On Temporal Graph Exploration

Thomas Erlebach[1], Michael Hoffmann[1], and Frank Kammer[2]

[1] Department of Computer Science, University of Leicester, England
{te17,mh55}@leicester.ac.uk
[2] Institut für Informatik, Universität Augsburg, Germany
kammer@informatik.uni-augsburg.de

Abstract. A temporal graph is a graph in which the edge set can change from step to step. The temporal graph exploration problem TEXP is the problem of computing a foremost exploration schedule for a temporal graph, i.e., a temporal walk that starts at a given start node, visits all nodes of the graph, and has the smallest arrival time. We consider only temporal graphs that are connected at each step. For such temporal graphs with n nodes, we show that it is **NP**-hard to approximate TEXP with ratio $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$. We also provide an explicit construction of temporal graphs that require $\Theta(n^2)$ steps to be explored. We then consider TEXP under the assumption that the underlying graph (i.e. the graph that contains all edges that are present in the temporal graph in at least one step) belongs to a specific class of graphs. Among other results, we show that temporal graphs can be explored in $O(n^{1.5}k^2 \log n)$ steps if the underlying graph has treewidth k and in $O(n \log^3 n)$ steps if the underlying graph is a $2 \times n$ grid. We also show that sparse temporal graphs with regularly present edges can always be explored in $O(n)$ steps.
Keywords: inapproximability, planar graphs, bounded treewidth, regularly present edges, irregularly present edges

1 Introduction

Many networks are not static and change over time. For example, connections in a transport network may only operate at certain times. Connections in social networks are created and removed over time. Links in wired or wireless networks may change dynamically. Dynamic networks have been studied in the context of faulty networks, scheduled networks, time-varying networks, etc. For an overview, see [5, 15, 18]. We consider a model of time-varying networks called *temporal graphs*. A temporal graph $\mathcal{G}$ is given by a sequence of graphs $G_0 = (V, E_0)$, $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, $\ldots$, $G_L = (V, E_L)$ that all share the same vertex set V, but whose edge sets may differ. The number L is called the *lifetime* of $\mathcal{G}$. We assume that the whole temporal graph is presented to the algorithm.

Standard algorithms for well known problems such as connected components, diameter, reachability, shortest paths, graph exploration, etc. cannot be used directly in temporal graphs. In particular, Berman [2] observes that the vertex version of Menger's theorem does not hold for temporal graphs. Kempe et al. [10]

characterize the temporal graphs in which Menger's theorem holds and show that it is **NP**-complete to decide whether there are two node-disjoint time-respecting paths between a given source and sink. Mertzios et al. [14] show that there is a natural variation of Menger's theorem that holds for temporal graphs. Moreover, the standard algorithms usually optimize only one parameter, but problems in temporal graphs usually have more than one parameter to optimize, e.g., one can search for a *shortest*, a *foremost*, or a *fastest s-t-path* [3], i.e., a path from s to t with a minimal number of edges, earliest arrival time, and a shortest duration, respectively.

We consider the temporal graph exploration problem, introduced in [16] and denoted TEXP, whose goal is to compute a schedule (or temporal walk) with the earliest arrival time such that an agent can visit all vertices in V. The agent is initially located at a start node $s \in V$. In step i ($i \geq 0$) the agent can either remain at its current node or move to an adjacent node via an edge that is present in E_i. We remark that static undirected graphs can easily be explored in less than $2|V|$ steps using depth-first search, while there are static directed graphs for which exploration requires $\Theta(|V|^2)$ steps. The problem to explore a graph (as part of an exploration of a maze) was already formulated by Shannon [19] in 1951.

Flocchini et al. [7] consider the graph exploration problem on temporal graphs with periodicity defined by the periodic movements of carriers. Much of the research is based on models where edges appear with a certain probability [1, 9, 11] or with some kind of periodicity [4, 13]. Except in Sect. 5, we do not assume that edges appear with some periodicity or certain probabilistic properties. Instead, unless stated otherwise, we only assume that the given temporal graph is always connected. Michail and Spirakis [16] observe that without the assumption that the given temporal graph is connected at all times, it is even **NP**-complete to decide if the graph can be explored at all. They also show that, under this assumption, any temporal graph can be explored with an arrival time n^2. They also prove that there is no $(2-\varepsilon)$-approximation for TEXP for any $\varepsilon > 0$ unless $\mathbf{P} = \mathbf{NP}$. They define the dynamic diameter of a temporal graph to be the minimum integer d such that for any time i and any vertex v, any other vertex w can be reached in d steps on a temporal walk that starts at v at time i. They provide a d-approximation algorithm for TEXP, where d is the dynamic diameter of the temporal graph. We note that d can be as large as $n-1$, and hence the approximation ratio of their algorithm in terms of n is only $n-1$. Thus, there is a significant gap between the lower bound of $2-\varepsilon$ and the upper bound of $n-1$ on the best possible approximation ratio, which we address in this paper.

Our contributions. We close the gap between the upper and lower bound on the approximation ratio of TEXP by proving that it is **NP**-hard to approximate TEXP with ratio $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$. Furthermore, we provide an explicit construction of undirected temporal graphs that require $\Theta(n^2)$ steps to be explored. We then consider TEXP under the assumption that the underlying graph (i.e. the graph that contains all edges that are present in the temporal graph in at least one step) belongs to a specific class of graphs. We show that temporal graphs can be explored in $O(n^{1.5}k^2 \log n)$ steps if the underlying graph has treewidth k, in $O(n \log^3 n)$ steps if the underlying graph is a $2 \times n$ grid, and in $O(n)$ steps if the underlying graph is a cycle or a cycle with a chord. Several of these results use a technique by which we specify an exploration schedule for multiple agents and then apply a general reduction from the multi-agent case to the single-agent case. We also show that there exist temporal graphs where the underlying graph is a bounded-degree planar graph and each G_i is a path such that the optimal arrival time of the exploration walk is $\Omega(n \log n)$. Finally, we consider a setting where the underlying graph is sparse and edges are present with a certain regularity and show that temporal graphs can always be explored with an arrival time $O(n)$. A full version of our paper can be found in [6].

The remainder of the paper is structured as follows. In Sect. 2, we give some definitions and preliminary results. Section 3 presents our inapproximability result for general temporal graphs. The results for temporal graphs with restricted underlying graphs are given in Sect. 4. Temporal graphs with regularly present edges are considered in Sect. 5, and Sect. 6 concludes the paper.

2 Preliminaries

Definitions. A *temporal graph* $\mathcal{G}$ with vertex set V and lifetime L is given by a sequence of graphs $(G_i)_{0 \le i \le L}$ with $G_i = (V, E_i)$. Throughout the paper, we only consider temporal graphs for which each G_i is connected and undirected. We refer to i, $0 \le i \le L$, as *time* i or *step* i. The graph $G = (V, E)$ with $E = \bigcup_{0 \le i \le L} E_i$ is called the *underlying graph* of $\mathcal{G}$. If the underlying graph is an X, we call the temporal graph a *temporal X* or a *temporal realization of X*. For example, a temporal cycle is a temporal graph whose underlying graph is a cycle, and a temporal graph of bounded treewidth is a temporal graph whose underlying graph has bounded treewidth.

If an edge e is in E_i, we use the edge-time pair (e, i) to denote the existence of e at time i. A *temporal* (or *time-respecting*) *walk* from $v_0 \in V$ starting at time t to $v_k \in V$ is an alternating sequence of vertices and edge-time pairs $v_0, (e_0, i_0), v_1, \ldots, (e_{k-1}, i_{k-1}), v_k$ such that $e_j = \{v_j, v_{j+1}\} \in E_{i_j}$ for $0 \le j \le k-1$ and $t \le i_0 < i_1 < \cdots < i_{k-1}$. The walk reaches v_k at time $i_{k-1} + 1$. We often explain the construction of a temporal walk by describing the actions of an agent that is initially located at v and can in every step i either stay at its current node or move to a node that is adjacent to v in E_i.

For a given temporal graph $\mathcal{G}$ with source node s, an *exploration schedule* $\mathcal{S}$ is a temporal walk that starts at s at time 0 and visits all vertices. The *arrival time* of $\mathcal{S}$ is the time step in which the walk reaches the last unvisited vertex. An exploration schedule with smallest arrival time is called *foremost*. The temporal exploration problem TEXP is defined as follows: Given a temporal graph $\mathcal{G}$ with source node s and lifetime at least $|V|^2$, compute a foremost exploration schedule. To ensure the existence of a feasible solution, we assume that the lifetime of the given temporal graph $\mathcal{G}$ is at least $|V|^2$. We also consider a multi-agent variant k-TEXP of TEXP in which there are k agents initially located at s. An exploration schedule $\mathcal{S}$ comprises temporal walks for all k agents such that each node of $\mathcal{G}$ is visited by at least one agent. The arrival time of $\mathcal{S}$ is then the time when the last unvisited node is reached by an agent.

A ρ-approximation algorithm for TEXP or k-TEXP is an algorithm that runs in polynomial time and outputs an exploration schedule whose arrival time is at most ρ times the arrival time of the optimal exploration schedule.

Preliminary Results. We establish some preliminary results that will be useful for the proofs of our main results. The following lemma allows us to bound the steps of a temporal walk from one vertex to another vertex in a temporal graph.

Lemma 1 (Reachability). *Let $\mathcal{G}$ be a temporal graph with vertex set V. Assume that an agent is at vertex u. Let v be another vertex and H a subset of the vertices that includes u and v and has size k. If in each of $k-1$ steps the subgraph induced by H contains a path from u to v (which can be a different path in each step), then the agent can move from u to v in these $k-1$ steps.*

Proof. For $i \geq 0$, let S_i be the set of vertices that the agent could have reached after i steps. We have $S_0 = \{u\}$. We claim that as long as $v \notin S_i$, at least one vertex of H is added to S_i to form S_{i+1}. To see this, consider the graph in step $i+1$. By the assumption, the graph induced by H contains a path from u to v. The first vertex on this path that is not in S_i is added to S_{i+1}. As H contains only k vertices, there can be at most $k-1$ steps until v is reached. □

We now show that a solution to k-TEXP yields a solution to TEXP.

Lemma 2 (Multi-agent to single-agent). *Let G be a graph with n vertices. If any temporal realization of G can be explored in t steps with k agents, any temporal realization of G can be explored in $O((t+n)k \log n)$ steps with one agent.*

Proof. Let $\mathcal{G}$ be a temporal realization of G. Consider the exploration schedule constructed as follows: In the first t steps, the k agents explore $\mathcal{G}$ in t steps. Then all k agents move back to the start vertex in n steps. Refer to these $t+n$ steps as a *phase*. Note that the phase can be repeated as often as we like. We construct a schedule for a single agent x by copying one of the k agents in each phase. In each phase, the k agents together visit all n vertices, so the agent that visits the largest number of vertices that have not yet been explored by x must visit at least a $1/k$ fraction of these unexplored vertices. We let x copy that agent in this phase. This is repeated until x has visited all vertices.

The number of unexplored vertices is n initially. Each iteration takes $t+n$ steps and reduces the number of unexplored vertices by a factor of $1-1/k$. Then after $\lceil k \ln n \rceil + 1$ iterations, the number of unexplored vertices is less than $n \cdot (1-1/k)^{k \ln n} \leq ne^{-\ln n} = 1$ and therefore all vertices are explored. □

The next lemma shows that edge contractions do not increase the arrival time of an exploration in the worst case.

Lemma 3 (Edge contraction). *Let G be a graph such that any temporal realization of G can be explored in t steps. Let G' be a graph that is obtained from G by contracting edges. Then any temporal realization of G' can also be explored in t steps.*

Proof. Consider a temporal realization of G'. Consider the corresponding temporal realization of G in which all the contracted edges are always present. Let S be a schedule with an arrival time t that explores the temporal realization of G. S can be executed in t steps in the temporal realization of G' simply by ignoring moves along edges that were contracted. □

Corollary 1. *Let $c < 1$ be a constant and $t(n)$ a function that is monotone increasing and satisfies $t(kn) = O(t(n))$ for any constant $k > 0$, e.g., a polynomial. Let $\mathcal{C}$ be a class of graphs such that any temporal realization of a graph G in the class can be explored in $t(n)$ steps, where n is the number of nodes of G. Let $\mathcal{D}$ be the class of graphs that contains all graphs that can be obtained from a graph G in $\mathcal{C}$ with n vertices by at most cn edge contractions. Then any temporal realization of a graph in $\mathcal{D}$ with n' vertices can be explored in $O(t(n'))$ steps.*

Proof. Let G be a graph in the class $\mathcal{C}$, and let H be obtained from G by at most cn edge contractions. Furthermore, let n and n' be the number of vertices of G and H, respectively. Thus, $n' \geq (1-c)n$. Since any temporal realization of G can be explored in $t(n)$ steps, by Lemma 3, any realization of H can also be explored in $t(n) \leq t(n'/(1-c)) = O(t(n'))$ steps. □

3 Lower Bounds for General Temporal Graphs

While static undirected graphs with n nodes can always be explored in less than $2n$ steps, the following lemma shows that there are temporal graphs that require $\Omega(n^2)$ steps.

Lemma 4. *There is an infinite family of temporal graphs that, for every $n \geq 1$, contains a $2n$-vertex temporal graph $\mathcal{G}$ that requires $\Omega(n^2)$ steps to be explored.*

Proof. Let $V = \{c_j, \ell_j \mid 0 \leq j \leq n-1\}$ be the vertex set of $\mathcal{G}$. For any step $i \geq 0$, the graph G_i is a star with center $c_{i \bmod n}$. The start vertex is c_0. If an agent is at a vertex that is not the current center, the agent can only wait or travel to the current center. As in the next step the center will have changed, the agent is again at a vertex that is not the current center. Hence, to get from one vertex ℓ_j to another vertex ℓ_k for $k \neq j$, n steps are needed: The fastest way is to move from ℓ_j to the center of the current star, and then to wait for $n-1$ steps until that vertex is again the center of a star, and then to move to ℓ_k. The total number of steps is $\Omega(n^2)$. □

Lemmas 2 and 4 also imply the following.

Corollary 2. *For any constant number of agents, there is an infinite family of temporal graphs such that each n-vertex temporal graph in the family cannot be explored in $o(n^2/\log n)$ steps.*

The underlying graph of the temporal graph in the proof of Lemma 4 has maximum degree $|V|-1$. For graphs with maximum degree bounded by d, we can show a lower bound of $\Omega(dn)$ in the following lemma.

Lemma 5. *For every even $d \geq 2$, there is an infinite family of temporal graphs with underlying graphs of maximum degree d that require $\Omega(dn)$ steps to be explored, where n is the number of vertices of the graph.*

Proof. Without loss of generality, n is a multiple of d. We construct $\mathcal{G}$ in two steps. First, we construct n/d copies of a temporal graph $\mathcal{G}'$, which we connect in the end. $\mathcal{G}'$ is the graph with d vertices constructed as in the proof of Lemma 4 (by setting the n to $d/2$). Note that moving from a vertex ℓ_j in a copy of $\mathcal{G}'$ to a vertex ℓ_k for $k \neq j$ in the same copy of $\mathcal{G}'$ requires $\Omega(d)$ steps.

Let $\mathcal{G}_1, \ldots, \mathcal{G}_{n/d}$ be the n/d copies of $\mathcal{G}'$. For all $i = 1, \ldots, n/d - 1$, connect $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$ by merging vertex ℓ_1 of $\mathcal{G}_i$ with ℓ_0 of $\mathcal{G}_{i+1}$. Let $\mathcal{G}$ be the graph obtained. Note that the underlying graph of $\mathcal{G}$ has maximum degree d (the vertices that have been merged have degree d, all other vertices ℓ_j have degree $d/2$, and all vertices c_j have degree $d-1$). Note that, by our way of merging, $\mathcal{G}$ is connected at all times as this is true for all copies of $\mathcal{G}'$.

Let us consider an exploration schedule of $\mathcal{G}$. Similar to the arguments used in the proof of Lemma 4, we can now observe that getting from any ℓ_i in one copy of $\mathcal{G}'$ to a different vertex ℓ_j in the same or another copy of $\mathcal{G}'$ takes at least $d/2$ steps (in most of these, the agent may not move). As there are at least $n/d \cdot (d/2 - 2) = \Omega(n)$ such pairs in every exploration schedule of $\mathcal{G}$, we need $\Omega(dn)$ steps in total. □

Theorem 1. *Approximating temporal graph exploration with ratio $O(n^{1-\varepsilon})$ is* **NP***-hard.*

Proof. We give a reduction from the Hamiltonian s-t path problem, which is **NP**-hard [8]. Assume we are given an instance I' of the Hamiltonian s-t path problem consisting of an undirected n'-vertex graph G', a start vertex s, and an end vertex t. We now construct an instance I of the temporal graph exploration problem as follows: Take the temporal graph as constructed in the proof of Lemma 4 with $n = (n')^c$ for some constant c. In addition, replace each ℓ_i by a copy of G'. Call it the *ith copy* of G'. The edges in each copy of G' are present in every step. The edge $\{c_j, \ell_i\}$ is replaced by an edge connecting c_j and vertex s in the ith copy. We also call the vertices c_i the *center vertices*. In addition, we have so-called *quick links*. Each quick link is an edge that connects the vertex t of the i-th copy with the vertex s of the $(i+1)$-th one only in step $i \cdot n'$ for every $1 \leq i < n - 1$. Denote by $\mathcal{G}$ the resulting temporal graph. Note that $\mathcal{G}$ has $n^* = n(1 + n')$ vertices and that $n = \Theta((n^*)^{c/(c+1)})$.

Clearly, if G' has a Hamiltonian path from s to t, then $\mathcal{G}$ can be explored in $O(n^*)$ steps: The agent starts at c_0 and then explores the first copy of G' in n' steps by following the Hamiltonian s-t-path. The agent arrives at t in the first copy of G' at step n', and we can use a quick link in step n' to move to s in the second copy of G', etc. After exploring all copies of G', we can explore all remaining center vertices c_i in $O(n^*)$ steps, i.e., $\mathcal{G}$ can be explored in $O(n^*)$ steps.

Now assume that G' does not have a Hamiltonian s-t-path. This means that a copy of G' cannot be explored in one visit while using both available quick link connections. Hence in the exploration, every copy must either be visited or left via a center vertex. As moving from one copy to another via a center vertex takes n steps, exploring the n copies takes at least $\frac{1}{2}n(n-1)$ steps. So a total of at least $\Omega(n^2) = \Omega((n^*)^{2c/(c+1)}) = \Omega((n^*)^{2-\varepsilon})$ steps are needed, where ε can be made arbitrarily small by choosing c large enough.

Distinguishing whether $\mathcal{G}$ can be explored in $O(n^*)$ steps or whether it requires $\Omega((n^*)^{2-\varepsilon})$ steps therefore solves the Hamiltonian s-t-path problem, and the theorem follows. □

4 Restricted Underlying Graphs

In Sect. 3, we showed that arbitrary temporal graphs may require $\Omega(n^2)$ steps to be explored and that it is **NP**-hard to approximate the optimal arrival time of an exploration schedule within $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$. This motivates us to consider the case where the underlying graph is from a restricted class of graphs. In particular, the underlying graph of the construction from Lemma 4 is dense (it contains $\Omega(n^2)$ edges) and has large maximum degree. For the case of underlying graphs with degree bound d, we could only show that there are graphs that require $\Omega(dn)$ steps. It is therefore interesting to consider cases of underlying graphs that are sparse, or have bounded degree, or are planar. We consider several such cases in this section.

4.1 Lower Bound for Planar Bounded-Degree Graphs

First, we show that even the restriction to underlying graphs that are planar and have bounded degree is not sufficient to ensure the existence of an exploration schedule with a linear number of steps.

Theorem 2. *Even if the underlying graph $G = (V, E)$ of a temporal graph $\mathcal{G}$ is planar with maximum degree 4 and the graph G_i in every step $i \geq 0$ is a simple path, an optimal exploration can take $\Omega(n \log n)$ steps, where $n = |V|$.*

Proof (sketch). Without loss of generality, we assume that $n = 2^k$ for some $k \geq 3$. Consider the following underlying graph G: It contains vertices $V_0 = \{t_i, b_i \mid 0 \leq i \leq n/4 - 1\}$, the edges $\{t_i, t_{i+1}\}, \{b_i, b_{i+1}\}, \{t_i, b_{i+1}\}$ and $\{b_i, t_{i+1}\}$ for $0 \leq i < n/4 - 1$, and a path P of $n/2$ additional vertices that connects t_0 and b_0. It is not hard to see that G is planar: Arrange the vertices as in Figure 1. For each $0 \leq i < n/4 - 1$, draw the edge $\{b_i, t_{i+1}\}$ as shown in the figure and the edge $\{t_i, b_{i+1}\}$ around the outside. We refer to the edges $\{t_{i-1}, t_i\}$ and $\{b_{i-1}, b_i\}$ as *horizontal* edges of column i, and the edges $\{t_{i-1}, b_i\}$ and $\{b_{i-1}, t_i\}$ as *cross* edges of column i. Consider the following temporal realization of G:

The path P is always present. We divide the time into *rounds*, the first round consists of the first $n/2$ steps, etc. For the first round, the graph additionally contains the horizontal edges of all columns. For the next round, the horizontal edges of column $n/8$ are replaced by the cross edges. For the next round, the horizontal edges of columns $n/16$ and $3n/16$ are replaced by the cross edges.

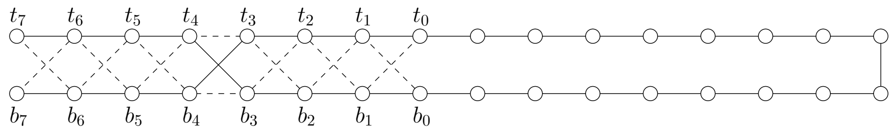

Fig. 1. The underlying graph constructed in the proof of Theorem 2 for $n = 32$. Edges present at the second round are drawn solid, the remaining edges are drawn dashed.

Following the same pattern of replacements (each time the horizontal edges of the middle column in each stretch of horizontal edges are replaced by the cross edges), this is repeated for $O(\log n)$ rounds.

Observe that with $n/2$ steps, any agent can explore either the vertices in V_0 connected to t_0 or those connected to b_0. Furthermore, no matter which of the two sets of vertices the algorithm visits, in the next $n/2$ steps half of the unvisited vertices will be connected to t_0 and half to b_0. Thus, for all start positions of an agent, it requires $\Omega(\log n)$ rounds until all vertices are visited. □

4.2 Underlying Graphs with Bounded Treewidth

Theorem 3. *Any temporal graph whose underlying graph has treewidth at most k can be explored in $O(n^{1.5}k^2 \log n)$ steps.*

Proof. Consider a nice tree decomposition [12, 17] of the underlying graph, i.e., the tree is a binary tree and all nodes are so-called join nodes, introduce nodes, or forget nodes. Select bags as separators via the following procedure: Visit the bags in a post-order traversal of the tree. Select a bag B as a *separator* if the number of unmarked vertices below the bag exceeds $\sqrt{n}$, or if the number of selected bags that are below B and are not descendants of another selected bag is at least 2. If a bag B is selected, mark all vertices in B and below B. Vertices in B are called *separator vertices*. The number of bags selected as separators is $O(\sqrt{n})$. This can be shown as follows. At any point of the procedure, call a selected bag a topmost bag if it is not a descendant of another selected bag. If a bag is selected because there are more than $\sqrt{n}$ unmarked vertices below, the number of topmost bags increases by at most one and $\sqrt{n}$ unmarked vertices become marked. This can happen at most $\sqrt{n}$ times. If a bag is selected because there are two topmost bags below it, the number of topmost bags decreases by one. As the number of topmost bags increases by one at most $\sqrt{n}$ times, it can also decrease at most $\sqrt{n}$ times, and hence at most $\sqrt{n}$ bags are selected because there are two topmost selected bags immediately below them.

The selected separators split the graph into $O(\sqrt{n})$ components (that are not necessarily connected) such that each component contains at most $2\sqrt{n}$ vertices (not counting separators) and is connected to a constant number of separators, i.e., to at most ck separator vertices for some constant c. The algorithm now explores the components one by one. Each component H is explored with ck agents as follows: First, in n steps, move one virtual agent to each of the ck

vertices in the separators that separate the component from the rest of the graph. Then repeat the following operation: Let v be an arbitrary unvisited vertex in H. In each of the next $4ck\sqrt{n}$ steps, v is connected to at least one of the ck separator vertices, so there exists one separator vertex s to which v is connected in at least $4\sqrt{n}$ steps. The agent from s can visit v and return to s in these steps. Therefore, all of the up to $2\sqrt{n}$ vertices in H can be visited in $2\sqrt{n} \cdot 4ck\sqrt{n} = O(kn)$ steps by ck agents. Using the idea in the proof of Lemma 2, this implies that one agent can explore H in $O(k^2 n \log n)$ steps. As there are $O(\sqrt{n})$ components, the whole graph can be explored in $O(n^{1.5}k^2 \log n)$ steps. □

4.3 Cycles and Cycles with Chords

Theorem 4. *Any temporal cycle $\mathcal{C}$ of length n can be explored in $3n$ steps and the optimal number of steps can be computed in polynomial time.*

Proof. Consider two virtual agents, one moving clockwise and one counterclockwise. Since $\mathcal{C}$ is connected, at most one edge of $\mathcal{C}$ is missing at all times. Thus, in each step, one of the two agents can move, except when the agents are in adjacent places and the edge between them is absent. If the edge stays absent for the next n steps, one of the agents can visit the whole cycle by turning around and traversing the cycle. If the edge is present in one of the next n steps, the agents can use the edge to pass each other and continue the traversal of the cycle. One of the virtual agents will have completed the traversal of the whole cycle in at most $3n$ steps. Pick that agent and use it as the solution.

By shortcutting backward and forward moves of the agents such that no vertices are skipped completely, the optimal schedule is of one of a constant number of types: move clockwise around the cycle; move counter-clockwise around the cycle; move clockwise to some vertex v, then counter-clockwise until the cycle is explored; move counter-clockwise to some vertex w, then clockwise until the cycle is explored. The types can be enumerated in polynomial time, and the optimal schedule for each can be calculated in a greedy way. □

Observation 1 *There is a temporal cycle graph in which the optimal exploration requires at least $2n - 3$ steps.*

Proof (sketch). Assume that u, v, w is a subpath of the cycle and the agent is initially at u. Let the edge $\{u, v\}$ be absent for the first $n - 2$ steps, and let the edge $\{v, w\}$ be absent in all steps after that. □

Theorem 5. *A temporal cycle with one chord can be explored in $O(n)$ time.*

Proof. Let the left and right cycle be the two cycles that contain the chord. Check how often the chord is present in the first $10n$ steps. If the chord is present in more than $7n$ steps, use $3n$ of these to explore the (left or right) cycle in which the start node is contained, n to move to the other cycle, and $3n$ to explore that cycle. Otherwise, there are $3n$ steps in which the chord is absent and the remaining graph is a cycle instance. The cycle can be explored in these steps. □

We conjecture that Theorem 5 can be extended to $O(1)$ chords.

4.4 The $2 \times n$ Grid

Theorem 6. *Any temporal $2 \times n$ grid can be explored in $O(n \log n)$ steps with $4 \log n$ agents.*

Proof. We show a slightly more general statement. We show that, if we are given an underlying graph G' being a grid of size $2 \times n'$ and a subgrid G'' of size $2 \times n''$ of G' such that each pair of vertices in G'' is connected in G', then $4 \log n'$ agents initially on some vertices of G'' can explore G'' in $T(n') = O(n'(\log n'))$ time. The theorem follows by taking $G' = G'' = G$.

We start with exploring the left half H' of G''. The idea is to move 4 agents to the corners of H', one to each corner, and all remaining $4(\log n') - 4$ agents to a suitable *middle location* of H'—specified below—using the first $2n'$ steps. This is possible by Lemma 1. For the next $T(n'/2) + n'/2$ steps, in each step where it is possible, we move the 2 agents ℓ_1 and ℓ_2 on the left corners of H' in parallel to the right using only horizontal edges. Similarly, we move the 2 agents r_1 and r_2 on the right corners to the left in parallel. Let i and j be the number of steps of ℓ_1 and r_1, respectively. The middle location is any position between the final position of ℓ_1 and ℓ_2 on the left and the final position of r_1 and r_2 on the right. If the agents on the left and on the right meet, they stop moving and H' is explored. In particular, if H' is a 2×1 grid, ℓ_1 and r_1 (as well as ℓ_2 and r_2) are at the same vertex, i.e., we can stop immediately and $T(1) = O(1)$. Otherwise, in the same $T(n'/2) + n'/2$ steps where the 4 agents move, we explore recursively the subgrid H'' of H' consisting of the columns that are not visited by the 4 corner agents. More precisely, whenever neither the 2 agents ℓ_1 and ℓ_2 nor the 2 agents r_1 and r_2 move, each pair of vertices of H'' is connected in H' and the agents starting in the middle location can explore H'' in $T(n'/2)$ steps. Consequently, after the first $2n'$ steps to place the agents, the next $T(n'/2) + i + j \leq T(n'/2) + n'/2$ steps are enough to explore H'.

We subsequently explore the right half in the same way. The total time to explore G'' is $T(n') \leq 2(2n' + T(n'/2) + n'/2) = O(n' \log n')$. □

Using Lemma 2, we can reduce the number of agents to one.

Corollary 3. *A temporal $2 \times n$ grid can be explored in $O(n \log^3 n)$ steps by one agent.*

5 Temporal Graphs with Regularly Present Edges

We say that a temporal graph has regularly present edges if for every edge e there is a constant integer I_e such that the number of consecutive steps in which e is absent from the temporal graph is at most I_e and at least I_e/c for some constant $c > 1$.

Theorem 7. *A temporal graph $\mathcal{G}$ with regularly present edges that has n vertices and $O(n)$ edges can be explored in $O(n)$ steps.*

Proof (sketch). Round all I_e down to the nearest power of 2; denote the result by J_e. Calculate a minimum spanning tree T with respect to edge weights J_e. Explore the graph by following an Euler tour of T. Moving over an edge e takes at most $I_e \leq 2J_e$ steps, so the total exploration takes at most $2\sum_{e\in T} J_e$ steps.

We next show that $\sum_{e\in T} J_e = O(n)$. Consider any $k \geq 0$ such that T contains at least one edge e with $J_e = 2^k$. Consider the connected components $C_1, \ldots, C_{r_k}$ of $T \setminus \{e \in T \mid J_e = 2^k\}$. Observe that every edge *leaving* a component C_i (i.e., with one endpoint in C_i) must have weight at least 2^k. Let E_i be the set of edges of the underlying graph of $\mathcal{G}$ that leave C_i. Since in each step the graph is connected and hence in each step at least one of the edges of E_i must be present, $\sum_{e\in E_i} 1/(I_e/c) \geq 1$. Thus, $\sum_{e\in E_i} \frac{c}{J_e} \geq 1$. Assign a charge of $c2^k/J_e$ to each $e \in E_i$. The total charge that C_i assigns to E_i is $\sum_{e\in E_i} c2^k/J_e = 2^k \sum_{e\in E_i} c/J_e \geq 2^k$. As an edge receives charge $c2^k/J_e$ from at most two components C_i, no edge receives more than $2c2^k/J_e$ of charge for every fixed k.

The total weight of edges of weight 2^k in T is $2^k(r_k - 1)$. Each of the r_k components assigns a charge of 2^k to edges, so the total charge of the r_k components is greater than the total cost of edges of weight 2^k in T. To bound the total charge that an edge e of G can receive, let the weight of e be $J_e = 2^j$. For $k > j$, e does not receive any charge. For each $k \leq j$, e receives charge at most $2c2^k/2^j$. The total charge received by e is then at most $\sum_{k\leq j} \frac{2c2^k}{2^j} \leq \frac{2c2^{j+1}}{2^j} = 4c$.

So we have that all the weight of T is charged to edges of G, and no edge of G receives more than $8c$ of charge. As G has $O(n)$ edges, the total charge is at most $O(4cn) = O(n)$, and hence the weight of T is $O(n)$. $\square$

6 Conclusion

The study of temporal graphs is still in its infancy, and we do not yet have intuition and a range of techniques comparable to what has been developed over many years for static graphs. Even seemingly simple tasks such as constructing temporal graphs (possibly with an underlying graph from a given family) that cannot be explored quickly is surprisingly difficult. We hope that the methods used in this paper to prove results for temporal graphs, e.g., the general conversion of multi-agent solutions to single-agent solutions, contribute to the formation of a growing toolbox for dealing with temporal graphs.

Our results directly suggest a number of questions for future work. In particular, deriving tight bounds on the largest number of steps required to explore a temporal graph whose underlying graph is an $m \times n$ grid, a bounded degree graph, or a planar graph would be interesting. It would also be interesting to study the approximability of TEXP for restricted underlying graphs, and to identify further cases of underlying graphs, where the temporal exploration problem can be solved optimally in polynomial time.

An interesting variation of TEXP is to allow the agent to make two moves (instead of one) in every time step. The temporal graph constructed in the proof of Lemma 4 can be explored with an arrival time $O(n)$ in the modified model. It would be interesting to determine tight bounds for the modified model.

References

1. Ajtai, M., Komlós, J., Szemerédi, E.: Largest random component of a k-cube. Combinatorica 2(1), 1–7 (1982)
2. Berman, K.A.: Vulnerability of scheduled networks and a generalization of Menger's theorem. Networks 28(3), 125–134 (1996)
3. Bui-Xuan, B., Ferreira, A., Jarry, A.: Computing shortest, fastest, and foremost journeys in dynamic networks. Int. J. Found. Comput. Sci. 14(2), 267–285 (2003)
4. Casteigts, A., Flocchini, P., Mans, B., Santoro, N.: Measuring temporal lags in delay-tolerant networks. In: Proc. 25th Conference on IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2011). pp. 209–218. IEEE (2011)
5. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. IJPEDS 27(5), 387–408 (2012)
6. Erlebach, T., Hoffmann, M., Kammer, F.: On temporal graph exploration. CoRR abs/1504.07976 (2015), http://arxiv.org/abs/1504.07976
7. Flocchini, P., Mans, B., Santoro, N.: Exploration of periodically varying graphs. In: Proc. 20th International Symposium on Algorithms and Computation (ISAAC 2009). LNCS, vol. 5878, pp. 534–543. Springer (2009)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., San Francisco, CA (1979)
9. Karlin, A.R., Nelson, G., Tamaki, H.: On the fault tolerance of the butterfly. In: Proc. 26th Annual ACM Symposium on Theory of Computing (STOC 1994). pp. 125–133. ACM (1994)
10. Kempe, D., Kleinberg, J.M., Kumar, A.: Connectivity and inference problems for temporal networks. J. Comput. Syst. Sci. 64(4), 820–842 (2002)
11. Kesten, H.: The critical probability of bond percolation on the square lattice equals $\frac{1}{2}$. Comm. Math. Phys. 74(1), 41–59 (1980)
12. Kloks, T.: Treewidth, Computations and Approximations, LNCS, vol. 842. Springer (1994)
13. Liu, C., Wu, J.: Scalable routing in cyclic mobile networks. IEEE Trans. Parallel Distrib. Syst. 20(9), 1325–1338 (2009)
14. Mertzios, G.B., Michail, O., Chatzigiannakis, I., Spirakis, P.G.: Temporal network optimization subject to connectivity constraints. In: Proc. 40th International Colloquium on Automata, Languages, and Programming (ICALP 2013), Part II. LNCS, vol. 7966, pp. 657–668. Springer (2013)
15. Michail, O.: An introduction to temporal graphs: An algorithmic perspective. CoRR abs/1503.00278 (2015), http://arxiv.org/abs/1503.00278
16. Michail, O., Spirakis, P.G.: Traveling salesman problems in temporal graphs. In: Proc. 39th International Symposium on Mathematical Foundations of Computer Science 2014 (MFCS 2014), Part II. LNCS, vol. 8635, pp. 553–564 (2014)
17. Scheffler, P.: A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. Technical Report 396, Department of Mathematics, Technische Universität Berlin (1994)
18. Scheideler, C.: Models and techniques for communication in dynamic networks. In: Proc. 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2002). LNCS, vol. 2285, pp. 27–49. Springer (2002)
19. Shannon, C.: Presentation of a maze-solving machine. In: Proc. 8th Conference of the Josiah Macy Jr. Found (Cybernetics). pp. 173–180 (1951)