices, called *unseen vertices*. We first show in Section 3.1 that $O(n^{1.8})$ time steps suffice, and then improve the analysis further to get $O(n^{1.75})$ time steps in Section 3.2.

3.1 Phases, Subphases, Labels and Forbidden Sets

Recall that the reachability lemma allows us to easily construct a temporal walk with arrival time $O(n^2)$ by picking an arbitrary order $s = v_1, v_2, \ldots, v_n$ of vertices and searching for a temporal walk from v_1 to v_2, from v_2 to v_3, and so on, each starting when the previous walk arrives and using $O(n)$ time steps. To find a better solution, we have to avoid fixing the order of the vertices without considering the given temporal graph. We divide our construction into so-called *phases*, each consisting of $O(n)$ time steps. Within each phase we construct several *temporal subwalks*, one ending at each unseen vertex. We say that a subwalk is better than another subwalk if it contains more unseen vertices. Among the constructed subwalks, we then choose a best subwalk and use it to extend the temporal walk constructed so far.

By leaving $O(n)$ time steps between two phases, we can easily connect the subwalks of two subsequent phases by the reachability lemma.

Let ℓ be the number of unseen vertices at the beginning of the current phase. The phase is split into *subphases* $1, 2, 3, \ldots$ where the goal of each subphase is to replace a constant fraction of the subwalks by subwalks that have at least one more unseen vertex. For every unseen vertex v we use a *label* $L(v)$ to store the (ordered sequence of the) unseen vertices of the subwalk ending at v. We also keep track of the subwalk that visits all the unseen vertices in $L(v)$, and we use $L(v)$ to refer to that subwalk if no confusion can arise.

We set $L(v) = v$ at the beginning of the phase. Moreover, we maintain the invariant that after the ith subphase, no subwalk has more than $i + 1$ unseen vertices, i.e., $|L(v)| \leq i + 1$ for all unseen vertices v. We cannot guarantee that there is an improvement w.r.t. the total number of unseen vertices of the subwalks in every time step. Instead, at the beginning of each subphase, we place labels on all unseen vertices. In each time step the idea is to propagate labels from a vertex to one of its neighbors and analyze the total improvement in the spread of the labels. Whenever a label reaches another unseen vertex w, a longer subwalk is found that may replace the label of w in the next subphase. A similar approach was used by Erlebach and Spooner [7].

To measure the improvement during the time steps, we additionally define for every vertex v a *home set* $H(v)$ consisting of labels of unseen vertices that can reach v within the time steps of the current subphase. At the beginning of a subphase, we set $H(v) = \{L(v)\}$ for every unseen vertex v, and $H(v) = \emptyset$ for the remaining vertices. For technical reasons, in subphase i, the size of each home set is bounded by $2i + 1$.

During the extension of a subwalk we have to avoid adding the same unseen vertices to a subwalk again. For this purpose, we store for every unseen vertex v a set of vertices whose subwalks cannot be extended by v (because they already contain v). More precisely, we store a *forbidden set* $F(v)$ of every unseen vertex v that consist of all unseen vertices w with $v \in L(w)$. Note that $v \in F(v)$ for every unseen vertex v. We next describe our first operation on the home sets.

Operation 1: If u and v are connected by an edge in the current time step, we are allowed to copy a label L from the home set $H(v)$ into the home set $H(u)$. In addition, we can delete a label L' from $H(u)$, i.e., we can replace a label L' by L in $H(u)$.

As one can easily see by induction, a label of an unseen vertex u can be part of a home set of a vertex v by the start of time step t only if there is a temporal walk from u to v with arrival time t in the current subphase.

In the following, we motivate a second operation. Let v and u be unseen vertices. If $L(u) \in H(w)$ where $w = v$ or w is a neighbor of v, $u \notin F(v)$ and $|L(u)| \geq |L(v)|$ holds, the subwalk $L(u)$ ending at u can be extended by v and this can be a subwalk ending at v that is better than the current subwalk $L(v)$. In this case, we can take the better subwalk for v, i.e., we define $L_{\mathrm{new}}(v) := L(u) \circ v$. Note that we have that $|L(u)| \le i$ at the end of subphase $i-1$ by our invariant and hence it follows that $|L_{\mathrm{new}}(v)| \le i+1$. Thus, the invariant holds also in the next subphase.

For the remaining time steps of the current subphase, it would seem to be useful to start propagating the new label of v starting from v. However, then we would have several labels ending at v and this would increase the total size of the forbidden sets, which we cannot afford. Therefore, we set $L(v) := L_{\mathrm{new}}(v)$ only at the end of the current subphase. For the remainder of the current subphase, we leave all $L(v)$ in the home sets unchanged. In particular, it is still possible that the label $L(v)$ reaches another unseen vertex w – which means that we may also be able to set $L_{\mathrm{new}}(w) := L(v) \circ w$. Altogether, we get the following operation.

Operation 2: If a label $L(u)$ is in $H(x)$ for some vertex x in the closed neighborhood of an unseen vertex v and if $|L(v)| \le |L(u)|$ as well as $u \notin F(v)$, then we can define $L_{\mathrm{new}}(v) := L(u) \circ v$.

In the following we measure our progress by an increase in the potential function $\phi = |\{u \text{ unseen vertex} \mid L'(u) \neq L(u)\}| + \sum_{v \in V} \sum_{L \in H(v)} |L|$ where $L'(u) := L_{\mathrm{new}}(u)$ if $L_{\mathrm{new}}(u)$ is defined in the current subphase and $L'(u) := L(u)$ otherwise.

▶ **Lemma 3.1.** *Assume that we are in the ith subphase. Given a set of unseen vertices U of size k with $L(u) \in H(u)$, $L'(u) = L(u)$ and $|F(u)| \le 2i$ for all $u \in U$, the operations from above allow us to modify the labels of the unseen vertices and the home sets of all vertices in such a way that ϕ increases by $\Omega(k/i)$ within one time step.*

Proof. We consider a spanning tree T in the graph of $\mathcal{G}$ in the time step under consideration. Since the graph has maximum degree at most Δ, also T obeys this degree bound.

Next we pair up the unseen vertices in such a way that they are connected by pairwise vertex-disjoint paths. However, we are not allowed to pair up vertices where one is in the forbidden set of the other. Therefore, we compute a set of unseen vertices $U' \subseteq U$ such that we can pair up the vertices in U' without taking the forbidden sets into consideration. To determine U', start with $U' = \emptyset$ and greedily iterate over the vertices $u \in U$. Whenever $F(u) \cap U' = \emptyset$ and $L(u) \cap U' = \emptyset$ holds, add u to U'. The latter condition guarantees $F(v) \cap (U' \cup \{u\}) = \emptyset$ for all v already in U'. Since each vertex u in U' can prevent the insertion of at most $|L(u) \setminus \{u\}| + |F(u) \setminus \{u\}| \overset{\text{invariant}}{\le} 3i - 1$ vertices into U', $|U'| = \Omega(k/i)$. Next, pair up the vertices of U' in such a way that, in T, the paths between every two pairs of vertices are pairwise vertex-disjoint. By Lemma 2.4, we obtain at least $\Omega(k/i)$ pairs since the current graph of $\mathcal{G}$ and thus T has constant degree Δ.

We next show that, for each of the chosen pairs of vertices u and v, ϕ increases by at least one. For this purpose, we focus on the path P between u and v in T, and let w be the neighbor of u in P. W.l.o.g. $|L(u)| \le |L(v)|$ holds. We consider two cases.

Case 1. The home set of w contains a label $L(x)$ with $x \notin F(u)$ and $|L(x)| \ge |L(u)|$. Then we define $L_{\mathrm{new}}(u) := L(x) \circ u$ using Operation 2 and remove u from U, but leave all occurrences of $L(u)$ in the home sets unchanged. In this way we get a potential increase since u newly satisfies the condition $L'(u) \neq L(u)$.

Case 2. Otherwise, we argue as follows. Note that, if $|H(w)| = 2i+1$ and $|L(x)| \ge |L(u)|$ for all $L(x) \in H(w)$, then we are in Case 1 since $|F(u)| \le 2i$ for all $u \in U$. Thus, either $|H(w)| < 2i+1$ or $H(w)$ contains a label $L(x)$ that is strictly shorter than $|L(u)|$. Moreover, since $u, v \in U'$, we have $v \notin F(u)$. Since we are not in Case 1, $L(v) \notin H(w)$.

Let f be the function that removes from a set all labels that are shorter than $|L(v)|$. Note that $|f(H(w))| \leq 2i$ because either $|H(w)| \leq 2i$ or $H(w)$ contains a label that is strictly shorter than $|L(u)|$ and thus also strictly shorter than $|L(v)|$.

Now, observe that it is not possible that, for each pair of subsequent vertices x (possibly w) and y (possibly v) on the path from w to v, where x is closer to w than y, $f(H(x)) \supseteq f(H(y))$ holds – this would be a contradiction to $L(v) \notin f(H(w))$ and $L(v) \in f(H(v))$. Therefore, there must be a pair of consecutive vertices x and y on the path, with x closer to w, such that $f(H(x))$ is not a superset of $f(H(y))$. Among all such pairs of vertices, choose the one such that x is closest to w. As we have $|f(H(w))| \leq 2i$, it follows that $|f(H(x))| \leq 2i$. By Operation 1 we copy a label $L \in f(H(y)) \setminus f(H(x))$ from $H(y)$ to $H(x)$. Possibly, we additionally have to remove a shorter label L' from $H(x)$ in order to ensure that $|H(x)|$ remains bounded by $2i+1$. If $|H(x)| = 2i+1$, then L' must exist since $H(x)$ has at most $2i$ labels being $\geq |L(v)|$ long. Thus, ϕ increases. In case $x \in U$, we have to be careful that we do not remove $L(x)$ from $H(x)$ in this operation. Therefore, in that case we do not replace $L(x)$ in $H(x)$ by a longer label unless all other labels in $H(x)$ are longer than $L(x)$. If this is the case and we replace $L(x)$ in $H(x)$, then $H(x)$ now contains $2i+1$ labels longer than $L(x)$ and, since $|F(x)| \leq 2i$, one of these is $L(v')$ for some $v' \notin F(x)$ and we can apply Operation 2 and set $L_{\text{new}}(x) = L(v') \circ x$ and remove x from U. ◀

To guarantee the condition on the size of the forbidden sets in the lemma above, we do not add all unseen vertices u with $L'(u) = L(u)$ into the set U. Instead, we initially define $U = \{u \text{ unseen vertex} \mid L'(u) = L(u) \text{ and } |F(u)| \leq 2i\}$. Since the total label length is at most $i\ell$ by our invariant, there can be at most $\ell/2$ unseen vertices u with $|F(u)| > 2i$. As long as we have $L'(u) \neq L(u)$ for at most $\ell/4$ unseen vertices u in the current subphase, we have $|U| \geq \ell - \ell/2 - \ell/4 = \ell/4$.

Within each time step, the set of vertices u with $L'(u) = L(u)$ shrinks whenever we apply Operation 2. We apply the lemma only while at most $\ell/4$ vertices have been removed from U, so this does not cause a problem.

▶ **Lemma 3.2.** *Given a set of unseen vertices U of size ℓ in the beginning of the ith subphase, $\Theta(ni^3/\ell)$ time steps allow us to increase the length of at least $\ell/4$ subpaths (i.e., labels of unseen vertices) by one.*

Proof. By Lemma 3.1, $\phi \geq \ell/4 + i(2i+1)n$ after $\Theta(i/4 + i^2(2i+1)(n/\ell)) = \Theta(ni^3/\ell)$ time steps (or we set L_{new} for at least $\ell/4$ vertices even earlier). This can happen only if $L'(u) \neq L(u)$ for at least $\ell/4$ unseen vertices u since we have in the home sets at most $(2i+1)$ labels of length i for every vertex. ◀

We stop the construction of the temporal walk in phases when fewer than n^α unseen vertices remain for some α with $0 < \alpha < 1$ and finish the temporal walk by adding the remaining n^α unseen vertices in $O(n^{1+\alpha})$ time steps (by visiting them in arbitrary order using the reachability lemma). Therefore, we can assume for the next lemma that the number of unseen vertices is $\ell \geq n^\alpha$.

▶ **Lemma 3.3.** *After $O(n)$ time steps within one phase, a temporal subwalk consisting of $\Theta(\ell^{1/4}) = \Omega(n^{\alpha/4})$ unseen vertices has been found.*

Proof. By Lemma 3.2, we spend $\Theta(ni^3/\ell)$ time steps in subphase i. Since we have $O(n)$ time steps, the number of subphases x within one phase is bounded by the following equation: $\Theta((n/\ell)(1^3+2^3+\ldots+x^3)) \subseteq O(n)$. This means that we can have $x = \Theta(\ell^{1/4})$ subphases. ◀

By choosing $\alpha = 4/5$, we have to run $(n - n^{\alpha})/n^{\alpha/4} = \Theta(n^{4/5})$ phases until n^{α} vertices remain. Thus, this part uses $\Theta(n^{9/5})$ time steps, which is also true for the construction of the rest of the temporal walk.

▶ **Theorem 3.4.** *Let $\mathcal{G}$ be a temporal graph with n vertices that is connected and has constant degree in every time step. Then, $\mathcal{G}$ admits a temporal walk that visits all vertices and arrives at the last unvisited vertex after $O(n^{9/5}) = O(n^{1.8})$ time steps.*

3.2 A Tighter Analysis

For the analysis in the previous section, we have applied Lemma 3.3 with the pessimistic assumption that the number of unseen vertices at the start of each phase is only n^{α} (giving subwalks visiting $\Theta(n^{\alpha/4}) = \Theta(n^{1/5})$ unvisited vertices). There are clearly more unseen vertices in the earlier phases (e.g., $n-1$ unseen vertices at the start of the first phase, thus admitting a subwalk visiting $\Theta(n^{1/4})$ vertices by Lemma 3.3). To get a tighter bound, we analyze the required number of phases more carefully. Let γ, $0 < \gamma < 1$, be the constant hidden in the bound $\Theta(\ell^{1/4})$ from Lemma 3.3, i.e., each phase visits at least $\gamma \cdot \ell^{1/4}$ unseen vertices if there are ℓ unseen vertices at the start of the phase. An upper bound $f(t)$ on the number of unseen vertices after t phases is now given by the following equation system:

$$\begin{aligned} &f(0) = n \\ &f(t) = f(t-1) - \gamma \cdot (f(t-1))^{1/4}, \text{ for } t \geq 1 \end{aligned} \tag{1}$$

The following claim establishes a closed formula for an upper bound on $f(t)$.

▷ Claim 3.5. $f(t) \leq (n^{3/4} - \frac{3}{4}\gamma t)^{4/3}$ for all $t \geq 0$.

Proof. We prove the claim by induction. For the base of the induction, note that for $t = 0$ we have $f(0) = n = (n^{3/4} - \frac{3}{4}\gamma \cdot 0)^{4/3}$. For the induction step, assume that the claim holds for $t-1$. We want to show that it also holds for t. By the induction hypothesis and using the fact that the function $g(x) = x - \gamma x^{1/4}$ is monotone increasing for $x \geq 1$ (even for $x \geq (\gamma/4)^{4/5}$) we get:

$$\begin{aligned} f(t) &= f(t-1) - \gamma \cdot (f(t-1))^{1/4} \\ &\leq (n^{3/4} - \tfrac{3}{4}\gamma(t-1))^{4/3} - \gamma \cdot \sqrt[4]{(n^{3/4} - \tfrac{3}{4}\gamma(t-1))^{4/3}} \\ &= (n^{3/4} - \tfrac{3}{4}\gamma t + \tfrac{3}{4}\gamma)^{4/3} - \gamma \cdot (n^{3/4} - \tfrac{3}{4}\gamma t + \tfrac{3}{4}\gamma)^{1/3} \end{aligned}$$

We need to show that the right-hand side is bounded by $(n^{3/4} - \frac{3}{4}\gamma t)^{4/3}$. Setting $y = n^{3/4} - \frac{3}{4}\gamma t$, this is equivalent to:

$$\begin{aligned} & (y + \tfrac{3}{4}\gamma)^{4/3} - \gamma \cdot (y + \tfrac{3}{4}\gamma)^{1/3} \leq y^{4/3} \\ \Leftrightarrow \quad & (y + \tfrac{3}{4}\gamma)^{4/3} - y^{4/3} \leq \gamma \cdot (y + \tfrac{3}{4}\gamma)^{1/3} \end{aligned}$$

With $h(x) = x^{4/3}$, the latter inequality is equivalent to

$$h(y + \tfrac{3}{4}\gamma) - h(y) \leq \tfrac{3}{4}\gamma \cdot h'(y + \tfrac{3}{4}\gamma),$$

which holds because the function $h(x)$ is convex. Hence, the inductive step is complete. ◁

Equation (1) is valid as long as the number of unseen vertices is sufficiently large. The value of t for which $f(t)$ becomes smaller than $n^{3/4}$ is clearly smaller than $\frac{4}{3\gamma}n^{3/4}$. Hence, $O(n^{3/4})$ phases of length $O(n)$, a total of $O(n^{1.75})$ time steps, suffice to reduce the number of unseen vertices to a value below $n^{3/4}$, and the remaining unseen vertices can be visited in $O(n^{1.75})$ steps via the reachability lemma.

▶ **Theorem 3.6.** *Let $\mathcal{G}$ be a temporal graph with n vertices and constant degree that is connected in every time step. Then, $\mathcal{G}$ has a temporal walk that visits all vertices and uses $O(n^{7/4}) = O(n^{1.75})$ time steps.*

4 Two Moves per Time Step in a Graph of Unbounded Degree

Erlebach, Hoffmann and Kammer [6] showed that there are temporal graphs of unbounded degree where a temporal walk visiting all vertices needs $\Theta(n^2)$ time steps. We show in this section that we can break this lower bound not only if the graph in each step has bounded degree, but also if we allow up to *two moves* per time step, i.e., in each time step we are allowed to move from a vertex v to a neighbor w of v and then to a neighbor of w.

We handle the two moves in a temporal graph $\mathcal{G} = \langle G_1, G_2, \ldots, G_\tau \rangle$ for some $\tau \in \mathbb{N}$ by adding additional edges between each pair of vertices of distance 2 in each graph G_i ($i = 1, \ldots, \tau$) to obtain a graph G_i^2, the so-called *square graph* of G_i. Afterwards, we can simply use the "one-move-per-time-step approach" in $\mathcal{G}^2 = \langle G_1^2, G_2^2, \ldots, G_\tau^2 \rangle$.

▶ **Lemma 4.1.** *The square of a connected graph has a spanning tree with maximum degree 3.*

Proof. Let G be a connected graph and G^2 the square of G. Compute a spanning tree T in G. This spanning tree T is also a spanning tree in G^2. Now color all edges of T black. Note that, because the children of every vertex of T are directly connected in G^2, we are allowed to connect them in T and still have a subgraph of G^2.

We next reduce the degree of each vertex in T. First, delete all black edges from each vertex v to all its children except one. Starting from that child, we then connect the children of v by a red path, see also Fig. 1. After applying these changes to each vertex v we can observe the following: Each vertex v (except the root) was either the first child in T or not. In the former case, v has a red edge and the remaining black edges can be to a parent and to at most one child. In the second case, v has at most two red edges to siblings and possible one black edge to a child. Hence, in both cases v has maximum degree 3. ◀

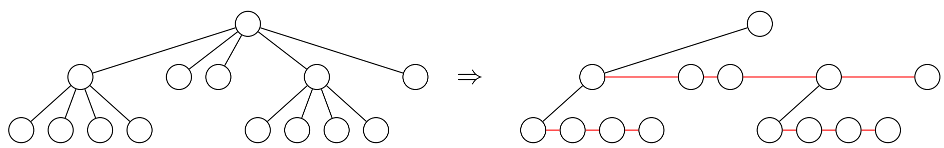

Figure 1 Left: A spanning tree T of a graph G. Right: The transformation of T into a spanning tree in G^2 of maximum degree 3 as shown in Lemma 4.1.

By using the lemma above for the construction of a spanning tree in the proof of Lemma 3.1, we get the following theorem.

▶ **Theorem 4.2.** *Let $\mathcal{G}$ be a temporal graph with n vertices and unbounded degree that is connected in every time step. Then $\mathcal{G}$ has a temporal walk that visits all vertices and uses $O(n^{7/4}) = O(n^{1.75})$ time steps if we allow at least two moves per time step.*

5 Bounded Benefit of c Moves per Step for Sparse Graphs

In the previous section we showed that allowing more than one move per time step enables us to break the existing lower bound of $\Omega(n^2)$ steps on general temporal graphs [6, Theorem 3.5]. We next show that there are certain temporal graph classes where c moves per time step, for an arbitrary integer $c = O(1)$, cannot decrease the worst-case bound on the exploration time by more than a constant factor.

Let $\mathcal{G}$ be a temporal graph with underlying graph $G = (V, E)$. The vertices of G are called *original vertices*. Denote the graph in time step i by G_i. For odd ℓ, we define an operation called *edge-path transformation of length* ℓ that produces a temporal graph $\mathcal{G}'$ with underlying graph G' and graph G'_i in time step i as follows: To construct G' from the underlying graph $G = (V, E)$ of $\mathcal{G}$, the transformation replaces each edge $\{u, v\} \in E$ by a path $\pi_{u,v} = u, a_1, a_2, \ldots, a_{\ell-1}, v$ of ℓ edges, where each a_i is a new *artificial vertex*. For an artificial vertex a_i, we define $o(a_i)$ to be the nearest original vertex, i.e., $o(a_i) = u$ if $i < \ell/2$ and $o(a_i) = v$ if $i > \ell/2$. For an original vertex w, we set $o(w) = w$. If $\{u, v\}$ is present in G_i, the whole path $\pi_{u,v}$ is present in G'_i. If $\{u, v\}$ is not present in G_i, the path $\pi_{u,v}$ without its *middle edge* $\{a_{\lfloor \ell/2 \rfloor}, a_{\lceil \ell/2 \rceil}\}$ is present in G'_i.

A temporal graph with $n \in \mathbb{N}$ vertices is called *sparse* if its underlying graph has $O(n)$ edges. We say that a function $f : \mathbb{N} \to \mathbb{N}$ is *nice* if $f(n/b) = \Omega(f(n))$ holds for any constant $b \geq 1$. For example, it is easy to see that all functions of the form $f(n) = n^g (\log n)^h$ for constant g, h are nice.

▶ **Theorem 5.1.** *Let $c = O(1)$ and let $\mathcal{G}^*$ be a class of sparse temporal graphs such that $\mathcal{G}^*$ is closed under the edge-path transformation of the odd length $\ell \in \{c, c+1\}$. If there is a lower bound on the number of steps for TEXP on $\mathcal{G}^*$ for one move per time step, given as a nice function of the number of vertices, then the same lower bound, up to a constant factor, also applies to TEXP on $\mathcal{G}^*$ if c moves per time step are allowed.*

Proof. Let $\mathcal{G}'$ be the temporal graph obtained from a temporal graph $\mathcal{G} \in \mathcal{G}^*$ by the edge-path transformation of length ℓ. We claim the following: If a temporal walk W' visiting all vertices in $\mathcal{G}'$ uses k time steps with $\leq \ell$ moves per time step, then there is a temporal walk W visiting all vertices in $\mathcal{G}$ consisting of k time steps with one move per time step.

To construct W, we iterate over the time steps and analyze the moves in W' within the current time step. Before and after each time step, we ensure the following invariants. (1) If W' is at a vertex a, then W is at $o(a)$. (2) All original vertices already visited by W' have also been visited by W.

Let a and a' be the (possibly artificial) vertices at which W' is located just before and just after the current time step, respectively. Let $u = o(a)$. Note that W is located at u just before the current step by (1). We let W move from u to $o(a')$ in the current step. Note that, if W' visits during the current step an original vertex that has not yet been visited by W, then that vertex must be $o(a')$. This is because W' starts on the far side of the middle edge on the path to that original vertex, and thus must end on the near side of the middle edge on the same or another path incident with that original vertex. It is easy to see that both invariants hold again after the time step.

Assume there is a lower bound on the worst-case exploration time for $\mathcal{G}^*$ given by a nice function f of the number of vertices. Let $\mathcal{G} \in \mathcal{G}^*$ be a worst case instance where TEXP requires at least $k = f(n)$ steps. Let $\mathcal{G}'$ be obtained from $\mathcal{G}$ by the edge-path-transformation of length ℓ. Using the claim above, we can conclude that TEXP requires at least k steps in $\mathcal{G}'$ even if $\ell \geq c$ moves per time step are allowed. Since $\mathcal{G}' \in \mathcal{G}^*$ and since $\mathcal{G}$ and $\mathcal{G}'$ have the same number of vertices and edges up to constant factors, we have that $k = f(n'/b)$ for some constant b, where n' is the number of vertices of $\mathcal{G}'$. As f is a nice function, we get a lower bound of $\Omega(f(n'))$ for the case where c moves per time step are allowed. ◀

As an application of the theorem above, we can conclude the following from the known lower bound of $\Omega(n \log n)$ steps for sparse temporal graphs in the one-move-per-step model (Theorem 4.1 in [6]).

▶ **Corollary 5.2.** *For a temporal n-vertex graph $\mathcal{G}$ whose underlying graph is planar with maximum degree 4, an optimal exploration can take $\Omega(n \log n)$ steps even if $c = O(1)$ moves per time step are allowed.*

6 Inapproximability Result

▶ **Theorem 6.1.** *It is NP-hard to approximate TEXP with two moves per step on an n'-vertex temporal graph with ratio smaller than $b \log n'$ for some constant $b > 0$.*

Proof. We give a reduction from the Hamiltonian s-t-path problem, which is NP-complete even if the input graph is planar and has maximum degree 3 as shown by Garey, Johnson, and Tarjan [9]. Let an instance of the Hamiltonian s-t-path problem be given by a graph G with n vertices and $s, t \in V(G)$. Assume without loss of generality that $n = 2k$ for some $k \in \mathbb{N}$ with $k \geq 2$ (otherwise, simply add a new leaf t' adjacent to t and consider the Hamiltonian s-t'-path problem in the new graph).

We now construct an n'-vertex temporal graph $\mathcal{G}'$ in two *phases*, for some n' specified later. First, we specify the vertex set and all edges that will be present during at least one step of the first phase.

Underlying graph during the first phase. The construction is illustrated in Fig. 2. Create $2n^c$ copies of G (for an arbitrary constant $c \geq 2$). Form two groups of these copies, each of size n^c, calling the first group T and the second group B. Let $T(i)$ be the *i-th T-copy* of G and $B(j)$ the *j-th B-copy* of G. For all $i \in \{1, \ldots, n^c - 1\}$, connect vertex $t \in V(T(i))$ and $s \in V(T(i + 1))$ by a *quick link* $e_t(i, i + 1)$. Create similar quick links $e_b(i, i + 1)$ between the B-copies. Between $t \in V(T(n^c))$ and $s \in V(B(1))$, let there be a *super quick link*.

Moreover, build a path P consisting of further vertices $v_t(1), \ldots, v_t(12n^{c+1})$. Use the first (last) quarter of the path to connect each vertex s in the T-copies (B-copies) of G to the path such that the minimal distance of two such vertices s on the path is at least $3n$. More precisely, for $k \in \{0, \ldots, n^c - 1\}$, connect vertex $v_t(3kn + 1)$ with $s \in V(T(k + 1))$, and connect $v_t(9n^{c+1} + 3kn + 1)$ with $s \in V(B(n^c - k))$. In total, $\mathcal{G}'$ has $n' = O(n^{c+1})$ vertices.

Temporal realization of the first phase. Let the start vertex be $s \in V(T(1))$. Let the steps $t \in \{1, 2, \ldots, n^{c+1}\}$ constitute the first phase of $\mathcal{G}'$'s lifetime. During this phase, the edges of all $2n^c$ copies of G, the path of length $12n^{c+1}$ and the connections between the T-copies or B-copies and the path P described in the previous paragraph always exist. Additionally, let the edge $e_t(i, i + 1)$ be present only in step $in/2$, for all $i \in \{1, \ldots, n^c - 1\}$. Let the super quick link be present only in step $n^{c+1}/2$. Let the edge $e_b(i, i + 1)$ be present only in step $(n^{c+1} + in)/2$, again for all $i \in \{1, \ldots, n^c - 1\}$.

It is not hard to see that, if G has a Hamiltonian s-t-path, then a temporal walk with two moves per step can visit all vertices in all copies of G within the first phase: In each copy of G, it uses the $n/2$ time steps to follow the $n - 1$ edges of the Hamiltonian s-t-path and then the quick link (or super quick link) to move to the vertex s of the next copy.

Assume now that G does not have a Hamiltonian s-t-path. If a temporal walk W with two moves per step does not use the super quick link, then none of the vertices in the B-copies have been visited in the first phase.

Otherwise, W must have used all quick links connecting the T-copies. We claim that in any two consecutive T-copies, W has missed at least one vertex. To see this, note that an s-t walk in G consists of at least n edges if there is no Hamiltonian s-t-path. Thus, for the walk to arrive at the ith T-copy via a quick link in step $(i - 1)n/2$, visit all vertices of the copy, and leave via a quick link in step $in/2$, it must have used the quick links as the first move in step $(i-1)n/2$ and as the second move in step $in/2$, respectively. But then it is not possible for the walk to visit all vertices of the $(i+1)$th copy and leave it via a quick link.

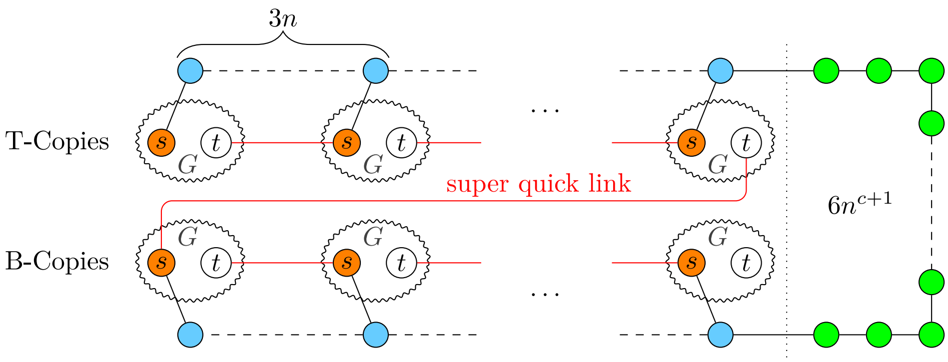

Figure 2 The underlying graph during the first phase. A red line is a quick link.

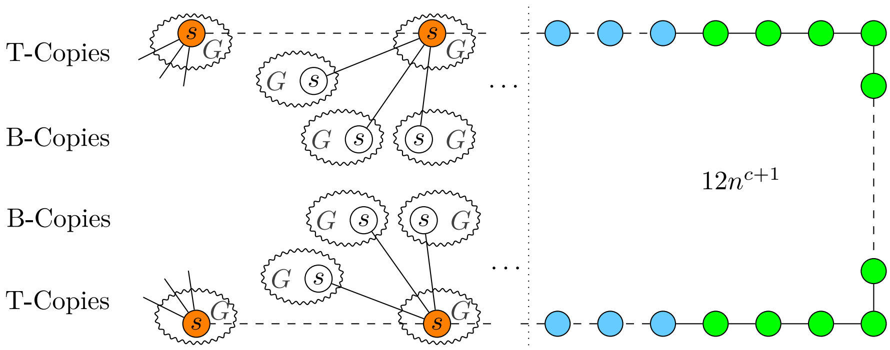

Figure 3 Construction of the temporal graph in the second phase. The connections between the orange vertices differ in the time steps and result from the connections between the labeled vertices in the temporal graph in Fig. 3 in [6].

The second phase. For the whole second phase, the path P is present. Thus, all vertices of P can be visited in $O(n')$ time steps. Therefore, if G has a Hamiltonian s-t-path, $\mathcal{G}'$ can be explored in $O(n')$ time steps.

Erlebach et al. [6, Theorem 4.1] show that TEXP takes $\Omega(N \log N)$ steps on a planar N-vertex temporal graph. More precisely, the proof of their theorem shows that, for arbitrary parameters $r, m \in \mathbb{N}$, there is a planar temporal graph $\mathcal{H}$ with $r+m$ vertices, consisting of r vertices that we call *path vertices* and m vertices that we call *target vertices*, in which $\Omega(r \log m)$ time steps are necessary for visiting all m target vertices. In the temporal graph $\mathcal{H}$, the r path vertices are connected in the form of a path Q in all time steps. The lower bound of $\Omega(r \log m)$ steps for visiting the m target vertices also holds for the model with two moves per step if the length of the rounds in the construction of $\mathcal{H}$ is divided by two. Let $\mathcal{H}'$ be $\mathcal{H}$ with this modification of the length of the rounds.

The idea is now to take in the second phase of our construction the temporal graph $\mathcal{H}'$ for parameters $r = 12n^{c+1}$ and $m = n^c/2$. However, we must construct the temporal graph with the vertices used in Phase 1. Therefore, the second phase of our temporal graph $\mathcal{G}'$ is constructed as follows – see also Fig. 3: Each of the m target vertices v in $\mathcal{H}'$ is identified with a vertex s of a T-copy $T(2i)$ such that no two such vertices s are identified with the same vertex v. Furthermore, the vertex s of each T-copy $T(2i-1)$ is made adjacent to the vertex s of $T(2i)$. In addition, connect exactly the vertices s of two B-copies to each such vertex v. Our path P becomes the path Q. As in Phase 1, the edges in the copies of G are present in all time steps. Note that the only edges that change during the time steps of the second phase are the edges between the m target vertices of $\mathcal{H}'$. Such an edge exists in the ith time step within Phase 2 exactly if it exists in $\mathcal{H}'$ in the ith time step.

Analysis. Assume that G does not have a Hamiltonian s-t-path. Since we have an unvisited vertex in each pair of consecutive T-copies or in each B-copy after Phase 1, we must visit all m target vertices of $\mathcal{H}'$ to reach the unvisited vertices. Thus, we have to spend $\Omega(r \log m) = \Omega(n^{c+1} \log n^c) = \Omega(n' \log n')$ time steps to explore all vertices of $\mathcal{G}'$, where the last equality follows from $r = \Theta(n')$ and $m = n^c/2 = \Theta((n')^{c/c+1})$.

Since establishing whether $\mathcal{G}'$ can be explored in its entirety in just $O(n')$ steps or requires $\Omega(n' \log n')$ steps also decides whether there exists a Hamiltonian s-t-path in G, we get that it is *NP*-hard to approximate TEXP with two moves per step with approximation ratio smaller than $b \log n'$ for some constant $b > 0$. ◀

Using a simple variation of the proof of Theorem 6.1, we also get:

▶ **Corollary 6.2.** *It is NP-hard to approximate TEXP on n-vertex temporal graphs whose underlying graph has bounded degree with ratio smaller than $c \log n$ for some constant $c > 0$.*

7 Conclusion

In this paper we have shown that temporal graphs can be explored in $O(n^{1.75})$ time steps if the graph in each step has bounded degree or if two moves per step are allowed. We remark that our proofs are constructive and also yield polynomial-time algorithms to compute such exploration schedules. Moreover, we have shown that TEXP is *NP*-hard to approximate with ratio better than $b \log n$ for some constant b for the considered cases (bounded degree or two moves per step). The main open problem for these cases is to further reduce the gap between the lower bound of $\Omega(n \log n)$ steps (proved in [6] for bounded degree and in Corollary 5.2 for two moves per step) and the upper bound proved in this paper.

References

1 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. doi:10.1007/s00224-017-9757-x.

2 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal Vertex Cover with a Sliding Time Window. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPIcs*, pages 148:1–148:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.148.

3 Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Online and Offline Algorithms for the Time-Dependent TSP with Time Zones. *Algorithmica*, 39(4):299–319, 2004. doi:10.1007/s00453-004-1088-z.

4 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. doi:10.1080/17445760.2012.668546.

5 Giuseppe Antonio Di Luna, Stefan Dobrev, Paola Flocchini, and Nicola Santoro. Live Exploration of Dynamic Rings. In *Proceedings of the 36th IEEE International Conference on Distributed Computing Systems (ICDCS 2016)*, pages 570–579. IEEE Computer Society, 2016. doi:10.1109/ICDCS.2016.59.

6 Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On Temporal Graph Exploration. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015), Part I*, volume 9134 of *LNCS*, pages 444–455. Springer, 2015. doi:10.1007/978-3-662-47672-7_36.

7 Thomas Erlebach and Jakob T. Spooner. Faster Exploration of Degree-Bounded Temporal Graphs. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *LIPIcs*, pages 36:1–36:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.36.

8 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, and Philipp Zschoche. Temporal Graph Classes: A View Through Temporal Separators. In *Proceedings of the 44th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2018)*, volume 11159 of *LNCS*, pages 216–227. Springer, 2018. doi:10.1007/978-3-030-00256-5_18.

9 M. R. Garey, David S. Johnson, and Robert Endre Tarjan. The Planar Hamiltonian Circuit Problem is NP-Complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi:10.1137/0205049.

10 David Ilcinkas, Ralf Klasing, and Ahmed Mouhamadou Wade. Exploration of Constantly Connected Dynamic Graphs Based on Cactuses. In *Proceedings of the 21st International Colloquium on Structural Information and Communication Complexity (SIROCCO 2014)*, volume 8576 of *LNCS*, pages 250–262. Springer, 2014. doi:10.1007/978-3-319-09620-9_20.

11 David Ilcinkas and Ahmed Mouhamadou Wade. Exploration of the T-Interval-Connected Dynamic Graphs: the Case of the Ring. *Theory of Computing Systems*, 62(5):1144–1160, 2018. doi:10.1007/s00224-017-9796-3.

12 Othon Michail. An Introduction to Temporal Graphs: An Algorithmic Perspective. *Internet Mathematics*, 12(4):239–280, 2016. doi:10.1080/15427951.2016.1177801.

13 Othon Michail and Paul G. Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016. doi:10.1016/j.tcs.2016.04.006.

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

15 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The Complexity of Finding Small Separators in Temporal Graphs. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *LIPIcs*, pages 45:1–45:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.45.